UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MULTI-CRITERIA SIMULATION OPTIMIZATION WITH STOCHASTIC COEFFICIENTS: METHODS, PERFORMANCE MEASURES, AND TEST BED PROBLEMS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

GLENN W. KURIGER
Norman, Oklahoma
2006

UMI Number: 3238430

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

MULTI-CRITERIA SIMULATION OPTIMIZATION WITH STOCHASTIC COEFFICIENTS: METHODS, PERFORMANCE MEASURES, AND TEST BED PROBLEMS

A DISSERTATION APPROVED FOR THE SCHOOL OF INDUSTRIAL ENGINEERING

BY

____
Dr. Hank Grant

____
Dr. S. Lakshmivarahan

____
Dr. Scott Moses

____
Dr. Shivakumar Raman

____
Dr. Theodore Trafalis

Acknowledgments

I would like to thank my committee members: Dr. Grant, Dr. Moses, Dr. Raman, Dr. Trafalis, and Dr. Lakshmivarahan. They have all provided important insights and direction to my research. This dissertation would not have been possible if it had not been for all of the guidance and support provided by Dr. Grant. I would especially like to thank him for his encouragement and friendship throughout this process.

I would like to thank the Center for the Study of Wireless Electromagnetic Compatibility at the University of Oklahoma (OU Wireless EMC Center) for all of the support, facilities, and resources that they provided me.

Finally, I would like to thank my family and my friends. I would like to thank my parents for always supporting me in whatever I have done. I would especially like to thank my lovely wife Sarah and my adorable daughter Isabelle. They were my inspiration. Their love, support, and encouragement allowed me to endure all of the highs and lows that occurred throughout my doctoral program and especially during my dissertation work.

Table of Contents

List of Tables

List of Figures

Abstract

Many real-world problems involve variables that exhibit random behavior. These problems cannot be adequately solved using traditional optimization techniques. Solving optimization problems that involve stochastic coefficients can be accomplished by incorporating a simulation model within the optimization routine. The simulation model addresses the variability in the system, while the optimization technique addresses the optimization. This process is referred to as simulation optimization. Multi-criteria simulation optimization explores the case where multiple and usually conflicting objectives are desired to be optimized.

Three methods were developed to solve multi-criteria simulation optimization problems by adapting and extending genetic algorithms (GA), tabu search (TS), and lexicographic Nelder-Mead (LNM) methods. These modifications were accomplished largely by incorporating simulation and using goal programming. All three methods included a memory component to ensure that each solution was only simulated once per replication of the method. The three resulting methods were the GA simulation optimization (GA-SO) method, the TS simulation optimization (TS-SO) method, and the LNM simulation optimization (LNM-SO) method. These methods were designed to address multiple objectives, but they could also be used to handle single objective problems.

Currently, there are not sufficient performance measures to allow for the effective comparison of simulation optimization methods. To address this need, this research proposed four global performance measures to allow for the analysis and comparison of simulation optimization methods. These global performance measures examined the

computational efficiency, the quality of solution, and a combination of the efficiency and quality.

There were two measures used to evaluate the computational efficiency or the computational speed. These measures evaluated the number of times the simulation optimization method called the simulation model. These were referred to as the representative operation counts. The number of calls to the simulation was chosen as this was the component of the method that required the most time. The representative operation counts (ROC) were evaluated at two separate points: the number of calls to the simulation model required to complete the method (ROCCM) and the number of calls to the simulation model required to find the best solution (ROCBS). These measures have the advantage that, unlike CPU time, they are not dependent on the particular computer characteristics or programming language. These measures were shown to successfully replace CPU time as a measure of computational efficiency.

The quality of solution was evaluated based on the best solution found (BSF) measure. BSF determines the best solution observed through the implementation of the simulation optimization method that generates the best objective function value. BSF proved to be a very effective measure.

The overall performance of a method was determined based on a combination of the computational speed and the quality of solution. A formula was determined for a performance measure called the time-quality estimator (TQE). This formula was used to determine the overall performance based on whether a decision-maker was most interested in getting a decent solution as fast as possible (time-sensitive), in getting the highest quality solution regardless of the time required (quality-sensitive), or in finding the best solution in the fastest time possible (time-quality neutral). The TQE was determined for each sensitivity level based on weighted values for both speed and quality.

A test bed of problems was developed to allow for current and future simulation optimization methods to be evaluated based on the same set of test problems. Five problems were developed representing five different domains. These problems included the inventory, logistics, project management (PERT), production, and reliability domains.

An experimental study involving the test bed problems was conducted. The independent variables were the domain (representing the test problems) and the simulation optimization method. The dependent variables analyzed included ROCCM, ROCBS, Count, BSF, and Init_Zdif. Count was an additional measure used to evaluate the computational speed. Count was defined as the count of the total number of solutions evaluated. This value included duplicate solutions. Init_Zdif calculated the difference in the objective function value between the BSF and the randomly generated initial solution. This measure was used to further evaluate the quality of solution and it helped determine how dependent the method was on the location of the initial solution.

Three hundred runs were completed and analyzed using a 5×3 factorial design with twenty replications. In addition to the test measures, the percent savings between Count and ROCCM and between ROCCM and ROCBS were examined. Finally, TQE was used to evaluate each multi-criteria simulation optimization method. The GA-SO method was programmed in C++ while the TS-SO and LNM-SO methods were coded using MATLAB.

The global performance measures all were shown to be un-biased and very effective. Thus they provided an excellent source for the comparison of multi-criteria simulation optimization methods. ROCCM and ROCBS allowed for comparison of methods based on computational efficiency. BSF effectively compared methods based on the quality of solution each method generated. The overall performance of the methods, in terms of quality and speed, was evaluated using TQE. The overall performance was determined by whether the decision-maker would prefer a time-sensitive solution, a quality-sensitive solution, or a time-quality neutral solution. Each global performance measure will also work very well for single objective problems.

Each of the test problems performed well and made excellent multi-criteria simulation optimization test bed problems. These problems provide a means for the comparison of multi-criteria simulation optimization methods. Additionally, these test problems can be easily extended to create additional problems to add to the test bed.

All of the simulation optimization methods performed well. The GA-SO method was generally the best method in terms of quality of solution. The LNM-SO or TS-SO methods were the best methods in terms of speed of execution. The LNM-SO method was by far the best method in terms of overall performance for all sensitivity levels.

Chapter 1

Introduction

Simulation is a powerful tool used to model real-world systems having stochastic characteristics. Any system that depends on an unknown arrival or service time clearly falls into this category. Traditionally, the system would be simulated for a given set of input parameters. The system would then be evaluated at various levels and combinations of the input parameters. The scenario that provided the best result would be chosen as the best configuration.

This approach, however, is often difficult to implement in practice. A series of scenarios have to be chosen and simulated. Then these scenarios must be compared to determine the best examined scenario. Performing this type of examination in practice raises several areas of concern. One major concern is how to determine the input combinations that need to be simulated. Clearly, if the best scenario is not considered, then it cannot be chosen. Another concern is the amount of time needed to set-up and run alternative scenarios. For this reason, the number of scenarios considered is normally fairly small. This, however, increases the possibility that a better scenario exists outside the examined set of scenarios. One approach to account for these concerns is simulation optimization.

Simulation optimization essentially combines simulation with an optimization technique to determine the combination of input parameters that produces a near-optimal solution for one or more output measures. Sometimes, and probably more accurately, simulation optimization is referred to as optimization via simulation. The optimization problem involves stochastic responses in the objective function coefficients, the constraint coefficients, or both. Thus, the simulation is used to address this stochastic nature to find feasible and representative values for the coefficients of the model, and optimization is used to find an optimal or near-optimal solution. Multi-criteria simulation optimization explores the case where multiple and usually conflicting objectives are desired to be optimized.

There are many different approaches to simulation optimization that use different optimization techniques and researchers continue to propose new simulation optimization methods. In order to show the effectiveness of a new simulation optimization method, some of the common strategies that researchers use include:

1. The results of the new method for a specific problem are compared against a simple optimization technique such as random search,
2. The new method is used to solve a test bed problem and then its results are compared against previous results, or
3. The new method is used to solve a problem presented in another research study and then the results are compared against the results from the approach used in the other study.

The difficulty of comparing simulation optimization methods through these strategies is that global performance measures to allow for adequate comparisons of approaches are difficult to identify across a range of problem domains. In general, the two most important features of a simulation optimization method are the quality of solution obtained and the speed of execution.

Specifically, common simulation optimization evaluation strategies have the following limitations:

- Results, especially CPU time, are highly dependent on computer capabilities (Processor, RAM, etc…),
- There are very few test bed problems for simulation optimization and no known test bed problems for multi-objective simulation optimization,
- It is typically difficult to exactly replicate another study's problem or program, even for simple test-bed problems, and
- Results are highly dependent on the particular programming methods and platforms used by the researcher. This includes both the simulation program and the optimization program.

1.1 Objective of the Dissertation

This dissertation proposes several means to better compare multi-objective simulation optimization methods to overcome the aforementioned limitations and provides experimental results to evaluate these comparisons. The main objectives of this research are to:

- Develop efficient and robust methods to solve problems with multiple conflicting objectives by integrating simulation and optimization.
- Establish a set of global performance measures that can be used to effectively compare multi-criteria simulation optimization methods, specifically, in comparing the speed of execution between multi-criteria simulation optimization methods. These performance measures should

also be very effective in comparing single objective simulation optimization problems.

- Identify a set of problems from several different domains that can serve as test bed problems for multi-criteria simulation optimization. These problems should be robust and realistic, yet simple enough to allow for correct implementation by future researchers for effective comparison of results.

This research will present three multi-objective simulation optimization methods that use genetic algorithms (GA), tabu search (TS), and lexicographic Nelder-Mead (LNM) optimization techniques, respectively. Both GA and TS have previously been used in simulation optimization methods, but LNM has not. However, all of the methods used in this research are unique in the particular implementation and how they address the existence of multiple objectives or criteria. These optimization techniques were chosen because they only require knowledge of the state of the goals and the corresponding objective function value to guide the search. Additionally, GA and TS have been shown to work well for a wide range of problems from various contexts. Thus, they have proven to be robust and should provide good quality solutions and should serve as good methods to compare the effectiveness of the LNM method, as well as future simulation optimization methods.

The three methods developed for this study adapt and extend a specific optimization technique by incorporating simulation and goal programming. The three resulting methods were the GA simulation optimization (GA-SO) method, the TS simulation optimization (TS-SO) method, and the LNM simulation optimization (LNM-SO) method.

Additionally, this research will explore four global performance measures. Two of these measures are concerned with computational speed or effort. These measures seek to determine the number of calls to the simulation needed to complete the simulation optimization method and the number of calls needed to find the best solution. One measure, the best solution found, is used to determine the quality of solution. A final measure is used to evaluate the overall performance. This measure combines the computational speed and quality of solution to provide an overall rating of how effective the methods are. This measure will vary depending on whether the decision-maker is more concerned with computational speed, quality of solution, or is equally concerned with both.

Finally, five test bed problems were created and analyzed. These problems represent five common and well-studied domains. Specifically, the domains examined include the inventory, logistics, project management (PERT), production, and reliability domains.

The multi-objective simulation optimization methods will be used to evaluate the effectiveness of the proposed global performance measures and the test bed problems. The effectiveness of each of the methods will also be evaluated. Experimental results will be provided to evaluate each of these objectives.

1.2 Outline of the Dissertation

This dissertation is divided into five chapters which describe the techniques, methodology, and the results of the computational analysis evaluating the comparison of performance measures, test bed problems, and the simulation optimization methods. Chapter 2 provides a thorough review of related literature that is needed both for understanding this dissertation and for examining alternative approaches. Chapter 3 presents the methodology that is used to implement the study and explains the simulation optimization description, the description of the simulation model, the optimization methods, and the performance measures. Chapter 4 describes the test problems, the experimental design, and the results of the computational analysis. Chapter 5 presents the general conclusions that can be drawn from this research and offers recommendations for further research.

Chapter 2

Literature Review

This chapter summarizes the relevant literature for this dissertation and explains the concepts and basic methods used throughout this research. The literature review addresses the following categories: general concepts, multi-criteria optimization, simulation optimization, multi-criteria simulation optimization, and evaluation of performance.

2.1 General Concepts

This section discusses the general concepts needed to adequately understand the methods proposed in this dissertation. These concepts are: simulation and optimization.

2.1.1 Simulation

"Computer simulation refers to methods for studying a wide variety of models of real-world systems by numerical evaluation using software designed to imitate the system's characteristics, often over time" (KeltonLaw, et al, 2007). Essentially, simulation is the process of developing a computer model that represents a real-world system as it changes over time. Banks and Carson (1984) define a system as "a group of objects that are joined together in some regular interaction or interdependence toward the accomplishment of some purpose." A system can be either discrete or continuous. A discrete system is one where the dependent variables change only at separate points in time. Contrarily, in continuous systems the variables change continuously over time

(Banks and Carson, 1984). This dissertation is focused on discrete systems and consequently looks at discrete-event simulation.

Simulation uses “a computer to evaluate a model numerically, and data are gathered in order to estimate the desired true characteristics of the model” (Law and Kelton, 2000). Simulation has the distinct advantage of being able to evaluate a system without having to build the system, without disrupting the system, and without destroying the system (Pritsker and O’Reilly, 1999).

Pritsker and O’Reilly (1999) define six steps that should be used in developing and simulating a model of a system. These steps are:

1. formulate the problem,
2. specify the model,
3. build the model,
    a. develop the simulation model
    b. collect data
    c. define the experimental controls
4. simulate the model,
    a. run the model
    b. verify the model
    c. validate the model
5. use the model, and
6. support decision making.

Traditionally, simulation is used to evaluate various system configurations, from which the configuration that provided the best performance would be chosen. The main disadvantage of this approach is that generally only a select number of configurations are considered. The lack of an optimization technique included in the simulation, thus really limits the potential effectiveness of the simulation (Law and McComas 2002).

2.1.2 Optimization

Optimization is the means of finding the best solution from a series of alternatives without having to explicitly enumerate and evaluate each possible solution (Reklaitis et al, 1983). Essentially, the goal is to find a set of input parameters that generate an optimal output solution, i.e., a solution that maximizes or minimizes a specified objective. The optimization is modeled through a mathematical formulation and some form of mathematical results and numerical techniques guide an algorithm to identify the optimum solution.

Optimization, however, has difficulty handling many real-world problems that often involve non-linear constraints or objective functions, combinatorial relationships, or uncertainties. These types of problems are too complex for traditional mathematical formulations. These types of problems, however, are well suited to be addressed by simulation. However, simulation is not in itself an optimization tool. Therefore, it becomes desirable to combine simulation and optimization to address these problems (April et al, 2004).

Simulation optimization is particularly complicated, because with the stochastic nature of the systems, evaluation of particular designs can only be estimated. Thus, it is difficult to conclusively determine the best alternative (Banks et al, 2000).

2.2 Multi-Criteria Optimization

Multi-criteria optimization (also referred to as multiple criteria decision making – MCDM) is a technique used in an attempt to obtain an optimal solution to a problem that has multiple and usually conflicting objectives (e.g., minimize costs and maximize quality). Thus, improving one objective can often be accomplished only at the expense of another objective. So, frequently, it may not be possible to find a point that optimizes all criteria individually. Goal programming is one approach for solving these problems. Indeed goal programming is one of the oldest and most important methods for solving multiple criteria decision making problems (Hanne, 2001).

2.2.1 Pre-emptive Goal Programming

Goal programming uses standard linear programming techniques to solve problems with conflicting objectives by assigning priorities to the objectives. The highest objectives will have the highest priority. Goal programming seeks to minimize the deviations from goals instead of strictly trying to optimize the objectives (Lee and Morris, 1977).

Goal programming requires that a decision-maker set target values or goals on all criteria. These goals are not absolute constraints but are simply desirable values for the various criteria. It is possible to have absolute (real) constraints that must be satisfied in addition to the goals.

A goal is optimized when the target value of the goal is met. Alternatively, if it is not possible to achieve the target value, a goal is optimized when the best possible value is found (that is the value closest to the target value). Thus, it is possible to optimize a goal without achieving or satisfying the goal. The state of the goal is the current function value of the goal or the fact that the goal is achieved. Thus, it is only possible to improve the state of the goal if the goal is not achieved (since all values that achieve the goal are considered equivalent). The state of the goal can become worse if an achieved goal is no longer achieved or if the current function value moves further from the target value.

The decision-maker ranks the different goals in order of importance. The goals are then satisfied in that order. Thus, it is most important to satisfy the first goal. Once this goal is optimized, the second goal is optimized such that the state of the first goal does not become worse. This process continues for all goals. Goal programming seeks to satisfy all of the goals, if possible, or seeks to satisfy as many goals as possible in order of importance, hence the name pre-emptive goal programming. This discussion on goal programming is based on the discussion by Ravindran, Phillips, and Solberg (1987).

If more than one point exists that optimizes a goal, the goal is said to have alternative optimal solutions. Thus, the only way to improve the state of a lower priority goal is if alternative optima exist for the higher priority goal. The success of goal programming relies on goals having alternative optima. Remember that the decision maker sets the level of achievement for each goal. Thus, if the target for a goal is relaxed, theoretically there will be more alternative optimal solutions for that goal. It is also important to note that if all of the goals of a problem are satisfied, then the optimal solution most likely occurs at a dominated point. That is, there exists a point that will improve all of the goals, or at least the higher priority goals. If this case occurs, it is often desirable to raise the target values on one or more of the goals, so that the overall achievement of the problem can be improved.

2.2.2 Partitioning

One method to solve goal programming problems is partitioning. The partitioning concept, as described by Saber and Ravindran (1995), is a method of setting up a series of subproblems. The first subproblem attempts to satisfy the highest priority only. If this is accomplished, the second subproblem begins. The second subproblem attempts to optimize priority two, without violating priority one. This series of steps continues until it is not possible to satisfy the latest goal or until all the goals have been satisfied. The main advantage of partitioning is that it allows a complicated problem to be partitioned into several smaller and simpler subproblems.

2.2.3 Lexicographic Ordering

Lexicographic ordering is decided by the first non-equal term (A_i compared with B_i) in a pair of vectors (see Figure 2.1). Thus, vector A is lexicographically less than vector B if $A_1 = B_1$ and $A_2 < B_2$, regardless of A_3 and B_3.

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \cdot \\ \cdot \\ \cdot \\ A_i \end{bmatrix} \prec \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \cdot \\ \cdot \\ \cdot \\ B_i \end{bmatrix}$$

Figure 2.1: Lexicographic Comparison of the Vectors A and B.

2.2.4 Integer Goal Programming

In many realistic problems it is necessary to ensure that one or more of the variables have an integer value. If this type of problem has multiple objectives, one common approach to solving it is using integer goal programming. If all the variables are required to be integers, then the problem is a pure integer goal programming problem. Similarly, mixed integer goal programming occurs when some of the variables are restricted to be integers and other variables are not.

An example of a mixed integer program would be in a restaurant. The restaurant would like to purchase tomatoes, beef, and bread. The tomatoes and the beef are each measured in pounds, so it is possible to get a non-integer value for these items. Bread, however, is purchased by the loaf. Thus, its corresponding variable must be an integer (0,1,2,...), since it is not possible to buy partial loaves of bread. The restaurant has three objectives. First, they want to minimize the money spent to purchase the items. Second, they want to maximize the amount of each item available in storage. And third, they want to maximize the freshness of all the items.

This is clearly a good problem to solve using mixed integer goal programming. There are both integer and non-integer variables, there are three different and conflicting objectives, and the objectives have been prioritized. There are several different methods which can be used to solve the mixed-integer goal programming problem. Traditionally, the problem was just solved ignoring the integer restrictions, and then rounding or truncating the values to get integer values. This works fairly well for large value problems, but for small value problems this can have a huge impact.

To get the best integer solution it is necessary to do some form of multi-criteria integer programming. Two methods of integer goal programming will be discussed. They are the branch and bound method, and the branch and bound method with constraint partitioning.

The basic concept of the branch and bound method is to bound the decision variable with an upper and lower bound (Lee and Morris, 1977; and Schniederjans, 1984). Then two subproblems branch off of the main problem (one for the upper and one for the lower bound). The corresponding upper or lower bound generates a new constraint for the problem. A variable is bounded only if it is an integer restricted variable whose value is not an integer. Each branch can produce two more branches. The problem is finished when all branches have been terminated. Termination can occur if the appropriate variables are integers or if the subproblem at that branch is infeasible.

The branch and bound method with constraint partitioning is proposed by Arthur and Ravindran (1980a). The method combines constraint partitioning with traditional branch and bound methodology. The concept of constraint partitioning for linear goal programming was presented by Arthur and Ravindran (1980b). The idea is to break the goal programming problem into a series of subproblems. The first subproblem only considers priority 1 constraints, the second adds in the priority 2 constraints. This process continues until the optimal solution is met, or all the priorities have been considered. After solving a subproblem, the optimal tableau is examined to see if alternate optimal solutions exist. If they do, the next subproblem is considered. If there are no alternate optimal solutions, then it is not possible to improve the lower priorities, so the problem is completed (Arthur and Ravindran, 1980a).

Heuristic approaches like tabu search and genetic algorithms can also be used for problems containing integer variables. These methodologies are essentially the same as presented earlier. See Wolsey (1998) for a look at the specific algorithms.

2.3 Simulation optimization

Olafsson and Kim (2002) define simulation optimization, also frequently referred to as optimization via simulation, as "the process of finding the best values of some decision variables for a system where the performance is evaluated based on the output of a simulation model of this system." Carson and Maria (1997) define it as "the process of finding the best input variable values from among all possibilities without explicitly evaluating each possibility." Simulation optimization essentially combines simulation with an optimization technique.

Since an optimization technique is used, simulation optimization can be used to determine the system configuration (i.e., the set of input parameters) to provide a near-optimal if not an optimal solution (Law and McComas, 2002). Thus, it seeks to find more than simply the best configuration from a number of pre-selected scenarios. The main advantages of this approach are that: (1) it is not restricted to a set number of scenarios, (2) it is not restricted to only evaluating certain levels of specific input parameters, (3) it requires less operator interaction and decision making, and (4) it may not be necessary to evaluate all levels of input parameters to determine the near-optimal solution.

The optimization method interacts with the simulation and progresses the system until a stopping rule has been satisfied. The basic procedure can be defined as follows:

Step 1: Determine the initial levels of the decision variables, X_o.

Step 2: Run the simulation for the current input levels and get the corresponding output.

Step 3: Feed the output into the optimization technique and generate a new X.

Step 4: Has the stopping criteria been satisfied? If yes, stop and report the optimal solution obtained. If not, update the decision variables and go to Step 2.

In order to solve the problem, it first must be setup as a classic mathematical programming problem. The objective function (F) and/or one or more of the constraints (H_i) can involve stochastic effects. If so, the expected values of F and H_i are determined by running the simulation for a given **X** vector. Otherwise, they are determined through traditional means.

The general problem formulation used for this study is derived from a combination of several notations used in the literature. Particularly this notation is a combination of notations used by Law and Kelton (2000) and Tekin and Sabuncuoglu (2004). Thus, a simulation optimization programming problem can be stated as:

Minimize:	$E[F(\mathbf{X},\omega)]$		
Subject to:	$E[H_i(\mathbf{X},\omega)] \leq c_i$		$i=1,\ldots,m$
	$X_j^l \leq X_j \leq X_j^u$,	$X_j \in \Re$,	$j=1,\ldots,n$

where:

F is the objective function,

H_i are the constraints,

X is a vector containing n design variables,

ω represents the stochastic effects,

m are the number of constraints with the expected values of H_i,

n are the number of variables,

X_j^l and X_j^u are the lower and upper bounds of X_j, respectively, and

$\Re$ is the set of real numbers.

Further discussion on simulation optimization formulation can be found in Law and Kelton (2000).

There is much research in the area of simulation optimization, with many different optimization techniques used to address the optimization part of the process. Carson and Maria (1997) define six major categories for simulation optimization methods. These methods are: gradient based search methods, stochastic optimization, response surface methodology, statistical methods, heuristic methods, and A-team (asynchronous team). Tekin and Sabuncuoglu (2004) divide simulation optimization into local optimization and global optimization. Essentially, the first three methods mentioned above fall into the category of local optimization for continuous decision space, while the forth method falls into local optimization for discrete decision space. Heuristic methods fall into the global optimization category. A-Team works by combining various methods and thus can have characteristics of both categories.

This dissertation focuses on three heuristic methods (often referred to as metaheuristics). These are genetic algorithms, tabu search, and lexicographic Nelder-Mead search (which falls into the category of direct search methods). These methods will be discussed in detail later in this chapter. For a complete introduction to genetic algorithms, tabu search, and other typical simulation optimization methods, see Tekin and

Sabuncuoglu (2004), Olafsson and Kim (2002), Fu (2001), Swisher et al (2000), Azadivar (1999), Andradottir (1998), Carson and Maria (1997), Fu (1994a), and Safizadeh (1990).

Guimaraes and Kingsman (1989) combine a direct search routine based on the Powell-DSC method with simulation. This simulation optimization method was applied to the analysis of grain terminal operations. Henderson and Mason (1998) propose a new technique, called Rostering by Iterating Integer Programming and Simulation (RIIPS), to solve rostering problems when uncertainty is present.

Curry and Hartfiel (1983) propose an iterative optimization-simulation procedure that sacrifices the optimization performance to reduce computational effort. Barton and Ivey Jr. (1996) made modifications to the Nelder-Mead method to overcome the potential risk of an inappropriate termination when Nelder-Mead is used to solve a simulation optimization problem. These modifications were made to reduce the bias of contraction of shrinkage. It is this method that was modified to create the lexicographic Nelder-Mead search. These modifications were added to allow the method to be able to handle multiple objectives. This method will be discussed in more detail later in the chapter.

A specific simulation optimization method that uses a series of linear approximate optimization subproblems was proposed by Abspoel, et al (2001). This method is based on response surface modeling (RSM). Essentially, an integer programming problem may have nonlinear sections. To account for this, the method examines a series of linear response surface approximate subproblems. These subproblems are solved with an integer programming (IP) solver. The paper explains the corresponding method in detail.

Nelson and Staum (2006) derive a ranking and selection procedure that, instead of sample means, uses control-variate estimators. Hong and Nelson (2006) develop a simulation optimization algorithm with integer ordered decision variables. The method, called COMPASS, is proven to converge, with a probability of 1, to the set of local optimal solutions for terminating and steady-state simulation under mild conditions. Hong and Nelson (2005) build off their previous work and provide a general framework that uses locally convergent random search algorithms to solve discrete simulation optimization problems. The COMPASS algorithm is revised to only allocate simulation observations to a selected subset of visited solutions. This reduces the noise level in the function evaluations and increases the probability of finding a local optimal solution.

Pichitlamken and Nelson (2003) propose a combination of several methods to solve simulation optimization problems. They use the Nested Partition (NP) method as their global guidance system, they use Sequential Selection with Memory (SSM), outlined in Pichitlamken et al (2004), to select the best solution from among a list of candidates, and they use the hill-climbing (HC) algorithm as their local-improvement scheme. Their goal was to develop a method, called NP+SSM+HC, that has both provable convergence and good empirical performance.

2.3.1 Heuristic Search Methods

Heuristic methods involve some form of trial and error to solve problems that would be difficult to solve using traditional algorithms (Winston, 2004). The use of heuristic methods has many advantages and disadvantages. Boesel et al (2001) summarize these by stating that "although they lack desirable convergence properties, heuristic search algorithms provide good, reasonably fast, results on a wide variety of problems (Boesel et al, 2001)." Three of the most common heuristic search methods are simulated annealing, genetic algorithms, and tabu search.

Azadivar and Lee (1998) propose a heuristic approach, SIMICOM, to solve the discrete simulation optimization problem. The method is based on Box's complex search method. The method performs well in comparison with a couple of other methods for several test problems. Mason et al (1998) develop an integrated approach that uses simulation, heuristic descent, and integer programming to successfully solve personnel scheduling of customs staff at a New Zealand airport.

2.3.1.1 Simulated Annealing

Simulated annealing algorithms mimic the thermodynamic annealing process. Essentially, the algorithm works by searching neighboring feasible solutions and allowing movement to inferior neighbors based on an acceptance function. Points are evaluated based on their objective function value, but moves that make the objective function value worse are allowed based on a randomized scheme. The randomized scheme is based on a temperature parameter that starts off at a high level and then decreases over time, creating a cooling curve. In this way, the algorithm attempts to avoid converging early in the search to avoid getting caught in a potentially poor local optimum (Avello et al, 2004; Reeves, 1996). For a more detailed description of simulated annealing see Eglese (1990) and Fleischer (1995).

2.3.1.2 Genetic Algorithms

Genetic algorithms use a collection of solutions to mimic the evolutionary process to find better solutions (Rardin, 1998). Recombination and selective breeding strategies are used within a collection of solutions, or the population, to produce better solutions Crossover and mutation operators are used to construct new solutions from old solutions. (Reeves, 1996). Chambers (1995) defines a genetic algorithm as “a mathematical method using genetics as its model and applying the rules of reproduction, gene crossover, and mutation to assess a system's performance.”

New generations can be produced through several means. Frequently, the best solution or solutions from the previous generation are added to the current population. This is referred to as elitism. Another means is through immigrant solutions. For this case, solutions are randomly passed from one generation to the next. This is done to increase the diversity of the population.

Crossover is the combining of two “parent” solutions to form two “children.” This is accomplished by splitting both solution vectors at the same point and then replacing everything before the split line of one solution vector with everything after the split line of the other solution vector and vice versa (Rardin, 1998). The case where the crossover occurs after the second component is illustrated below:

Parent 1 = (1, 0,|0, 1, 0)   Parent 2 = (1, 1,|0, 0, 1)

Child 1 = (1, 0, 0, 0, 1)   Child 2 = (1, 1, 0, 1, 0).

The example above is referred to as single-point crossover, since the crossover occurs at only one point. More complex crossover possibilities are also possible. Two-point crossover has two crossover split lines. It is even possible to have multi-point crossover, where multiple split lines are used (Reeves, 1996).

Another operator used to populate a new generation is mutation. Mutation works by changing each individual component of the solution vector based on a small probability. The desire of mutation is to allow for new areas of the solution space to be examined. In the case where the second component is mutated, the original solution vector (1, 0, 0, 1, 0) would become (1, 1, 0, 1, 0) (Reeves, 1996).

The selection of the "parents" for mating is based on a fitness score. The fitness score can be a linear scaling of the objective function values or it could be a different scaling procedure (Wall, 1996). The specific mechanism for choosing selection is usually done using one of several different methods. One such method is the tournament selection, where groups of solution vectors of a predetermined size are chosen for comparison and the solution vector that has the best fitness of the group is selected (Reeves, 1996). Additional methods used include "Roulette wheel" selection, random selection, and stochastic sampling (Adcock, 2006)

Genetic algorithms, simulated annealing, and tabu search simulation search procedures were developed and used to find the optimum number of kanbans for a just-in-time manufacturing system. The three methods were compared to each other, as well as to a random search implementation. All the approaches found the same optimal solution, but tabu search found the solution while searching the fewest number of solutions.

2.3.1.3 Tabu Search

Tabu search is a neighborhood search technique that uses memory to aid the search and reduce the possibility of locking onto local optima. Essentially, tabu search examines the neighbors of the current solution and chooses the best available of these solutions to move to and continue the search. Certain neighbors may not be available if they are on a tabu list. The tabu list consists of points recently visited. After a set number of iterations, a point is removed from the tabu list and thus can again be selected. Additional tabu criteria can also be established. One common approach is to try to limit certain directional moves. Although the roots of tabu search go back to the 1970's, its current form was first presented by Glover in 1986 (Hertz et al, 1997).

The first step of tabu search is to identify an initial solution and then determine the neighborhood of potential solutions. The definition of the neighborhood, the potential moves, is defined at each iteration. The idea is to examine points that are in close proximity to the current solution. The actual way the neighbors are calculated is problem-specific (Cavin et al, 2004).

One objective of tabu search is to explore areas of the solution space which have not previously been visited (Reeves, 1996). This is accomplished by making certain moves forbidden or 'tabu.' Specifically, it is desirable to prevent the reversal of previous moves that might cause the search to get caught at a local optimum. After a specified number of iterations or another condition has been satisfied, these moves or visited points are taken off the tabu list enabling them to potentially be re-visited.

An additional element of tabu search is the ability to override tabu moves if certain aspiration levels are attained. One common aspiration condition used in many applications allows a tabu move to occur if the move provides a better objective function value than the current best value. Thus, tabu restrictions and aspiration level criteria form a complimentary role that guides and constrains the search process (Glover, 1989).

In addition to short-term memory, tabu search can also use intermediate and long-term memory to either intensify or diversify the search. Intensification uses intermediate-term memory and is accomplished by evaluating a set of best solutions to identify common characteristics between these solutions. Then, the search is encouraged to examine solutions that have these or similar characteristics. This is accomplished by temporarily changing the objective function to penalize moves that differ greatly from these identified characteristics (Glover, 1989).

Diversification uses long-term memory to essentially do the reverse of intensification. The goal is to examine solutions that have different characteristics to the solutions previously visited. This can be accomplished by altering the objection function to penalize moves that have similar characteristics to previously visited solutions. The purpose of diversification is to drive the search to new areas of the solution space (Glover, 1989). A complete discussion on tabu search and all of its components can be found in Glover and Laguna (1997).

Dengiz and Alabas (2000) use a tabu search algorithm with a simulation model of a just-in-time (JIT) system to find the optimal number of kanbans. They compare tabu search to random search and find that tabu search performs better. Yang et al (2004) present a tabu search simulation optimization method that is used to solve a flow shop with multiple processors scheduling sequence problem.

Konak and Kulturel-Konak (2005) examine different ways of performing simulation within a simulation optimization algorithm that uses tabu search. The authors found that for their example problem, making adjustments to tabu search procedures can have a large impact on the effectiveness of the algorithm without having to increase the number of simulation replications.

2.3.2 Direct Search Methods

Direct search methods are techniques to solve nonlinear programming (NLP) problems that are also based on heuristics. These techniques only require function values in order to iteratively move towards an optimal solution. Unconstrained search methods are methods for solving problems with no constraints, while constrained search methods solve constrained optimization problems (Reklaitis et al., 1983). There are several direct search methods. One method may perform well for one problem but not for another problem. The methods require that a starting point be specified for each problem, and the result is highly dependent on this starting point (Ignizio, 1976). These methods do not guarantee convergence to an optimal solution, but they ideally will converge to at least a local minimum. In general, these methods are computationally uncomplicated. However, they are often slower than other techniques used to solve NLP problems (Reklaitis et al., 1983).

2.3.2.1 Simplex Search

Spendley, Hext, and Himsworth (1962) presented the simplex search method. The basic concept, as presented by Reklaitis et al. (1983), is that a set of points (number of dimensions plus one) is placed equidistant from a starting point, thus forming an initial simplex (the set of vertices). The vertex with the worst function value of this simplex is projected through the centroid of the remaining vertices to establish a new simplex. This procedure continues until the simplex has shrunk so far that the points of the simplex are sufficiently close or until the standard deviation of the function values of the vertices becomes smaller than a specified value.

2.3.2.2 Nelder-Mead Search and Revised Nelder-Mead Search

Nelder and Mead (1965) developed a Revised Simplex method. The Nelder-Mead method, as presented by Reklaitis et al. (1983), was developed to account for the problems of the Simplex Search. These problems included: scaling problems, slow movement, and no means of expansion without recalculating the entire pattern. The Nelder-Mead method allows for expansion and contraction during reflection. The method also considers the highest function value point ($X^{(h)}$), the next highest function value point ($X^{(g)}$), and the point with the lowest function value ($X^{(l)}$). After these points are determined, a normal reflection of the simplex is carried out. If the method does not terminate at this point, a normal reflection, expansion, or contraction is carried out. To determine which of these operations is performed, the function value of the new point that was generated through normal reflection is compared to the function values of $X^{(h)}$, $X^{(g)}$, and $X^{(l)}$.

Barton and Ivey Jr. (1996) made modifications to the Nelder-Mead method. These modifications were made to reduce the bias of contraction of shrinkage. These modifications are: the shrink coefficient is increased from 0.5 to 0.9 (this does not reduce the simplex as much), the best point after a contraction step is re-evaluated before the next reflection, and the vertices after reflection are re-sampled before contracting to reduce the probability of attempting a contraction.

2.3.2.3 Lexicographic Nelder-Mead Search

Kuriger and Ravindran (2005) and Kuriger (1998), propose three methods to solve multi-criteria optimization problems. These methods used partitioning and lexicographic concepts to extend the Nelder-Mead, Complex Search, and Hooke and Jeeves Pattern Search methods to handle multiple criteria. All three methods performed well.

This research will use one of the methods, the Lexicographic Nelder-Mead (LNM) method, that was proposed in Kuriger and Ravindran (2005). The LNM method is based on the Revised Nelder-Mead method discussed in the previous section. The LNM method essentially seeks to optimize the multi-objective problem by lexicographically addressing the goal constraints. That is, it attempts to solve goal 1 first, then goal 2, and so on. A move will be accepted if it improves any goal without worsening a higher priority goal.

2.4 Multi-Criteria Simulation Optimization

Real problems often require that several objectives should be optimized. Multi-criteria problems involving conflicting objectives are difficult in themselves to solve. Combining multi-criteria optimization with simulation optimization adds the extra complexity of stochastic components into the problem (Tekin and Sabuncuoglu, 2004). According to Baesler and Sepulveda (2000), there have only been a few methods used to address multi-response simulation optimization and "the majority of them are focused on response surface methodology, utility theory and interactive processes where the decision maker interacts with the model and leads the search" One way to attempt to solve these problems is through goal programming. All of the problems examined in this study only include goal constraints; and upper and lower bound constraints. A general problem formulation for a simulation optimization goal programming problem can be stated as:

$$\text{Minimize} \qquad Z = \sum_{i=1}^{g} P_i d_i^+$$

$$\text{Subject to:} \qquad E[G_i(\mathbf{X},\omega)] + d_i^- - d_i^+ = T_i \qquad i = 1,\ldots,g$$

$$X_j^l \leq X_j \leq X_j^u, \qquad X_j \varepsilon R, \qquad j = 1,\ldots,n$$

$$d_i^-, d_i^+ \geq 0$$

$$\text{with } P_1 \gg P_2 \gg \ldots \gg P_g,$$

where:

Z is the objective function to be minimized;

P_i are the priority penalties;

$G_i(\mathbf{X})$ are the nonlinear goals;

d_i^- and d_i^+ are non-negative deviational variables;

T_i are the targets (note that target values can be positive or negative as needed to account for min or max characteristics);

g is the number of goals;

n is the number of variables;

$\mathbf{X}$ is vector of input variables composed of X_j, ($j = 1 \ldots n$);

ω represents the stochastic effects,

X_j^l and X_j^u are the lower and upper bounds of X_j, respectively.

Evans et al (1991) describe some issues unique to multi-criteria simulation optimization. These include: the relationship between the input and output variables are not of a closed form, the output variables maybe random or stochastic variables, and many local optima may be present on the response surface. The authors propose a framework for solving multi-criteria simulation optimization problems.

Mollaghesemi and Evans (1994) propose a simulation optimization method to solve multi-criteria problems that is a modification of the STEP method (introduced by Benayoun et al, 1971). The method is interactive, that is, the decision maker is asked to provide input after each iteration to guide the algorithm. An application of the algorithm was done for a job shop problem with six groups of machines. The problem considered six decision variables and four objectives.

Teleb and Azadivar (1994) propose an interactive algorithm to solve multi-criteria simulation optimization problems. The method is a modification of the complex search method. The method assumes that the stochastic objectives and constraints follow a normal distribution.

Boyle and Shin (1996) propose a new method, the Pairwise Comparison Stochastic Cutting Plane (PCSCP) method, for optimizing a stochastic computer simulation with multiple response variables. The method combines interactive multiple objective mathematical programming techniques with response surface methodology.

Szidarovszky and Eskandari (1999) use a combination of discrete multiple criterion decision making (MCDM) methodology and stochastic simulation to analyze a forest treatment problem with multiple criteria. Evans and Anderson (2006) present a discussion of criterion models used in simulation studies and a description of methodologies for selecting criterion model attributes.

Anderson et al (2006) interfaces simulation with criterion models to evaluate multiple performance measures for a logistics system. The proposed algorithm is based on the Scatter Search algorithm and the Simplex Method. The authors define utility functions for three different risk levels (risk-averse, risk-neutral, and risk-seeking) that allow the decision maker to select the level with which they are most comfortable. The authors present a graphical user interface that is used to assist the decision maker. The problem is solved and the results are presented for each risk level. Additionally, the authors attempt to improve the efficiency of the optimization process by applying variance reduction techniques, specifically, common random numbers and antithetic variates.

Avello et al (2004) present a simulation optimization method for multiple criteria that is based on simulated annealing. This method, called the Parallel Time-Space Phase Equilibrium Simulated Annealing (ParT-SPEq-SimAnn) method, incorporates alterations to traditional simulated annealing algorithms to allow for multiple objectives. This is accomplished by incorporating a global cooling curve and a particular cooling curve for each objective.

Clayton et al (1982) combine a modified pattern search and pre-emptive goal programming to solve multiple-response simulation models. Rees et al (1985) combines response surface methodology and lexicographic goal programming to solve multiple response simulation models.

Baesler and Sepulveda (2000) present an approach that combines simulation, genetic algorithms, and goal programming to solve multi-criteria simulation optimization problems. They modified the search procedure to account for both the mean and the variance of the system. This modification allows the search to be performed stochastically as opposed to deterministically like most approaches. This model was implemented on a case examining a cancer treatment center facility (Baesler and Sepulveda, 2001). The problem involved four objectives and four control variables. The authors report an improvement of the system over the actual set-up of the center.

Al-Aomar (2002) presents a multi-criteria simulation optimization methodology that integrates simulation modeling, genetic algorithms, a robustness model, and an entropy method. The robustness model incorporates both the mean and the variance into a single fitness value that is used to direct the GA. The entropy method is used to set the weights used to evaluate the multiple objectives through a multi-attribute utility function that is updated at each iteration of the GA.

Molnar (2005) implements a multi-criteria simulation optimization method using genetic algorithms to find the number of order pickers per shift and the sequence of retrieval of the pick lists to optimize three criteria. The method was designed to assist operative warehouse management in an automotive parts warehouse. The three objectives are combined into a single objective function by summing each separately and multiplying each by a weight set by the decision maker.

Eskandari et al (2005) propose an enhancement to genetic algorithms based on a new ranking strategy. This enhancement is then integrated with a simulation model and a stochastic nondomination-based multi-criteria optimization method. The authors use an elitism operator to ensure that the Pareto-optimal set is passed to the next generation. Additionally, an expansion operator is used to increase the population size as needed up to a user-specified maximum. Finally, the authors propose the use of an importation operator to explore new regions of the search space.

2.5 Evaluation of Performance

There are relatively few methods outlined for the evaluation of performance of simulation optimization methods. A paper by Keys et al (2002) outlines six such measures. The authors suggest two absolute performance measures and four relative performance measures. These measures are used to evaluate the accuracy of placing optima in the current location, the fit to the response, and the fit to the character of the surface. These measures assume that the function used to define the objective function is known, which for many simulation optimization problems is not the case.

2.5.1 Algorithm Analysis

"Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output" (Cormen, et al, 2001). It is important to evaluate the efficiency of algorithms and this can be done through algorithm analysis. This is essentially the science of predicting the resources, primarily time, required to complete the algorithm. It is desirable to eliminate any uncontrolled variables such as the particular computer, memory, processor, and computer language used to actually implement the algorithm. In this way, it is possible to compare algorithms regardless of the variability of these uncontrolled variables (Cormen, et al, 2001).

A complete algorithm can be described as a computational procedure that given an input value will terminate with a correct answer in finite time (Cormen, et al, 2001).

Due to the stochastic nature of simulation optimization it is not possible to declare a solution as optimal. Additionally, if the problem was run multiple times, multiple results would be generated. Thus, for a given input value, it is not possible to always terminate with a correct answer or for that matter even to declare any answer as correct. Thus, simulation optimization methods in general do not satisfy the requirement that a complete algorithm must terminate with a correct answer. Additionally, the methods used in this research are heuristic in nature. They do not guarantee that an optimal solution will be found even for problems that do not involve stochastic variables. Therefore, the simulation optimization methods examined in this study more correctly fall under the category of heuristic-based simulation optimization methods.

2.5.2 Evaluation of Heuristics

The evaluation of heuristic methods is in general best accomplished through empirical evaluation, that is, through the use of computational experiments. “The choice of performance measures for experiments on heuristics necessarily involves both solution quality and computation time (Rardin and Uzsoy, 2001).” Lin and Rardin (1980) provide guidelines for testing and comparing integer programming algorithms. Rardin and Uzsoy (2001) present an overview of the development, implementation, analysis, and reporting for the experimental evaluation of heuristic algorithms.

Barr et al (1995) provides an overview of the design and reporting of experimental studies of heuristic algorithms. The authors identify six general performance measures for testing heuristics. These are:

1. The quality of the best solution found
2. The time required to determine the best solution

3. The time needed to find good solutions
4. The robustness
5. The distance the best solution is from those more easily found
6. The tradeoff between feasibility and solution quality.

The use of a good experimental design is crucial to the effectiveness of the evaluation of heuristics. According to Montgomery (1991), "a designed experiment is a test or series of tests in which purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes in the output response." Full-factorial design, analysis of variance (ANOVA), and Duncan's multiple range tests can be used to evaluate the performance of algorithms (Rardin and Uzsoy, 2001). For a detailed breakdown of these and other experimental design procedures, see Montgomery (1991).

Hooker (1994) identifies the need to develop an empirical science of algorithms that will allow for the comparison and evaluation of heuristic algorithms. The author suggests that instead of characterizing the performance of an algorithm for a class of problems, it is possible to examine the performance of an algorithm based on problem characteristics. Essentially the focus is switched from examining an algorithm for a specific problem to simply examining the algorithm in general. McGeoch (1996b) describes these two contrasting approaches as "application-centered" and "algorithm-centered."

Hooker (1995) similarly calls for a more scientific evaluation of heuristic algorithms, termed controlled experimentation. The idea is to evaluate the affect of certain characteristics on the performance of the algorithm. The affects can have a positive or negative impact, but the important concept is that the results can show important insights into the mechanics of the algorithm. This is contrasted with the competitive nature approach where results are only published or presented if they outperform some existing method and does little to further the understanding of the performance of algorithms.

McGeoch (1996a) outlines a different approach to computational experiments of algorithms called algorithm simulation. The idea is to simulate the behavior of an algorithmic model to determine the particular characteristics of the algorithm that impact its effectiveness. In this way, algorithm simulation can be used to obtain insights into the algorithm that can lead to better models and better predictions of program performance. McGeoch (1996) also stresses the importance of a pilot study, whose results can identify areas and parameters that should be evaluated more thoroughly in the formal study.

The use of central processing unit (CPU) time as a performance measure must be used with caution. Orlin (1996) states that "papers that treat CPU time tend to focus on which algorithm is the 'winner' rather than focus on the strengths and weaknesses of different algorithms." CPU time is implementation dependent and difficult to replicate (Ahuja and Orlin, 1996). Additionally, CPU time often does not translate from one computer (or operating system) to another, thus different measures are desired to compare computational effort (Barr et al, 1995).

Ahuja and Orlin (1996) propose the use of representative operation counts in addition to CPU time to better compare computational effort between algorithms. They propose identifying the aspects of the algorithm that require a large amount of computation time and counting the number of times these bottleneck operations occur. This will allow for comparisons to be made between algorithms run on different computers and using different computer languages. Additionally, it will provide better insights into the operation of algorithms. An example of a bottleneck operation for simulation optimization would be the number of calls to the simulation program.

2.5.3 Evaluation of Heuristic-Based Simulation Optimization

There are currently no established evaluation criteria for heuristic-based simulation optimization methods. Thus, individual researchers develop their own criteria to analyze their results. Two common approaches involve either the evaluation of the heuristics or the evaluation of simulation models. It is also possible to perform evaluation based on some combination of these two evaluation procedures.

Thus, there is a need to establish an evaluation procedure or a set of tools to determine the effectiveness of a simulation optimization method and to allow for accurate comparison with other simulation optimization methods. This research will add to the field of the evaluation of heuristic-based optimization by introducing several global performance measures. These performance measures will allow for better comparison between simulation optimization methods in terms of computer efficiency, quality of solution, and the overall evaluation of performance.

Chapter 3

Methodology

This chapter provides an overview of the methodology used in this research and includes a description of the specific multi-criteria simulation optimization methods that are used to solve five test bed problems. These problems will be explained in detail in the next chapter. Additionally, this section describes the performance measures that are used to analyze the methods and the test bed problems.

3.1 Multi-criteria Simulation Optimization Methods

The multi-criteria simulation optimization method is used to determine the best combination of input values that provides a near-optimal solution to the multi-objective problem. The simulation optimization methods used in this study function by the interaction of an optimization program and a simulation model to solve multi-criteria simulation optimization problems. The multi-criteria component of the procedure is primarily handled in the formulation of the problem. This formulation is then incorporated into the optimization program. A summary of the flow of the simulation optimization method is shown in Figure 3.1. A further breakdown of the multi-criteria component, the optimization program, and the simulation model is presented in the following sections.

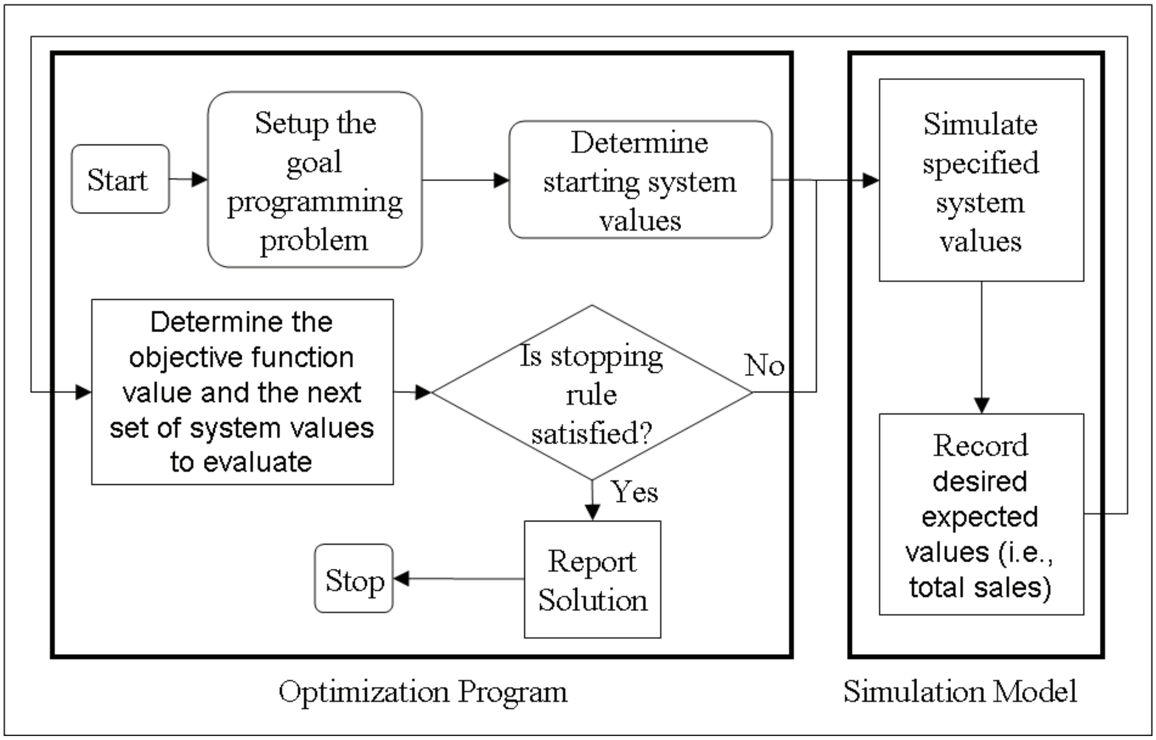


Figure 3.1: Flowchart of the Simulation Optimization Methods

3.1.1 Description of the Multi-Criteria Component of the Methodology

This research focused on problems that involved multiple objectives. Pre-emptive goal programming was the approach that was used to address these multiple criteria. Recall that for pre-emptive goal programming, it is assumed that the decision-maker wants to satisfy higher priority goals much more than lower priority goals. The decision-maker must rank the goals in order of decreasing importance (i.e., it is much more desirable to satisfy goal one than goal two, goal two than goal three, and so on). Thus, for a problem with g goals, this can be expressed as:

$$G_1 \gg G_2 \gg \ldots \gg G_{g-1} \gg G_g, \tag{3.1}$$

The goals for each problem would need to be determined by the decision-maker. It was assumed that each goal was independent of the other goals for each problem. Thus, it is important that the goals be setup properly to account for this. The issue of independence is very important for the statistical analysis of the results. If the goals are independent of each other, then the resulting variance for the objective function of the problem is simply the summation of the variance of each of the goals.

Additionally, the decision-maker would need to set target values for each goal. If a goal satisfies its target value, then it has been achieved. The only way to getter a lower value for this goal (for minimization) is to decrease its target value. Thus, if a solution exists that satisfies all the goals of a problem, it is possible that this is a dominated solution (i.e., there exists a solution for which all goals have equal or improved values, and at least one goal could be improved if the target values were adjusted). In this case, the target values were too relaxed. Conversely, for the case of conflicting objectives, the only way to improve the state of a lower priority goal is if alternate optima exist for higher priority goals. Thus, it is important that the target values are not set too tight.

Setting the target values for a problem is consequently, very important. The goals for each of the test problems used in this study were solved individually for one run to determine the near-optimal value for each goal. The target value for each goal was initially set by taking eighty-five percent of the corresponding near-optimal value. Test runs were then performed and the target values for each goal were adjusted as necessary to ensure the problems were functioning as desired. The overall objective of setting the target values was to ensure that it was not possible to satisfy all the goals. This was done to greatly increase the chances of selecting a non-dominated solution as the near-optimal or best solution.

All problems were setup as minimization problems. If a goal was to be maximized, it was simply converted into a minimization problem by multiplying the goal and the corresponding target value by negative one. Since these were minimization problems, the objective was to minimize the positive deviation (d_i^+) from each goal (note that this value would be zero if the goal was achieved). If a goal cannot be achieved, it is better to get it as close as possible to the target value.

The target values of various goals in a problem could have a wide range of possible values. For example, the highest priority goal of problem could be to maximize server utilization and another lower priority goal could be to maximize total profits. The corresponding target values could be 0.85 and \$100,000. Since these target values differ by such a large margin, it is possible they could skew the results and in essence make it more desirable to satisfy a lower priority goal. Thus, in order to make each goal more comparable to the other goals, the deviation from the target value was normalized by simply dividing it by the absolute value of the corresponding target value.

The objective function, Z, for each test problem was calculated by taking the sum of the normalization of the positive deviational variable (d_i^+) from each goal and multiplying it by the corresponding priority factor. Thus, Z was calculated using the following equation:

$$Z = \sum_{i=1}^{g} P_i \left(\frac{d_i^+}{|T_i|} \right) \tag{3.2}$$

where Z is the objective function to be minimized; g are the number of goals, T_i are the target values for the goals, P_i are the priority factors or penalties set by the decision-maker, d_i^+ are the non-negative positive deviational variables.

To ensure that the higher priority goals were satisfied first, it was important that the priority penalty be weighted heavily in favor of the higher priority goals (i.e., $P_1 \gg P_2 \gg \ldots \gg P_{g-1} \gg P_g$). To quantify this, the following formula was used for this study:

$$P_i = 100^{(g-i)} \qquad i = 1, \ldots, g, \qquad (3.3)$$

where g is the number of goals and i is the current goal. The value of one hundred was chosen based on experimentation. Thus, the objective function was penalized much more for not satisfying higher priority goals.

The basic structure of the goal programming model used in this research can then be expressed as:

Minimize $$Z = \sum_{i=1}^{g} P_i \left(\frac{d_i^+}{|T_i|} \right)$$

Subject to: $(\min G_i(\mathbf{X})): E(\mathbf{Y}(\mathbf{X},\omega)) + d_i^- - d_i^+ = T_i$

$X_j^l \le X_j \le X_j^u \qquad j = 1, ..., n$

$d_i^-, d_i^+ \ge 0, \qquad i = 1, \ldots, g$

$X_j \in \Re \qquad j = 1, \ldots, n$

$P_i = 100^{(g-i)} \qquad i = 1, \ldots, g$

where:

Z is the objective function to be minimized;

g is the number of goals;

n is the number of variables;

T_i are the targets for the corresponding goal;

P_i are the priority penalties;

G_i are the goals;

j is the number of parameters that are fed into the simulation model;

X_j are the input parameters, forming the vector **X**, that are fed into the simulation model;

ω represents the stochastic effects;

E(**Y**(**X**, ω)) is the expected value of the vector of output parameters, **Y**, determined from the simulation runs (see Figure 3.2, where for j input parameters to the simulation model, k output parameters are generated);

d_i^- and d_i^+ are non-negative deviational variables,

X_j^l and X_j^u are the lower and upper bounds, respectively, of the X_j input parameter.

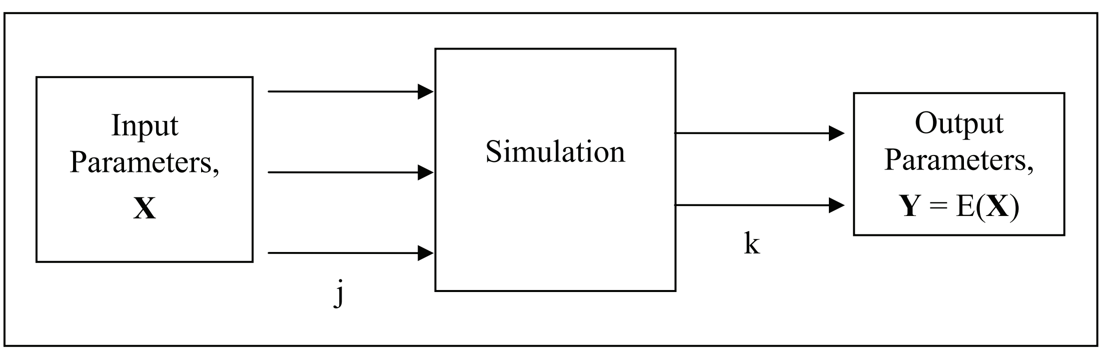


Figure 3.2: Simulation Model Input and Output Description.

3.1.2 Description of the Optimization Program and How It Interacts within the Simulation Optimization Methods

The optimization program starts the simulation optimization method by first defining the specific multi-criteria simulation optimization problem in terms of the appropriate goal programming formulation. It then determines the set of starting system parameters that will be fed into the simulation model. It also analyzes the output from the simulation model based on the goal programming formulation to calculate the current objective function value, the current goal values, and the next set of system values to evaluate. After each new set of values is determined, a stopping rule is checked. If the stopping rule has been satisfied, the optimization program records the final results and ends the simulation optimization method; otherwise, the next set of system values are fed into the simulation model.

The optimization program recorded every vector of input parameters that had been simulated and the values for the corresponding goals and objective function. In this way, the simulation optimization methods incorporated memory to ensure that points were not re-simulated. This was performed to reduce the number of calls to the simulation and thus the CPU time and computational effort needed to find a near-optimal solution. The optimization program also recorded all the simulation output data that were used to determine the quality of the solution.

3.1.2.1 Specific optimization techniques

Three heuristic-based optimization methods were selected to direct the optimization program. The multi-criteria simulation optimization methods examined were the genetic algorithm (GA-SO), tabu search (TS-SO), and the lexicographic Nelder-

Mead (LNM-SO) method. The specifics for each of the methods are described in the following sections.

The multi-criteria component of the method is addressed through the goal programming formulation. Therefore, once the goal programming formulation for the problem has been established, the optimization techniques simply work to solve the goal programming problem. The GA-SO and TS-SO methods only examine the objective function value. Thus, they essentially operate the same for multi-criteria problems as they would for single criteria problems. The LNM-SO method examines the individual goal values instead of the overall objective function value, and thus it operates somewhat differently. However, it can handle the case where a problem only has one goal or objective.

3.1.2.1.1 Genetic Algorithm Simulation Optimization (GA-SO) Method

The software for the GA-SO method was programmed in C++ and used the GAlib genetic algorithm package written by Matthew Wall at the Massachusetts Institute of Technology (Copyright (c) 1995-1996 Massachusetts Institute of Technology (MIT) and Copyright (c) 1996-2005 Matthew Wall (the Author). All rights reserved).

The starting solution was generated randomly, ensuring that it was feasible. The population size for each generation used for this study was 30 and a new population was created for each generation. The best solution from each population was passed to the next generation. The probability of crossover was 0.6 and the probability of mutation was 0.01. A roulette wheel approach was used to select the combination of input parameters or genomes for mating. Essentially, the better the fitness value for a particular genome, the higher the probability it would be selected for mating. However, every genome, regardless of fitness value, had a chance of being selected for mating. The fitness values were determined by comparing the objective function values for each genome in the population to the average objective function value of all genomes.

For GA-SO, a generation was defined as the creation of thirty solutions or genomes. This value was fixed and thus did not depend on the number of variables. Of course, the average number of calls to the simulation per generation would be less than thirty, since many of the solutions between generations would be the same. Additionally, the solutions that were passed directly from one generation to the next (i.e., the best solution from that generation) were not re-evaluated. Thus, the average number of solutions evaluated for each generation would also be less than thirty. For the GA-SO method, an iteration was defined as one generation.

The GA-SO multi-criteria method was terminated when either of the following conditions was satisfied:

1) the best score from the current generation divided by the average of the best scores from the previous 100 generations was at least 0.99, or
2) a total of 2000 solutions had been examined.

3.1.2.1.2 Tabu Search Simulation Optimization (TS-SO) Method

The Tabu Search (TS) simulation optimization (TS-SO) method was written using MATLAB. The basic tabu search code used to drive the TS-SO method's main program was based on "tabuks.m", a MATLAB file used to solve a knapsack problem using tabu search (Aurdal, 2006). This code was altered extensively to allow the code to work with real numbers and to be able to solve the TS-SO method.

The starting solution for the TS-SO method was generated randomly, ensuring that it was feasible. Neighbor solutions were determined by incrementing and decrementing each element of the set of input values, **X**, by α_1 and α_2. The values of α_1 and α_2 varied depending on the type of input parameter and the number of iterations already performed. The values of α_1 and α_2 start out larger. After 50 iterations and then again after an additional 25 iterations (or after 75 iterations total), the process starts over at the best point observed thus far with reduced values of α_1 and α_2. In this way, the search intensifies as the number of iterations increases. See Figure 3.3 for an example of the neighborhood selection procedure.

Neighborhood Definition:

$$X_j \pm \alpha_1 \text{ and } X_j \pm \alpha_2$$

For integer values:

$\alpha_1 = 2$ and $\alpha_2 = 1$

For real numbers:

for the first 50 iterations:	$\alpha_1 = 2$	and	$\alpha_2 = 1$
for the iterations 51-75,	$\alpha_1 = 1$	and	$\alpha_2 = 0.5$
for the remaining iterations,	$\alpha_1 = 0.25$	and	$\alpha_2 = 0.1$

Figure 3.3: Neighborhood Definition.

Three tabu lists, or restricted lists, were employed in this study. The first two lists, referred to as tabu increment and tabu decrement, attempted to force the search to examine several different areas and directions by restricting successive directional moves for each input parameter. The duration of these tabu lists were two iterations. For example, if X_1 was increased, it could not be increased again for two iterations. These lists, however, could be overridden if the tabu move resulted in the best solution found thus far.

A third tabu list restricted a solution from being selected as the best solution if it had been previously selected during the last ten iterations. This tabu list was referred to as tabu history. This restriction could not be overridden. The values chosen for each of the three tabu lists were determined by selecting the values that resulted in the best performance based on a few sample runs.

For TS-SO, an iteration is defined as the examination of the current solution and all of the solutions in its neighborhood. Thus, the maximum number of solutions examined for each iteration was variable. The total number of solutions to examine for each iteration for n variables was 4n+1. This was true since each variable had four neighbors. Of course, due to the definition of the neighborhood, many of the solutions examined from one iteration to another would be the same. Thus, the average number of calls to the simulation per iteration would be much less than 4n+1.

The TS-SO multi-criteria method was terminated when either of the following conditions was satisfied:

1) the objective function value for the point selected from the current iteration divided by the average of the points selected from the previous 100 iterations was at least 0.99, or
2) a total of 2000 solutions had been examined.

3.1.2.1.3 Lexicographic Nelder-Mead Simulation Optimization (LNM-SO) Method

The Lexicographic Nelder-Mead simulation optimization (LNM-SO) method used the Nelder-Mead Search method with the shrink coefficient equal to 0.9 as proposed by Barton and Ivey Jr. (1996). It also used a modified lexicographic ordering scheme to attempt to reach the preset target values. The modified lexicographic method works the same as the lexicographic method until a goal has been achieved. Once any of the points yields a value that is equal or better than a target value, the yielded value is viewed as being equal to the target value. This modification was done to allow lower priority goals the opportunity to improve, since improving a lower priority goal might worsen a higher priority goal's value. As long as the point still satisfied the target of the particular goal, the point is considered acceptable.

This method required that the function value of each goal be calculated at each step. There is a higher priority to satisfy goal one than the remaining goals, as in all pre-emptive goal programming. If the state of goal one becomes worse (that is the value of the goal is not less than the previous value), but the state of goal two improves, the new point was accepted. If a move improves the state of goal one but worsens the state of goal two, it is still approved (because of lexicographic ordering). Finally, if a move results in goal one being violated but improves the state of goal two, the move is rejected. The method continues until all goals are satisfied or until a goal cannot be satisfied or the stopping criteria has been met.

The first vertex of the initial simplex was chosen randomly, ensuring that it was feasible. The remaining vertices were determined by taking 20% of the difference of the upper and lower bounds for each variable. These values are then subtracted from the upper bound for each variable and added to the lower bound of their respective components to form high and low values. These formulas for Xhigh and Xlow can be formerly stated as follows:

$$Xhigh_j = X_j^u - 0.2 * (X_j^u - X_j^l) \qquad \forall j \tag{3.4}$$

$$Xlow_j = X_j^l + 0.2 * (X_j^u - X_j^l) \qquad \forall j \tag{3.5}$$

These Xhigh and Xlow values were then used to form the rest of the points of the initial simplex. The second vertex was set to the high values for all the variables. Similarly, the third vertex was set to the low values for all the variables. The remaining vertices were set by combining high and low values. This was done based on the pattern shown in the matrix found in Figure 3.4. The initial simplex was setup in this fashion to assure that it contained the majority of the solution space; since after experimentation, this approach appeared to provide the best results and reduce the likelihood of the problem from getting trapped at a local minimum.

$$\begin{bmatrix} \text{rand} & \text{rand} & \text{rand} & \text{rand} & \text{rand} & \text{rand} & \text{rand} & \text{rand} & \cdots \\ \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \cdots \\ \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \cdots \\ \text{Xhigh} & \text{Xhigh} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \cdots \\ \text{Xlow} & \text{Xlow} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \cdots \\ \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xlow} & \text{Xlow} & \text{Xhigh} & \text{Xhigh} & \cdots \\ \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xhigh} & \text{Xhigh} & \text{Xlow} & \text{Xlow} & \cdots \\ \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xhigh} & \text{Xlow} & \text{Xlow} & \cdots \\ \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xlow} & \text{Xhigh} & \text{Xhigh} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Figure 3.4: Formula for the Determination of the Initial Simplex.

An example calculation of the initial simplex for a problem where there are 6 design variables (i.e., n = 6) is performed in Figure 3.5. The initial simplex consists of n+1 or 7 vertices. The matrix Xrange provided the upper and lower bounds of each variable, respectively. The difference between the upper and lower bounds was calculated as Xdiff and then 20% of this value was calculated. The values of Xhigh were then calculated by subtracting 20% of Xdiff from the upper bounds. Similarly, the values of Xlow were calculated by adding 20% of Xdiff to the lower bounds. The Xhigh and Xlow values could also have been found directly using equations 3.4 and 3.5.

The first vertex for this example was chosen randomly, ensuring that it was feasible. The rest were chosen based on Xhigh and Xlow values as shown in Figure 3.5, following the pattern that was outlined in Figure 3.4.

For LNM-SO, an iteration is defined as the examination of the following: all the solutions in the simplex; the centroid solution; the reflection solution; and either the expansion, contraction, or shrink solution. As explained earlier, there are n+1 solutions or vertices in the simplex. Thus, the maximum number of unique solutions examined for each iteration was based on the number of variables. The maximum number of solutions to examine for each iteration for n variables was (n+1)+3 or n+4. Thus, for 4 variables, the maximum number of examinations per iteration would be 8. Of course, depending on the change the simplex undergoes after an iteration, many of the solutions examined from one iteration to another would be the same. Thus, the average number of calls to the simulation per iteration will be less than n+4.

n = 6 number of vertices = n +1 = 7

$$\text{Xrange} = \begin{array}{c} \begin{array}{cc} X^l & X^u \end{array} \\ \begin{bmatrix} 5 & 10 \\ 10 & 20 \\ 5 & 15 \\ 3 & 7 \\ 1 & 4 \\ 2 & 6 \end{bmatrix} \end{array} \qquad \text{Xdiff} = \begin{bmatrix} 5 \\ 10 \\ 10 \\ 4 \\ 3 \\ 4 \end{bmatrix} \qquad \therefore\ 20\% \text{ of Xdiff} = \begin{bmatrix} 1.0 \\ 2.0 \\ 2.0 \\ 0.8 \\ 0.6 \\ 0.8 \end{bmatrix}$$

$$\text{Xhigh} = \begin{bmatrix} 9 & 18 & 13 & 6.2 & 3.4 & 5.2 \end{bmatrix}$$

$$\text{Xlow} = \begin{bmatrix} 6 & 12 & 7 & 3.8 & 1.6 & 2.8 \end{bmatrix}$$

$$\text{Initial Simplex} = \begin{bmatrix} 6.6 & 10.93 & 11.27 & 6.12 & 2.58 & 4.2 \\ 9 & 18 & 13 & 6.2 & 3.4 & 5.2 \\ 6 & 12 & 7 & 3.8 & 1.6 & 2.8 \\ 9 & 18 & 7 & 3.8 & 1.6 & 2.8 \\ 6 & 12 & 13 & 6.2 & 3.4 & 5.2 \\ 6 & 12 & 13 & 6.2 & 1.6 & 2.8 \\ 9 & 18 & 7 & 3.8 & 3.4 & 5.2 \end{bmatrix}$$

Figure 3.5: Example Calculation of the Initial Simplex for LNM-SO.

The LNM-SO multi-criteria method was terminated when either of the following conditions was satisfied:

1) the objective function value for the point selected from the current iteration divided by the average of the points selected from the previous 100 iterations was at least 0.99, or
2) a total of 250 iterations had been performed.

3.1.3 General Description of the Simulation Model and How It Interacts within the Simulation Optimization Methods

The simulation model simulates the system based on the system parameters defined in the optimization program. The simulation model first reads in the input parameters as identified through the optimization program and any other relevant system parameters. The simulation was then run for a set length of time. It was desired to allow the simulation to run long enough to reach steady-state and to allow for enough data to be collected to provide reliable results. Additionally, a warm-up period was conducted to reduce the effects of the initial transient means. The value chosen for the simulation run length was 108 weeks. A warm-up period of 4 weeks was used. Thus, the data was collected for a total of 104 weeks, or two years.

In order to account for the stochastic nature of the simulation problem and to ensure a high level of confidence in the simulation results, it was necessary to conduct multiple runs for each set of input parameters. Thus, to ensure a 95 percent confidence interval that the sample mean was within $\pm$ 0.1 times the variance, four hundred replications were performed (Table 12-1, Pritsker and O'Reilly, 1999). At the conclusion of the final replication, the relevant output values from the simulation runs were recorded in a file and fed back into the optimization program. The simulation run was then terminated.

3.2 General Description of the Multi-Criteria Simulation Optimization Methods

The general multi-criteria simulation optimization method developed for this study can be divided into the following five sections: initial setup, optimization, simulation, evaluation, and termination. This process is outlined below and is visualized through a basic flowchart shown in Figure 3.6.

Initial Setup

Step 1: Setup the specifics for the goal programming problem.

Step 2: Set the initial conditions, including the number of goals, the target values, and the range of each input parameter.

Step 3: Randomly generate an initial feasible solution.

Optimization

Step 4: Generate additional feasible solutions to examine for the current iteration.

Step 5: Determine if the solution has previously been simulated.

Step 5a: If it has, go to Step 11.

Step 5b: If it has not, write the test solution to a file.

Simulation

Step 6: Read the input parameters from the file.

Step 7: Run the simulation for 400 replications.

Step 8: Record the appropriate output values and write them to a file.

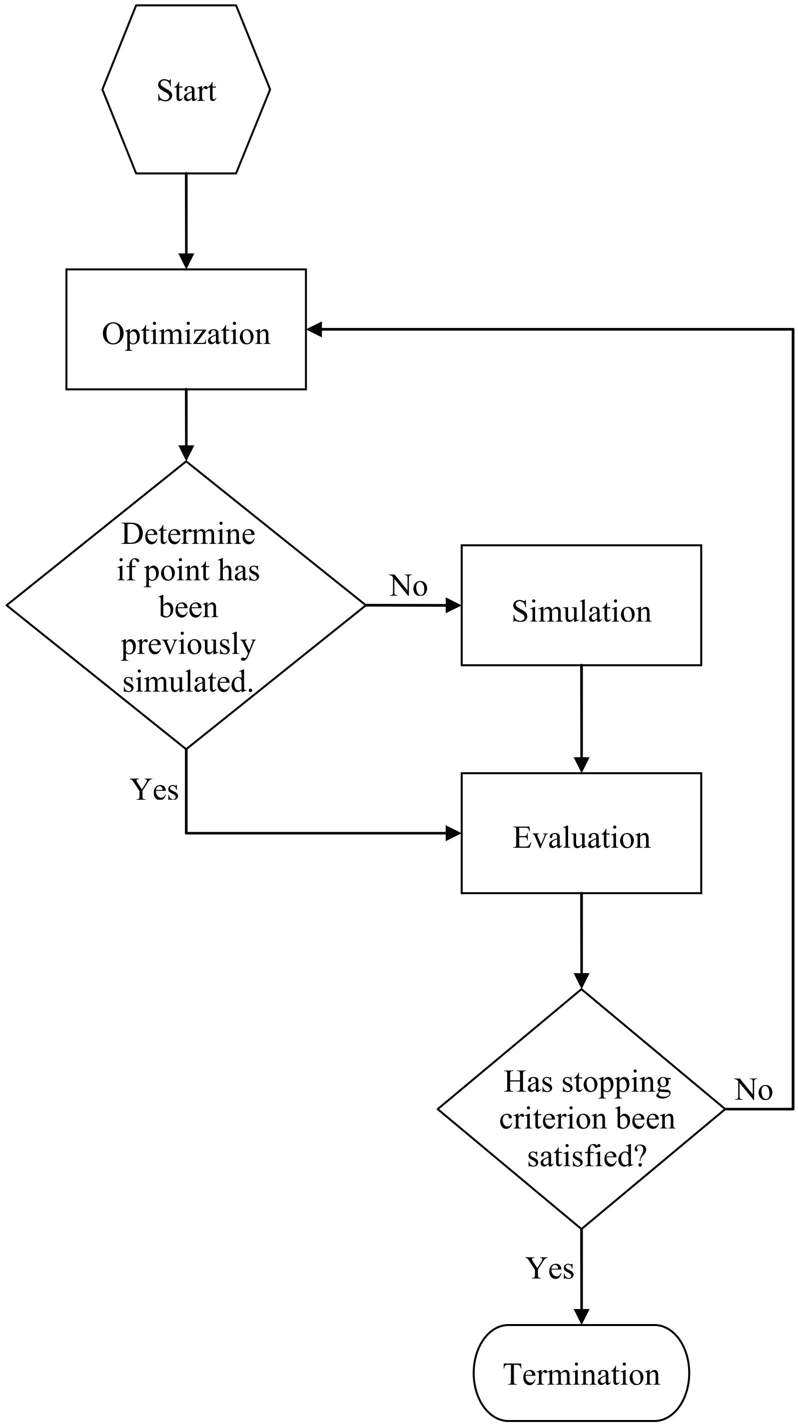


Figure 3.6: Basic Flowchart of the Simulation Optimization Methods.

Evaluation

Step 9: Read the simulation output values from file.

Step 10: Calculate the goal values and the objective function value and write the results to a file.

Step 11: Determine if this is the best objective function found thus far, if so, record the value and corresponding simulation input parameters (**X**).

Step 12: Have all the solutions in the population/neighborhood/simplex been examined? If so, go to Step 13; otherwise, select the next solution and go to Step 5.

Termination

Step 13: Check to see if the termination criterion has been satisfied. If so, go to step 14. Otherwise, record the best solution obtained during the current iteration to be used as the starting solution for the next iteration and go to Step 4.

Step 14: Record the final results, including the CPU time, the number of calls required to complete the simulation optimization method, the best solution found and its corresponding objective function value, and the number of total solutions evaluated (including duplicate solutions).

3.2.1 Details on Implementation of the Optimization Procedures

There are a few changes that are required to the general simulation optimization method to allow it to work with the specific optimization techniques. Additionally, the specific information required to implement the methods are also added. These changes and additions are described in the following sections. The actual computer programming codes used to implement each of these methods are provided in Appendix A.

3.2.1.1 GA-SO

There are only a few modifications needed to be made to the general multi-criteria simulation optimization method to allow it to work with the GA optimization technique. The primary modifications were in the initial parameters required to define the GA-SO and setup each generation.

The step-by-step procedure for the GA-SO method can be formulated and is outlined below. The actual C++ code used to implement the GA-SO method is provided in Appendix A.

Step-by-Step Procedure for the GA-SO Method.

Initial Setup

Step 1: Setup the specifics for the goal programming problem.

Step 2: Set the initial conditions, including the number of goals, the target values, and the range of each input parameter.

Step 2a: Set the initial population size, the mutation rate, and the crossover rate.

Step 3: Randomly generate an initial feasible solution.

Optimization

Step 4: Generate the population of feasible solutions to examine for the current generation.

Step 5: Determine if the solution has previously been simulated.

Step 5a: If it has, go to Step 11.

Step 5b: If it has not, write the test solution to a file.

Simulation

Step 6: Read the input parameters from the file.

Step 7: Run the simulation for 400 replications.

Step 8: Record the appropriate output values and write them to a file.

Evaluation

Step 9: Read the simulation output values from file.

Step 10: Calculate the goal values and the objective function value and write the results to a file.

Step 11: Determine if this is the best objective function found thus far, if so, record the value and corresponding simulation input parameters (**X**).

Step 12: Have all the solutions in the population been examined? If so, go to Step 13; otherwise, select the next solution in the population and go to Step 5.

Termination

Step 13: Check to see if the termination criterion has been satisfied. If so, go to step 14. Otherwise, record the best solution obtained during the current generation to be used as the starting solution for the next generation and go to Step 4 (Note: the best solution found from the current generation will always be the best solution found thus far, since the best solution from the current generation is always passed to the next generation).

Step 14: Record the final results, including the CPU time, the number of calls required to complete the simulation optimization method, the best solution found and its corresponding objective function value, and the number of total solutions evaluated (including duplicate solutions).

3.2.1.2 TS-SO

There were a few modifications and additions needed to be made to the general multi-criteria simulation optimization method to allow it to work with the TS optimization technique. The initial parameters required to set-up the method are stated as well as the definition of the neighborhoods. Additionally, the refinement in the values used to calculate the set of neighbor solutions based on the number of iterations performed and subsequently the re-initialization of the method are incorporated.

Thus, the step-by-step procedure for the TS-SO method can be formulated and is outlined below. The actual MATLAB codes used to implement the TS-SO method are provided in Appendix A.

Step-by-Step Procedure for the TS-SO Method.

Initial Setup

Step 1: Setup the specifics for the goal programming problem.

Step 2: Set the initial conditions, including the number of goals, the target values, the range of each input parameter, and the initial α_1 and α_2 values.

Step 2a: Set the tabu list sizes and set the initial lists as empty (thus, there are no initial tabu solutions).

Step 3: Randomly generate an initial feasible solution.

Optimization

Step 4: Generate the neighborhood of feasible solutions to examine for the current iteration (see section 3.1.2.1.2 for details on how neighborhoods are setup).

Step 5: Determine if the solution has previously been simulated.

Step 5a: If it has, go to Step 11.

Step 5b: If it has not, write the test solution to a file.

Simulation

Step 6: Read the input parameters from the file.

Step 7: Run the simulation for 400 replications.

Step 8: Record the appropriate output values and write them to a file.

Evaluation

Step 9: Read the simulation output values from file.

Step 10: Calculate the goal values and the objective function value and write the results to a file.

Step 11: Determine if this is the best objective function found thus far, if so, record the value and corresponding simulation input parameters (**X**).

Step 12: Have all the solutions in the neighborhood been examined? If so, go to Step 13; otherwise, select the next solution in the neighborhood and go to Step 5.

Termination

Step 13: Check to see if the termination criterion has been satisfied. If so, go to step 14. Otherwise, record the best solution obtained during the current iteration.

Step 13a: Update the tabu lists.

Step 13b: If 50 iterations have just been completed, select the best point found thus far as the starting solution for the next iteration and update the values of α_1 and α_2, appropriately. Re-initialize the tabu lists. Go to Step 4.

Step 13c: If 75 iterations have just been completed, select the best point found thus far as the starting solution for the next iteration and update the values of α_1 and α_2, appropriately. Re-initialize the tabu lists. Go to Step 4.

Step 13d: Go to Step 4.

Step 14: Record the final results, including the CPU time, the number of calls required to complete the simulation optimization method, the best solution found and its corresponding objective function value, and the number of total solutions evaluated (including duplicate solutions).

3.2.1.3 LNM-SO

There are several additions and modifications to the general multi-criteria simulation optimization method required to allow it to work with the LNM optimization technique. These have to do with calculating the initial simplex and determining which of these solutions have the highest function value, the second highest function value, and the lowest function value. These solutions are referred to as Xhigh, Xsechi, and Xlow; and their respective function values (that is the vectors for each solution containing the values for each goal) are f_Xhigh, f_Xsechi, and f_Xlow. Next, the centroid (Xcent) of all the points, excluding the "worst point" (Xold) is calculated. Then based on the function value of Xcent (f_Xcent), it is determined whether expansion or contraction should be attempted. Finally, a decision is made as to whether or not to shrink the simplex.

Thus, the step-by-step procedure for the LNM-SO method can be formulated and is outlined below. The actual MATLAB codes used to implement the LNM-SO method are provided in Appendix A.

Step-by-Step Procedure for the LNM-SO Method.

Initial Setup

Step 1: Setup the specifics for the goal programming problem.

Step 2: Set the initial conditions, including the number of goals, the target values, and the range of each input parameter, as well as the alpha, beta, and delta values.

Step 3: Randomly generate an initial feasible solution.

Optimization

Step 4: Calculate the remaining n vertices of the initial simplex, so there are a total of n+1 vertices (these are calculated as described in section. 3.1.2.1.3).

Step 5: Determine if the next solution of the simplex has previously been simulated.

Step 5a: If it has, go to Step 11.

Step 5b: If not, write the corresponding test solution to a file.

Simulation

Step 6: Read the input parameters from the file.

Step 7: Run the simulation for 400 replications.

Step 8: Record the appropriate output values and write them to a file.

Evaluation

Step 9: Read the simulation output values from file.

Step 10: Calculate the goal values and the objective function value and write the results to a file.

Step 11: Determine if this is the best objective function found thus far, if so, record the value and corresponding simulation input parameters (**X**).

Step 12: Have all the solutions in the simplex been examined? If so, determine the point with the highest function value (Xhigh), the second highest function value (Xsechi), and the lowest function value (Xlow) and go to Step 12a; otherwise, select the next solution and go to Step 5.

Step 12a: Determine the centroid (Xcent) of all the points, excluding the "worst point" (Xold). Reflect the worst point: Xreflect = Xold + (1 + alpha)*(Xcent - Xold) (perform steps 5 – 11 to calculate the function values of Xcent and Xreflect).

Step 12b: Is f_Xlow ≤ f(Xreflect) ≤ f_Xsechi? Yes: Xreflect replaces Xhigh.

Is f(Xreflect) < f_Xlow? Yes: Attempt expansion (go to Step 12c).

Is f(Xreflect) > f_Xsechi? Yes: Then attempt contraction (go to Step 12d).

If f(Xreflect) ≤ f_Xhigh, then Xreflect replaces Xhigh before going to Step 12d.

Step 12c: Attempt Expansion: Xexpansion = (gamma*Xreflect) + (1-gamma)*Xcent (perform steps 5 – 11 to calculate the function values of Xexpansion).

Step 12d: Attempt Contraction: Xcontraction = (beta*Xhigh) + (-beta)*Xcent (perform steps 5 – 11 to calculate the function values of Xcontraction).

Step 12e: Was Expansion successful? i.e., is f(Xexpansion) < f(Xreflect)?

Step 12f: Was Contraction successful? i.e., if f(Xcontraction) <= f_Xhigh, then continue.

If f(Xcontraction) > f_Xhigh, then contraction failed, so Shrink (go to Step 12g).

Step 12g: Shrink the entire simplex, except Xlow. Each vertex is replaced as follows: X_i = (delta*X_i) + (1-delta)*Xlow. Go to Step 4.

Termination

Step 13: Check to see if the termination criterion has been satisfied. If so, go to step 14. Otherwise, record the best solution obtained during the current iteration to be used as the starting solution for the next iteration and go to Step 4.

Step 14: Record the final results, including the CPU time, the number of calls required to complete the simulation optimization method, the best solution found and its corresponding objective function value, and the number of total solutions evaluated (including duplicate solutions).

3.3 Performance Measures

There are several performance measures developed for this study. These performance measures are intended to allow for comparison of the different simulation optimization methods based on the speed of computation and the quality of solution. The performance measures for this computational study include: the representative operation count (ROC) of simulation calls needed to reach the best solution (referred to as ROCBS), the count of simulation calls needed to complete the simulation optimization method (referred to as ROCCM), the best solution found (BSF), and an evaluation of the overall performance referred to as the time-quality estimator (TQE).

3.3.1 Speed of Computation

It is important to evaluate the speed of computation in order to determine the effectiveness of the simulation optimization method. Since CPU time is highly dependent on particular computer characteristics, one of the desires of this research was to be able to replace CPU time as a decision variable. Thus, it is important to determine or quantify the speed of computation through other measures.

The majority of the time used to run a simulation optimization method is used in calling and running the simulation model. ROC is method dependent and not computer dependent, and thus should be a much better comparison measure than CPU time. Therefore, ideally, as the CPU time increases, the ROC variables will increase at a similar rate. If this holds, CPU time can be replaced by ROC.

There are two “times” that are of particular interest and importance. These are the amount of computational effort required to complete the method and to find the best solution. Respectively, these are the **r**epresentative **o**peration **c**ounts required to **c**omplete the simulation optimization **m**ethod (ROCCM) and the **r**epresentative **o**peration **c**ounts required to find the **b**est **s**olution (ROCBS). These terms can be expressed by the following expressions:

$$\text{ROCCM} = \text{\# of simulation calls required to complete the method} \tag{3.6}$$

$$\text{ROCBS} = \text{\# of simulation calls required to find the best solution} \tag{3.7}$$

3.3.2 Quality of Solution

It is important to evaluate the quality of the solution to be able to determine the effectiveness of the simulation optimization method In order to quantify the quality of solution, the value of the best solution found (BSF) will be used. The best solution found is the best overall observed solution, or the solution found (SF), that generates the best objective function value for each set of runs (r) based on the domain.

$$BSF = \min\{SF_i\} \qquad i = 1,\ldots,r \qquad r = 400 \tag{3.8}$$

This allows for a direct comparison of each of the simulation optimization methods; however, it does not allow for comparison from one test problem to another. When discussing the best solution found (BSF) it is also important to examine the standard deviation, since this can have a large impact on the actual expected BSF values. Additionally, the initial or starting solution should also be examined. If the initial solution is very far from the best solution, it might be more difficult for the method to be able to reach the best solution, while the opposite is also true.

3.3.3 Overall Performance

The overall effectiveness of the simulation optimization method is based on both the computational speed and the quality of solution. Thus, it is desirable to combine these into one overall performance measure called the Time-Quality Estimator (TQE). It is important to realize that the overall effectiveness depends largely on the desires of the decision-maker and his/her priorities. Decision-makers can be classified as Time-Sensitive, Quality-Sensitive, or Time-Quality Neutral. If getting a decent quality solution in the fastest time possible is the priority of the decision-maker, they can be classified as time-sensitive. If getting a high quality solution in a reasonable amount of time is the priority of the decision-maker, they can be classified as quality-sensitive. Finally, time-quality neutral decision-makers want the best of both worlds and can be classified as wanting to get the best quality solution in the fastest time possible.

A confidence interval is a way to bound a mean of a population. For a 95% confidence interval for the mean of the population (μ), the formula can be expressed as:

$$\bar{x} - 1.96 * \frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + 1.96 * \frac{\sigma}{\sqrt{n}} \tag{3.9}$$

where $\bar{x}$ is the sample mean, σ is the standard deviation, and n is the sample size. The left hand side of the equation is known as the lower confidence limit and the right-hand side is known as the upper confidence limit (UCL). Thus, the equation for UCL is:

$$UCL = \bar{x} + 1.96 * \frac{\sigma}{\sqrt{n}} \tag{3.10}$$

To determine the TQE, the UCL for ROCBS and for BSF are used. ROCCM is not used since it is in part a function of the stopping criterion of the heuristic algorithm (or method), as well as the intermediate steps of the method. This is the case, because different methods may have different stopping criteria. Thus, for this measure to truly be a global performance measure, it should not be considered here. The UCL is used to examine a worst case scenario. This is true, since it is most desirable to have ROCBS and

BSF be as small as possible, that is we want to minimize the computational time and the best solution found. A weight is added for the importance of computational speed (w_S) and the quality of solution (w_Q). The weights vary depending on what case the decision-maker most closely associates with. Thus, the equation for TQE can be formulated as:

$$TQE = \frac{w_S(UCL_{ROCBS}) + w_Q(UCL_{BSF})}{w_S + w_Q} \tag{3.11}$$

The weights can fluctuate greatly, depending on how far the decision-maker sways towards being time versus quality sensitive. For the time-quality neutral case, w_S equals w_Q. For the other two cases, an appropriate ratio must be determined. These ratios will be set based on the experimental data and will be provided in the next chapter.

Chapter 4

Computational Results and Analysis

This chapter describes the test bed problems that were developed and used in this study. It also explains the experimental design that was implemented to generate the results of the study and to analyze the multi-criteria simulation optimization methods and the five test problems based on the defined performance measures.

4.1 Simulation Optimization Test Bed Problems

Five different test problems were developed and examined. These test bed problems represent the inventory (INV), logistic (LOG), program evaluation and review technique (PERT), production (PROD), and reliability (RELI) domains and are five of the most common and most important domains in industry. The success of a business could be determined by whether an adequate model of one or more of these domains has been implemented and applied in its development. Obviously, the better the policy, the better the business is able to make money and distribute products to their customers. Each of these domains have been well studied and should make for easier replication and implementation by other researchers that desire to compare new multi-criteria simulation optimization methods to those used in this research. For these reasons, the inventory, logistic, PERT, production, and reliability domains were selected to use as test problems for this research.

All problems and goals were setup to be minimization problems. Thus, if the goal was to maximize total profit, it would need to be converted to a minimization problem. This would be accomplished by simply multiplying the goal and the corresponding target value by negative one. Additionally, each of the goals were assumed to be independent of the other goals.

The solution space size is calculated by taking the product of all possible solutions of each variable. The equation can be stated as:

$$\text{Solution Space Size} = \prod_{j=1}^{n} \left(X_j^u - X_j^l \right) q_j \tag{4.1}$$

where n is the number of variables, q is the number of values per integer (i.e., if it is an integer, this value is 1), and X_j^u and X_j^l are the upper and lower bounds of X_j, respectively. Except for one variable of the logistic domain (which was an integer variable), the other variables could be recorded to two decimal points, thus these would have a q value of 100. For example, if the range of X_1 was 2, and X_2 was 3, and q_1 and q_2 were 100; the Solution Space Size would equal (2*100)*(3*100) = 60000. It is important to note that the solution space size depends only on the number of variables and it does not take into account the number of goals. A summary of the characteristics of each test problem is provided in Table 4.1 (the test problems will be explained in detail in the following sections).

Table 4.1: Breakdown of the Test Problem Characteristics.

Domain	Variables	Goals	Solution Space Size
INV	3	3	1.89E+10
LOG	5	2	3.90E+11
PERT	3	2	3.43E+08
PROD	6	2	2.40E+16
RELI	4	3	4.68E+11

The following sections provide a detailed explanation of each of the test problems. The problem definition was used to model the simulation network that was called by the simulation optimization method. The problems were also formulated as goal programming problems. This formulation was used to setup the simulation optimization methods to be able to solve each test bed problem. The actual AweSim simulation components (the network models, the control statements, and all other necessary components) used to simulate each problem are presented in Appendix B.

4.1.1 Inventory Domain Problem Definition

The first test problem examined in this study was an inventory model. The problem is based on Example 6-1 found in Pritsker and O'Reilly (1999), with several extensions. The problem examines an inventory system that has one bin with x capacity and y demand with periodic reviews, lost sales, and backorders. Initially, there are 72 units in inventory.

The time between demands is based on the lognormal distribution with a mean of 0.2 time units with a standard deviation of 0.2. Each demand represents an order for the purchase of one item. If there is not sufficient inventory on hand to fill the order, 80% of the customers will go elsewhere for their merchandise, while the remaining 20% will backorder the item.

The time between reviews was a constant value that was determined by the optimization routine (these values could range from one to ten time units). When the inventory level drops to or below the re-order point (R), additional units are ordered. The amount of units ordered is the re-order quantity (Q). The values of R could range from zero to thirty units, while the values of Q could range from thirty to one hundred units.

These values were determined through the optimization routine. The time required to order, receive, and re-stock the inventory (lead time) was based on the lognormal distribution with a mean and a standard deviation of 3 time units. When the new inventory was acquired, the backordered items were delivered to the customers, and then the inventory was available to fill any new orders.

A schematic flow chart of the problem is provided in Figure 4.1. The model was simulated to determine the expected values for the total number of sales (TNS), total orders (TO), total reviews (TR), average inventory (AvgInv), lost sales (LS), back orders (BO), bin wait time (BWT), and the customer wait time (CWT). The price charged for each unit was $65. The cost of each unit was $40, the cost of placing an order to replenish inventory levels was $50, the cost per review was $30, the inventory holding cost was $0.004 per item in inventory, the cost per lost sale was $20, and the cost per backorder was $10.

The objective of the model was to find the values of the re-order quantity (Q), re-order point (R), and the time between reviews (TBR) that best satisfy the following goals: maximize total profit, minimize the customer wait time, and minimize the bin wait time. The definitions and formulas used to calculate these goals were:

Goal 1: Maximize Total Profit (TP).

TP = Incomes – Expenses = Total Sales*(Price per unit - Cost per unit) – ((Cost per order*Total orders) + (Cost per review*Total reviews) + (Holding costs * Average Inventory on hand during period T * T) + (Cost per lost sale*Total lost sales) + (Cost per backorder*Total backorders)).

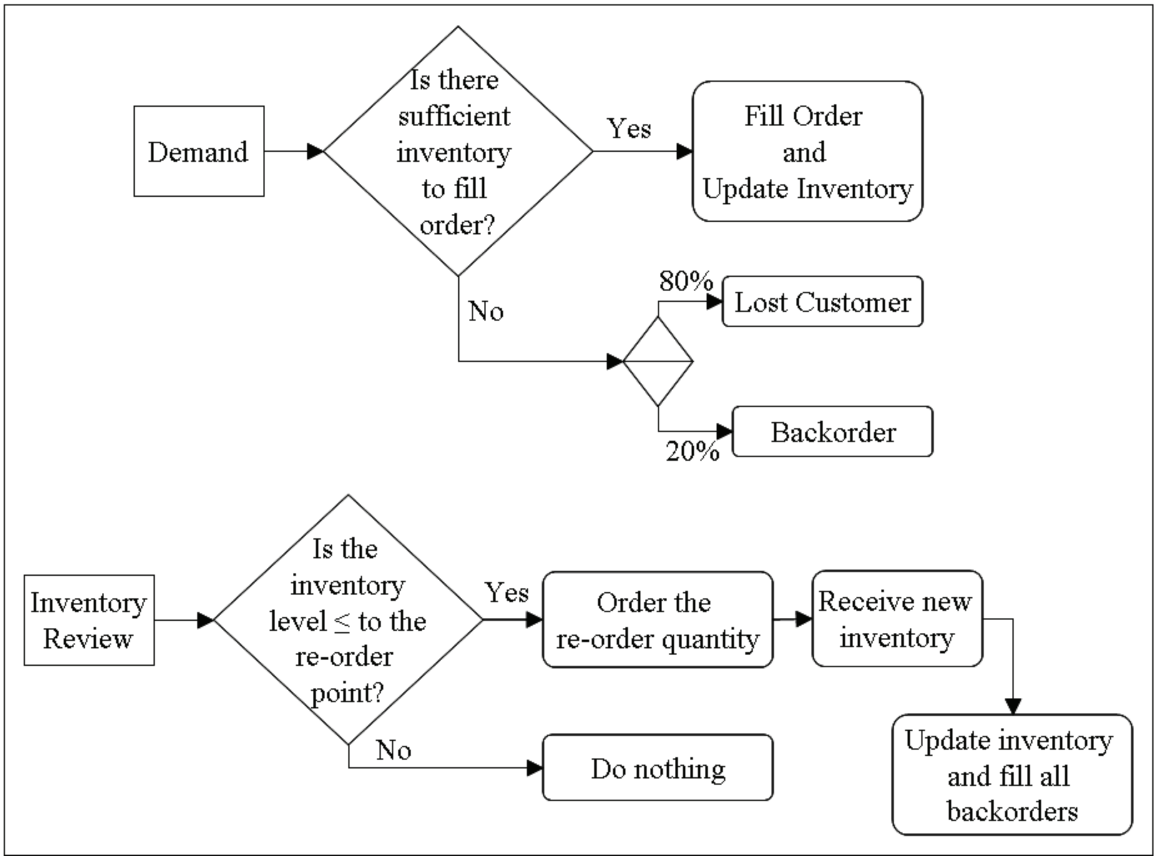


Figure 4.1: Schematic Flow Chart of the Inventory Problem.

Goal 2: Minimize customer wait time (CWT).

CWT = Time order is placed - time where inventory allows order to be filled.

Goal 3: Minimize the bin wait time (BWT).

BWT = Time bin is empty – time bin is filled.

The target values for each goal, respectively, were: $12,000, 0.05 time units, and 0.225 time units. It was most desirable to satisfy goal 1, then goal 2, and finally goal 3. The simulation model was setup to handle multiple bins and to have different standard deviation values for both the lead time and the time between demands; however, only one bin and one value for both standard deviations were examined in this study. Since the problem was setup to allow multiple bins, the simulation model was divided into a main network and a subnetwork.

The main network was used to read in and define the decision variables as identified through the optimization program and to call the subnetwork. It also defined the number of bins (one), the bin name, the initial inventory level, the mean lead time, and the mean time between demands.

The subnetwork consisted of a preliminary, a demand, a review, and an output component. The preliminary component defined the initial levels of the output variables that would be fed back into the optimization routine to determine each of the goal values. The preliminary component also called each of the other three components. The demand component was used to generate demands based on a lognormal distribution. It also was used to keep track of the number of demands, the total number of sales, the number of lost sales, and the number of backorders.

The review component was used to schedule additional reviews at a constant rate. It was also used to calculate the total number of reviews, the number of orders, the time necessary to receive orders, the bin wait time, and the customer wait time. The output component was used to determine the final values and write them to a file so that they could be fed into the optimization program in order to guide the simulation optimization method. The simulation network model, the subnetwork model, the control model, and the initial input file used to implement this simulation problem in AweSim are provided in Appendix B.

4.1.1.1 Inventory Goal Programming Formulation

The specific goal programming formulation used to implement the inventory test problem in each of the simulation optimization methods was:

$$\text{Minimize: } Z = \sum_{i=1}^{g} P_i \frac{d_i^+}{|T_i|} = 10000 \cdot \left(\frac{d_1^+}{12000} \right) + 100 \cdot \left(\frac{d_2^+}{0.05} \right) + \left(\frac{d_3^+}{0.225} \right)$$

Subject to:

$G_1 =$ -1 * ((E(TNS) * (65-40)) - (E(TO) * 50) - (E(TR) * 30) - (E(AvgInv) * 0.004 * Runtime) - (E(LS) * 20) - (E(BO) * 10)) $+ d_1^- - d_1^+) = -12000$

$G_2 = E(CWT(X)) + d_2^- - d_2^+ = 0.05$

$G_3 = E(BWT(X)) + d_3^- - d_3^+ = 0.225$

$$30 \le X_1 \le 100$$

$$0 \le X_2 \le 30$$

$$1 \le X_3 \le 10$$

$d_i^-, d_i^+ \ge 0, \qquad i=1,\ldots,g$

$X_j \in \Re$ (rounded to the nearest hundredth) $\qquad j = 1,\ldots,n$

$P_i = 100^{(g-i)}, \qquad i=1,\ldots,g$

where Z is the objective function to be minimized; g are the number of goals; n are the number of variables; T_i are the target values for the goals; P_i are the priority penalties; d_i^- and d_i^+ are non-negative deviational variables; X_1 is the re-order quantity (Q); X_2 is the re-order point (R); and X_3 is the time between reviews (TBREV).

4.1.2 Logistics Domain

The second test problem examined in this study was a logistics model. The specific problem is based on a logistics network provided in Pritsker et al (1994). The problem involves three different types of machines that breakdown periodically. The machines are self-operable and therefore are not constantly observed. Thus, the machine failures are not necessarily discovered immediately. The time to discover a machine failure (which is comprised of the time between machine failures, or breakdowns, and the time required to discover the machine failures) of each machine are based on the exponential distribution. The possible values of the mean time to discover failures could range: from two to five time units for machine 1, from five to ten time units for machine 2, and from seven to twelve time units from machine 3. The actual mean discover times for each machine type was determined through the optimization routine. For this problem, it was assumed that there was a limitless supply of each machine type.

Each time a machine fails, it requires a special part to fix it (the same part will fix all three machine types). Initially, there are 100 such parts located at the warehouse. If the warehouse has a repair part on hand, it will furnish it immediately. Repair parts are only ordered to re-stock the warehouse after a set number of failures have occurred. This value is referred to as "fail." The number of parts requested by the warehouse to re-stock the level of repair parts is equal to the number of failures accumulated before the stock levels are replenished (i.e., "fail" number of parts are ordered after "fail" number of failures). The possible values of "fail" could range from two to fifteen units. The actual value used was defined through the optimization routine. The time required to receive the parts necessary to re-stock the inventory was based on the exponential distribution, and the mean value was determined through the optimization routine. The possible values could range from one to five time units.

A schematic flow chart of the problem is provided in Figure 4.2. The model was simulated to determine the expected values for the average number of orders of parts for machine 1 (ORD1_AVG average), the average number of orders of parts for machine 2 (ORD2_AVG), the average number of orders of parts for machine 3 (ORD3_AVG), the average time to fill an order of demand type 1 (OUT1_AVG), the average time to fill an order of demand type 2 (OUT2_AVG), the average time to fill an order of demand type 3 (OUT3_AVG), and the average utilization of the resource part (UTIL_AVG).

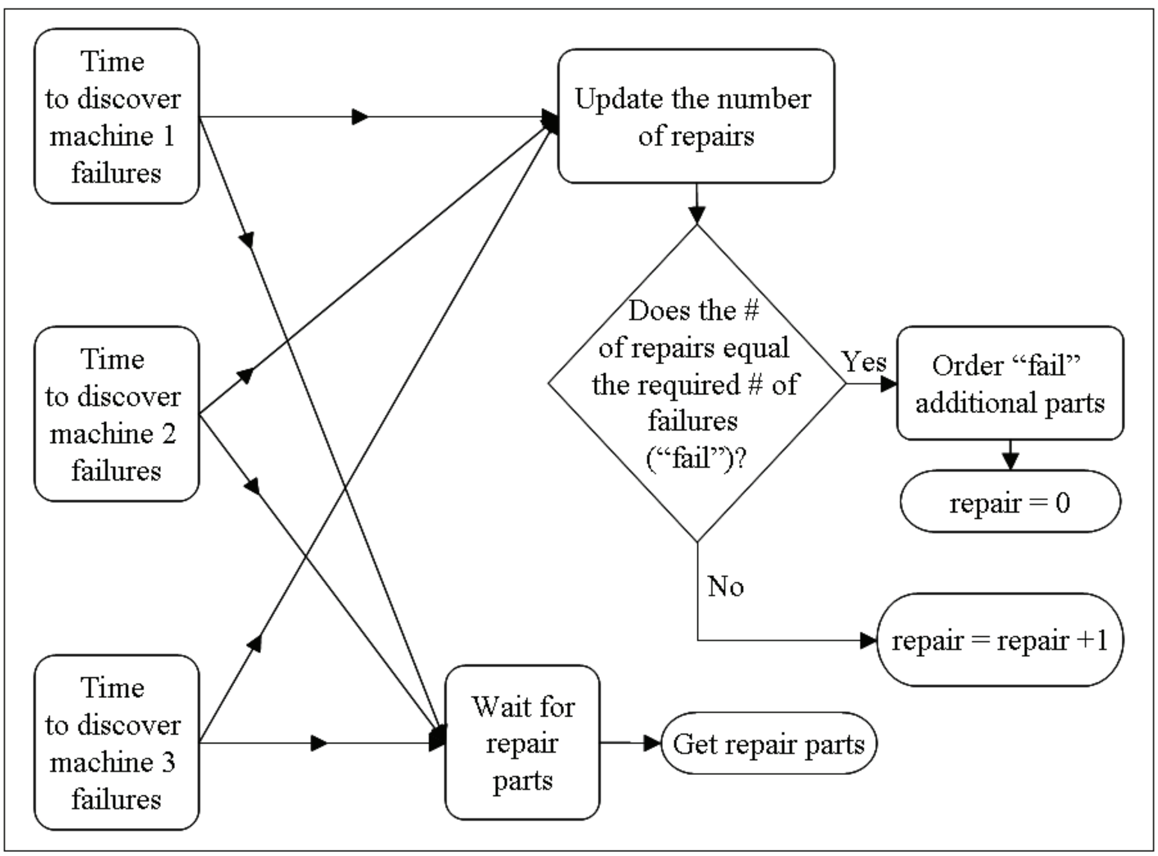


Figure 4.2: Schematic Flow Chart of the Logistics Problem.

Each machine requires the same part for repair, however, the urgency of the need for the part is dependent on the machine type. Thus, the company has to pay a different price to order parts because of a failure of a particular machine type. The cost required to order parts because of a machine type 1 failure was $25, for machine type 2 was $20, and for machine type 3 was $30. Additionally, there is a penalty cost required to fill the order of repair parts for machine type 1, 2, and 3. These costs are $3, $2, and $4, respectively.

Additional parts could only be ordered at set times. These times were determined by accumulating a set number of failures before ordering additional parts. The amount of parts to be re-ordered was the same as the number of failures accumulated before re-ordering could occur. Thus, the objective of the model was to find this accumulation value, the mean time parts take to arrive, and the mean time to discover the failure of machine types 1, 2, and 3 that best satisfy the following goals: minimize total costs and maximize average part utilization (i.e., the company does not want to have a warehouse full of repair parts that are not regularly needed). The definitions and formulas used to calculate these goals were:

Goal 1: Minimize Total Costs (TC).

TC = (Average # of orders for machine 1 * Cost per order of parts for machine 1) + (Average # of orders for machine 2 * Cost per order of parts for machine 2) + (Average # of orders for machine 3 * Cost per order of parts for machine 3) + (Average time required to receive a part for machine 1 * penalty cost for machine 1) + (Average time required to receive a part for machine 2 * penalty cost for machine 2) + (Average time required to receive a part for machine 3 * penalty cost for machine 3).

Goal 2: Maximize Average Part Utilization (UTIL_AVG).

UTIL_AVG = Average part utilization / the number of parts

The target values for each goal, respectively, were: \$409.80 and 0.85. It was most desirable to satisfy goal 1 and then goal 2. The simulation network was composed of a main section and a data section. The main section was used to simulate the logistics test problem. The data section was used to read the data from the optimization program and to record the final results of the simulation and write the appropriate values to a file so that they could be read into the optimization program in order to guide the simulation optimization method.. The simulation network model and the control statements used to implement this simulation problem in AweSim are provided in Appendix B.

4.1.2.1 Logistics Mixed-Integer Goal Programming Formulation

The specific goal programming formulation used to implement the logistics test problem in each of the simulation optimization methods was:

Minimize: $Z = \sum_{i=1}^{g} P_i \frac{d_i^+}{|T_i|} = 100 \cdot \left(\frac{d_1^+}{409.80}\right) + \left(\frac{d_2^+}{0.85}\right)$

Subject to:

$G_1 =$ E(ORD1_AVG) * 25 + E(ORD2_AVG) * 20 + E(ORD3_AVG)* 30 + E(OUT1_AVG) * 3 + E(OUT2_AVG) * 2 + E(OUT1_AVG) * 4 + $d_1^- - d_1^+ = 409.80$

$G_2 =$ -1 * (E(UTIL_AVG)/100) + $d_2^- - d_2^+ = -0.85$

$2 \le X_1 \le 15$ (integer variable)

$1 \le X_2 \le 5$

$$2 \le X_3 \le 5$$

$$5 \le X_4 \le 10$$

$$7 \le X_5 \le 12$$

$$d_i^-, d_i^+ \ge 0, \qquad i=1,\ldots,g$$

$$X_1 \in Z$$

$$X_j \in \Re \text{ (rounded to the nearest hundredth)} \qquad j = 2,\ldots,n$$

$$P_i = 100^{(g-i)}, \qquad i=1,\ldots,g$$

where Z is the objective function to be minimized; g are the number of goals; n are the number of variables; T_i are the target values for the goals; P_i are the priority penalties; d_i^- and d_i^+ are non-negative deviational variables; X_1 is an integer variable and it represents the number of parts to order after X_1 failures (FAIL); X_2 is the mean time parts take to arrive (ARRIVE); X_3 is the mean time to discover a failure of machine type 1 (DISC1); X_4 is the mean time to discover a failure of machine type 2 (DISC2); and X_5 is the mean time to discover a failure of machine type 3 (DISC3).

4.1.3 PERT Domain

The third test problem examined in this study was a PERT model comprised of nine activities and six nodes. The problem is based on Example 7-2 found in Pritsker and O'Reilly (1999). The problem has been extended to determine the criticality index of each activity. An activity's criticality index was calculated by determining the percentage of runs that the activity was on the critical path (CP) across all 400 runs. There was also a cost and a penalty factor (PF) associated with an activity being located on the critical path.

Thus, instead of defining the critical path (CP) based only on duration, it is also necessary to include the activity cost and PF. The cost and PF values for each activity are shown in Table 4.2.

The times needed to complete each activity were based on the triangular distribution. Table 4.2 shows the low, mean, and high values as well as the predecessor activities for each activity. The low values for activities 3, 4, and 6 (referred to as Low3, Low4, and Low6, respectively) were determined through the optimization routine. The possible low values for activity 3 ranged from five to twelve time units, while the possible low values for both activity 4 and activity 6 ranged from one to eight time units. Figure 4.3 shows a diagram of the PERT network. The model was simulated to determine the expected values for the criticality index of each activity, the time required to complete the project, and the average time of the last arrival to each node. The simulation for the PERT network was not performed for 104 weeks as it was for the other test problems. Instead, the network was simply started and allowed to run until the completion of the project.

Table 4.2: PERT Problem Information.

Activity #	Activity Durations Low	Mode	High	Predecessor Activities	CP Cost ($)	CP Penalty Factor (PF)
1	1	3	5	-	5	2
2	3	6	9	-	3	2
3	Low3	13	19	-	2	1
4	Low4	9	12	1	7	1
5	1	3	8	1	4	2
6	Low6	9	16	2, 5	3	1
7	4	7	13	2, 5	2	2
8	3	6	9	1, 4	5	1
9	1	3	8	3, 7	4	2

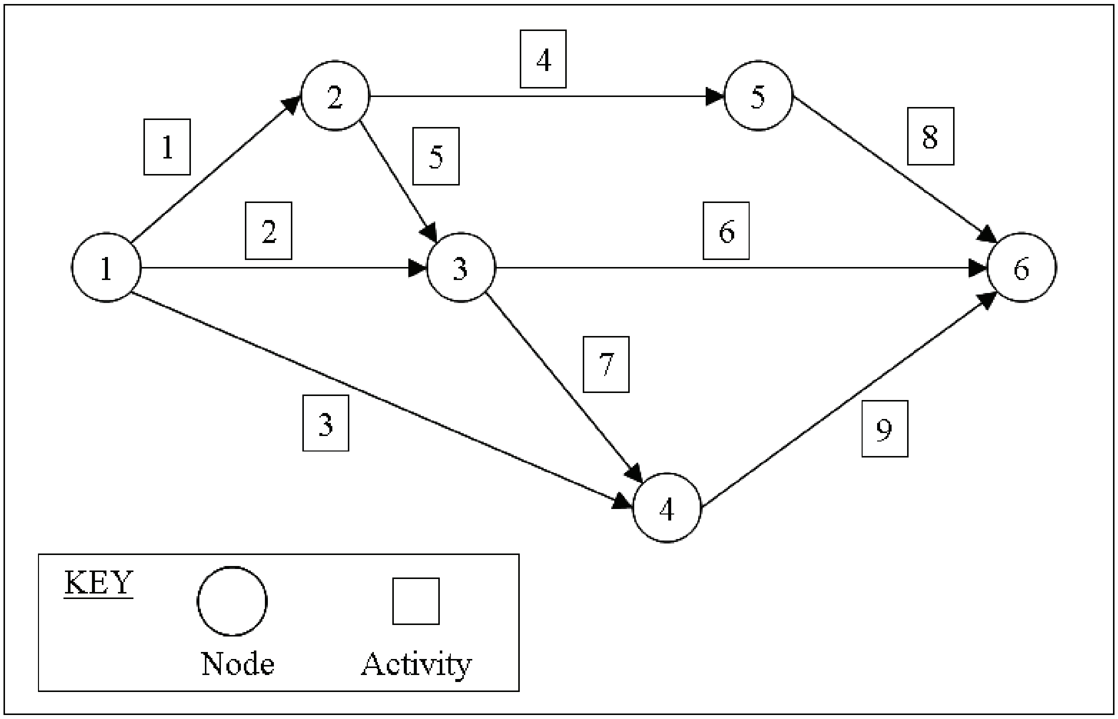


Figure 4.3: PERT Network Diagram.

The objective of the problem was to find the low times for the triangular distributions to complete activities 3, 4, and 6 that best satisfy the following goals: minimize the total cost and minimize the project completion time. The definitions and formulas used to calculate these goals were:

Goal 1: Minimize Total Cost (TC): $TC = \sum_{j=1}^{k} (E(CRIT_j) * C_j * PF_j)$

TC = (Criticality Index for activity 1 * Cost of activity 1 being on the critical path * the Penalty for choosing activity 1 on the critical path) + (Criticality Index for activity 2 * Cost of activity 2 being on the critical path * the Penalty for choosing activity 2 on the critical path) + (Criticality Index for activity 3 * Cost of activity 3 being on the critical path * the Penalty for choosing activity 3 on the critical path) +(Criticality Index for activity 4 * Cost of activity 4 being on the critical path * the Penalty for choosing activity 4 on the critical path) + (Criticality Index for activity 5 * Cost of activity 5 being on the critical path * the Penalty for choosing activity 5 on the critical path) + (Criticality Index for activity 6 * Cost of activity 6 being on the critical path * the Penalty for choosing activity 6 on the critical path) + (Criticality Index for activity 7 * Cost of activity 7 being on the critical path * the Penalty for choosing activity 7 on the critical path) + (Criticality Index for activity 8 * Cost of activity 8 being on the critical path * the Penalty for choosing activity 8 on the critical path) + (Criticality Index for activity 9 * Cost of activity 9 being on the critical path * the Penalty for choosing activity 9 on the critical path)

Goal 2: Minimize Project Completion Time (PT_AVG)

PT_AVG = the average time needed to complete the project.

The target values for each goal, respectively, were: \$21.76 and 18.04 time units. It was most desirable to satisfy goal 1 and then goal 2. The simulation network was composed of a main section and a data section. The main section was used to simulate the PERT test problem. The data section was used to read the data from the optimization program and to record the final results of the simulation and write the appropriate values to a file that so that they could be read into the optimization program in order to guide the simulation optimization method.. The simulation network model, the control statements, and the C program (used to calculate the criticality index) used to implement this simulation problem in AweSim are provided in Appendix B.

4.1.3.1 PERT Goal Programming Formulation

The specific goal programming formulation used to implement the PERT test problem in each of the simulation optimization methods was:

$$\text{Minimize} \quad Z = \sum_{i=1}^{g} P_i \frac{d_i^+}{|T_i|} = 100 \cdot \left(\frac{d_1^+}{21.76} \right) + \left(\frac{d_2^+}{18.04} \right)$$

Subject to:

$$\begin{aligned} G_1 = \; & (E(CRIT1) * 5 * 2) + (E(CRIT2) * 3 * 2) + (E(CRIT3) * 2 * 1) \\ & + (E(CRIT4) * 7 * 1) + (E(CRIT5) * 4 * 2) + (E(CRIT6) * 3 * \\ & 1) + (E(CRIT7) * 2 * 2) + (E(CRIT8) * 5 * 1) + (E(CRIT9) * 4 \\ & * 2) + d_1^- - d_1^+ = 21.76 \end{aligned}$$

$$G_2 = E(PT\_AVG) + d_2^- - d_2^+ = 18.04$$

$$5 \le X_1 \le 12$$

$$1 \le X_2 \le 8$$

$$1 \le X_3 \le 8$$

$$d_i^-, d_i^+ \ge 0, \qquad i=1,\ldots,g$$

$$X_j \in \Re \text{ (rounded to the nearest hundredth)} \qquad j = 1,\ldots,n$$

$$P_i = 100^{(g-i)}, \qquad i=1,\ldots,g$$

where Z is the objective function to be minimized; g are the number of goals; n are the number of variables; T_i are the targets for the goals; P_i are the priority penalties; d_i^- and d_i^+ are non-negative deviational variables; X_1 is the low time (of a triangular distribution) to complete activity 3 (LOW3); X_2 is the low time (of a triangular distribution) to complete activity 4 (LOW4); and X_3 is the low time (of a triangular distribution) to complete activity 6 (LOW6).

4.1.4 Production Domain

The fourth problem examined in this study was a production model. The specific problem is based on a production network provided in Pritsker et al (1994). Cargo arrives at an exponential rate with a mean of 5 time units. The cargo is assembled onto a pallet which is then placed onto a cart. There are six pallets and two carts available. The pallet is then transported via a cart to one of three parallel processing areas. The pallet will be delivered to the processing area that has the fewest number of pallets currently waiting at each location. It is then removed from the cart. Each processing area has a separate processing time. After processing, the pallets are loaded back onto carts and transported to an inspection station. The pallets are unloaded from the carts and then they are inspected. Finally, the pallets are transported to the unloading area (no cart is required for this transport) where they are unloaded and the pallet is freed up and available for new cargo.

The time to assemble or load the pallet is based on the exponential distribution with a mean value that is determined by the optimization routine. This value could range from five to ten time units. It takes virtually no time to load a pallet onto a cart or unload a pallet from a cart. The transportation times to each area (processing, inspection, and unloading) were based on the exponential distribution with mean times for each area determined by the optimization routine. These transportation values could range from three to seven time units, from one to four time units, and from two to six time units, respectively.

The time to process loads through processing areas 1, 2, and 3 were based on the exponential distribution and had mean processing times of 8, 10, and 12 time units respectively. The mean inspection time was exponentially distributed. Its value could range from 10 to 20 time units. The actual value used was determined from the optimization routine. The unloading or disassembly time of a pallet was based on the exponential distribution with a mean value that was determined by the optimization routine. The possible values could range from five to fifteen time units.

A schematic diagram of the problem is provided in Figure 4.4. The model was simulated to determine the expected values for throughput (TPUT), the average time in system (TIS_AVG), the average utilization of pallets (PU_AVG), and the average utilization of carts (CU_AVG).

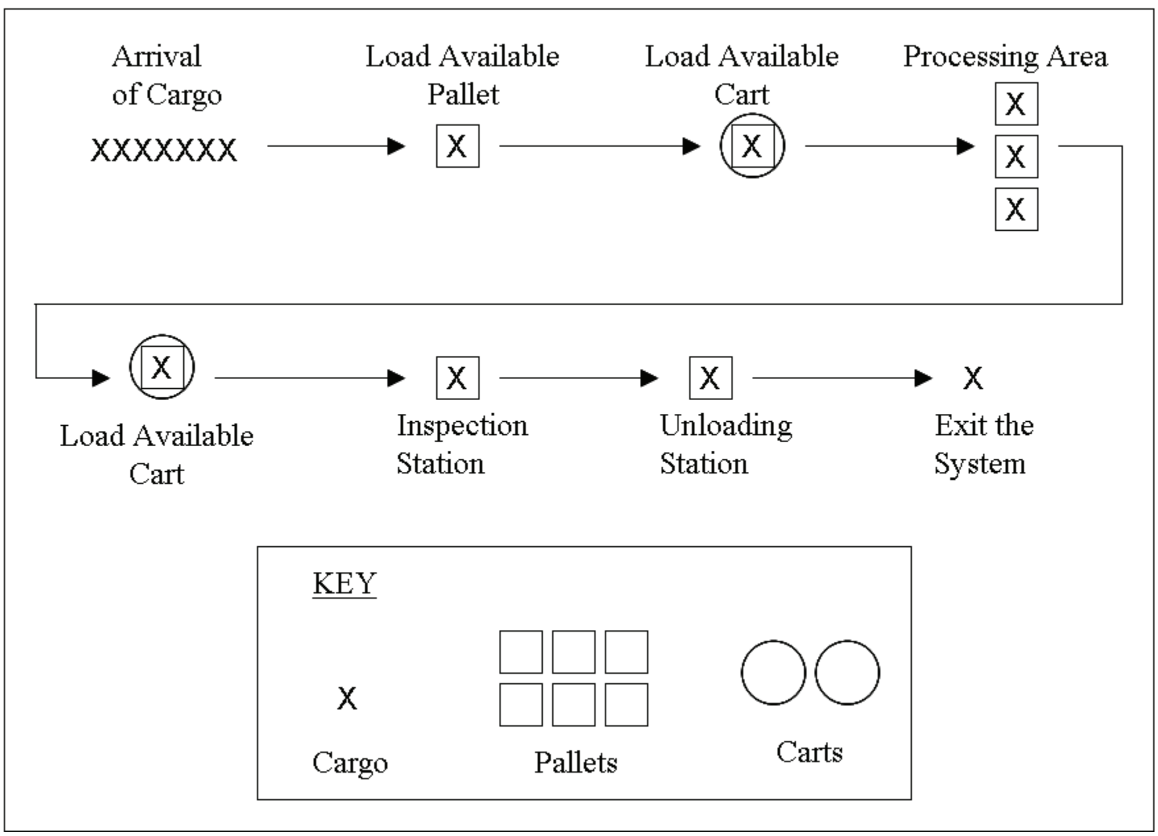


Figure 4.4: Schematic Diagram of the Production Problem.

The company charges $100 for each cargo load that passes through the system. The cost to process the cargo through the system varies depending on the quality of worker and transportation. Essentially, in order for the material to travel through the system faster, the company has to have more skilled labor and better quality equipment. Thus, it costs the company more to process cargo faster. This processing cost is $1000 per average time the cargo is in the system.

The objective of the model was to find the values for the mean time required to load the material onto a pallet (LOAD), the mean time required to inspect the material (INSPECT), the mean time required to unload the material from a pallet (UNLOAD), the mean time required to transport to the processing area (TRANSPORT1), the mean time required to transport to the inspection station (TRANSPORT2), and the mean time required to transport to the unload station (TRANSPORT3) that best satisfy the following goals: maximize total profit and maximize the pallet and cart utilization. The definitions and formulas used to calculate these goals were:

Goal 1: Maximize Total Profit (TP)

TP = Incomes – Expenses = (Price per unit * the number of units that pass through the system) – (Cost to process cargo / the average time in system)

Goal 2: Maximize Pallet and Cart Utilization

Total Utilization = Pallet Utilization + Cart Utilization

The target values for each goal, respectively, were: $4600.00 and 1.90. It was most desirable to satisfy goal 1 and then goal 2. The simulation network was composed of a main section and a data section. The main section was used to simulate the production test problem. The data section was used to read the data from the optimization program and to record the final results of the simulation and write the appropriate values to a file that so that they could be read into the optimization program in order to guide the simulation optimization method. The simulation network model, the control statements, and the initial data used to implement this simulation problem in AweSim are provided in Appendix B.

4.1.4.1 Production Goal Programming Formulation

The specific goal programming formulation used to implement the production test problem in each of the simulation optimization methods was:

Minimize: $$Z = \sum_{i=1}^{g} P_i \frac{d_i^+}{|T_i|} = 100 \cdot \left(\frac{d_1^+}{4600.00} \right) + \left(\frac{d_2^+}{1.09} \right)$$

Subject to:

$$G_1 = -1 * (100*(E(TPUT)) - (1000 * (1/E(TIS\_AVG)))) + d_1^- - d_1^+ = -4600.00$$

$$G_2 = -1 * (E(PU\_AVG)/6 + E(CU\_AVG)/2) + d_2^- - d_2^+ = -1.09$$

$$5 \le X_1 \le 10$$

$$10 \le X_2 \le 20$$

$$5 \le X_3 \le 15$$

$$3 \le X_4 \le 7$$

$$1 \le X_5 \le 4$$

$$2 \le X_6 \le 6$$

$$d_i^-, d_i^+ \ge 0, \qquad i=1,\dots,g$$

$$X_j \in \Re \text{ (rounded to the nearest hundredth)} \qquad j = 1,\dots,n$$

$$P_i = 100^{(g-i)}, \qquad i=1,\ldots,g$$

where Z is the objective function to be minimized; g are the number of goals; n are the number of variables; T_i are the target values for the goals; P_i are the priority penalties; d_i^- and d_i^+ are non-negative deviational variables; X_1 is the mean time required to load the cargo onto a pallet (LOAD); X_2 is the mean time of inspection (INSPECT); X_3 is the mean time to unload the cargo from a pallet (UNLOAD); X_4 is the mean transportation time to the processing area (TRANSPORT1); X_5 is the mean transportation time to the inspection station (TRANSPORT2); and X_6 is the mean transportation time to the unloading station (TRANSPORT3).

4.1.5 Reliability Domain

The final problem examined in this study was a reliability model. The specific problem is based on a reliability network provided in Pritsker et al (1994). There are two separate machines operating at a factory. Each machine has occasional breakdowns that require repairs to make them operational again.

The time between failures of machine 1 and machine 2 are both based on the exponential distribution. The mean failure time for each machine was determined through the optimization routine. The possible values for machine 1 could range from two to fifteen time units, while the possible values for machine 2 could range from one to five time units. The factory can continue to operate as long as both machines are not broken down at the same time. The repair time for machine 1 and for machine 2 are exponentially distributed with mean repair times that were determined through the optimization routine. The possible repair times could range from one to ten time units for machine 1 and from two to twelve time units for machine 2.

A schematic diagram of the problem is provided in Figure 4.5. The model was simulated to determine the expected values for the average number of machine 1 breakdowns (BD1_ AVG), the average number of machine 2 breakdowns (BD1_ AVG), the average number of system breakdowns (SYSBD_A), and the average system downtime (DT_AVG).

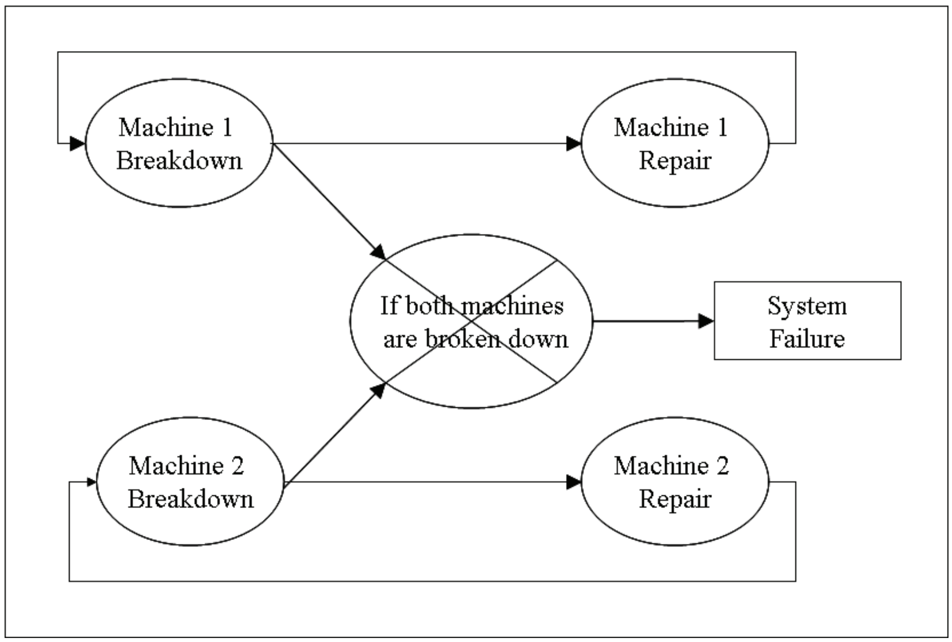


Figure 4.5: Schematic Flow Chart of the Reliability Problem.

Each machine breakdown costs the company. The breakdown cost for machine 1 is \$1000 and the breakdown cost for machine 2 is \$2000. If both machines go down at the same time, it costs the company an additional \$10,000. If the company purchases better machines, they will have fewer breakdowns. However, this will cost the company additional money. The additional cost varies depending on the average time between failures for each machine. Finally, the cost of repair must also be accounted for. The company can also purchase equipment that will allow each machine to be repaired faster.

The cost to repair machine 1 is $5000 per average repair time and the cost to repair machine 2 is $6000 per average repair time. Thus, it costs more to be able to repair machines faster.

The objective of the model was to find the values of the mean time to failure for machine 1 and 2, and the mean repair time for machine 1 and 2 that best satisfy the following goals: minimize total costs, minimize the total downtime, and minimize the total number of machine failures. The formulas used to determine the total cost and the total number of machine failures both use the average number of breakdowns for each machine. Thus, these goals are not necessarily independent. However, the goals can be thought of as being independent for certain situations. One such scenario is the case where there is a pool of machines to select from. It is possible to select better machines based on the probability of failure. Better quality machines should result in fewer failures, although this is not guaranteed.

The definitions and formulas used to calculate the goals were:

Goal 1: Minimize total costs (per thousand dollars)

Total Cost = (Average Number of Breakdowns of machine 1 * the Mean Time to Failure of machine 1 * Cost of breakdown of machine1) + (Average Number of Breakdowns of machine 2 * the Mean Time to Failure of machine 2 * Cost of breakdown of machine 2) + (the average number of system breakdowns * cost of breakdown * (cost of repair for machine 1 + cost of repair for machine 2))

Goal 2: Minimize Total Downtime

DT_AVG = the average system downtime

Goal 3: Minimize Total Failures (TF)

TF = the average number of machine 1 breakdowns + the average number of machine breakdowns + the average number of system breakdowns

The target values for each goal, respectively, were: $126.16 thousand, 0.10 time units, and 6.62 failures. It was most desirable to satisfy goal 1, then goal 2, and finally goal 3. The simulation network was composed of a main section and a data section. The main section was used to simulate the reliability test problem. The data section was used to read the data from the optimization program and to record the final results of the simulation and write the appropriate values to a file so that they could be read into the optimization program in order to guide the simulation optimization method. The simulation network model and the control statements used to implement this simulation problem in AweSim are provided in Appendix B.

4.1.5.1 Reliability Goal Programming Formulation

The specific goal programming formulation used to implement the reliability test problem in each of the simulation optimization methods was:

Minimize: $$Z = \sum_{i=1}^{g} P_i \frac{d_i^+}{|T_i|} = 10000 \cdot \left(\frac{d_1^+}{126.16}\right) + 100 \cdot \left(\frac{d_2^+}{0.10}\right) + \left(\frac{d_3^+}{6.62}\right)$$

Subject to:

$G_1 =$ (E(BD1_AVG) * (Fail1 * 1)) + (E(BD2_AVG) * (Fail2 * 2)) + (E(SYSBD_A) * 10 * ((1/Repair1 * 5) + (1/Repair2 * 6)) + d_1^- - d_1^+ = 126.16

$G_2 =$ E(DT_AVG) + d_2^- - d_2^+ = 0.10

$$G_3 = E(BD1\_AVG) + E(BD2\_AVG) + E(SYSBD\_A) + d_3^- - d_3^+ = 6.62$$

$$2 \le X_1 \le 15$$

$$1 \le X_2 \le 5$$

$$1 \le X_3 \le 10$$

$$2 \le X_4 \le 12$$

$$d_i^-, d_i^+ \ge 0, \qquad i=1,\dots,g$$

$$X_j \in \Re \text{ (rounded to the nearest hundredth)} \qquad j = 1,\dots,n$$

$$P_i = 100^{(g-i)}, \qquad i=1,\dots,g$$

where Z is the objective function to be minimized; g are the number of goals; n are the number of variables; T_i are the target values for the goals; P_i are the priority penalties; d_i^- and d_i^+ are non-negative deviational variables; X_1 is the mean time to failure for machine 1 (Fail1); X_2 is the mean time to failure for machine 2 (Fail2); X_3 is the mean repair time for machine 1 (Repair1); and X_4 is the mean repair time for machine 2 (Repair2).

4.2 Experimental Design

For each of the domains in the test bed, several cases were examined to evaluate the performance of the solution tools used. This was cast in a formal experimental design to test the significance of the various important parameters. The experimental design defines the independent variables or test factors, the levels of each independent variable, and the dependent variables or performance measures. The test environment must also be determined. Once all of this information is established, it is possible to develop a statistical model for the analysis of the data.

4.2.1 Independent Variables

The independent variables or test factors for this study were the domain and the multi-criteria simulation optimization method. Five different domains were examined. One test problem was examined for each domain. These problems were taken from the inventory, logistic, PERT, production, and reliability domains. The simulation optimization methods examined were: the genetic algorithms simulation optimization (GA-SO) method, the tabu search simulation optimization (TS-SO) method, and the Lexicographic Nelder-Mead simulation optimization (LNM-SO) method.

4.2.2 Dependent Variables

The dependent variables or performance measures for this computational study were the central processing unit (CPU) time required to complete the method (CPU_CM) and the CPU time required to find the best solution (CPU_BS), the representative operation count (ROC) of the number of calls to the simulation model required to complete the method (ROCCM) and the ROC of the number of calls to the simulation model required to find the best solution (ROCBS), the total number of solutions evaluated (Count), the best solution found (BSF), and the distance the initial solution for each run is from the BSF across all 60 runs within a domain (Init_Zdif).

The CPU time is the time the computer uses to solve each problem. After the user has entered the input for the problem into the PC, the time of the computer's internal clock is recorded. After every call to the simulation model and when the problem is completed, the time of the computer's internal clock is also recorded. The CPU time is calculated by taking the difference of these two times. Additionally, the CPU time required to find the best solution, is also recorded.

The representative operation counts (ROC) of simulation calls will count the number of calls to the simulation model. Simulating the model requires the most amount of time within the simulation optimization method. Thus, the fewer number of calls to the simulation that are required, the faster the simulation optimization method should perform. The ROC of simulation calls was recorded at two times: the number of simulation calls needed to reach the best solution (referred to as ROCBS) and the number of simulation calls needed to complete the simulation optimization method (referred to as ROCCM).

The simulation optimization methods were setup so that each solution would only be simulated once per run. However, it is possible that a solution would be evaluated multiple times. The count of the total number of solutions evaluated is referred to as Count.

The best solution found (BSF) is the solution that has the minimum objective function value (BSF_Z) for all runs. Since each run was started from a randomly generated initial solution, it was possible that this solution could be very close or very far from the BSF. Thus, the difference between the initial solution's objective function value (Init_Z) and BSF_Z was a very important test measure. This test measure was referred to as Init_Zdif.

4.2.3 Statistical Model

The statistical model is a 5x3 factorial design. The factors of this study are the independent variables. The equation representing this model is:

$$y = \mu + D_i + M_j + DM_{ij} + \varepsilon,$$

where y is CPU_BS, CPU_CM, ROCBS, ROCCM, Count, BSF, and Init_Zdif. μ is the overall mean, D represents the domain, M represents the simulation optimization method, DM is the interaction between D and M, and ε is the error term.

4.2.4 Test Runs

The experimental data are contained in a database composed of 300 lines of data, one line for each run, where a run is defined as one combination of a domain and a solution method (for example, D = Inventory and M = GA-SO, comprises one run). Twenty replications were performed for each test combination. The total number of runs was calculated as follows:

Total runs = (5 Domain Levels $*$ 3 Methods) $*$ 20 Replications = 300 runs.

4.2.5 Breakdown of Independent Variables

Domain: The five domains that were examined were:

1. Inventory (INV),
2. Logistics (LOG),
3. Project Management (PERT),
4. Production (PROD), and
5. Reliability (RELI).

Method: The three multi-criteria methods used were:

1. Genetic Algorithm Simulation Optimization (GA-SO),
2. Tabu Search Simulation Optimization (TS-SO), and
3. Lexicographic Nelder-Mead Simulation Optimization (LNM-SO).

4.2.6 Breakdown of Dependent Variables

The dependent variables examined were:

- Central processing unit (CPU) time to complete the simulation optimization method (CPU_CM),
- Central processing unit (CPU) time to find the best solution (CPU_BS)
- Representative operation counts to complete the simulation optimization method (ROCCM),
- Representative operation counts to find the best solution (ROCBS),
- The count of the total number of solutions evaluated (Count),
- The best solution found (BSF) and its corresponding objective function value (BSF_Z), and
- The difference between the objective function value of the initial solution (Init_Z) and the BSF (Init_Zdif).

A complete listing of the dependent variable and performance measure acronyms used throughout the remainder of this chapter is provided in Table 4.3.

4.2.7 Computational Equipment

The GA-SO method was programmed using C++ and used the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology (Copyright (c) 1995-1996 Massachusetts Institute of Technology (MIT) and Copyright (c) 1996-2005 Matthew Wall (the Author). All rights reserved). The TS-SO and the LNM-SO methods were programmed using MATLAB Version 7.0. The test problems were run in no specific order on one of several personal computers (PCs).

Several PCs used had a Pentium 4 processor, 1 GB RAM, and 3 GHz clock speed; one computer used had an AMD 64 Athlon™ 64 processor, 896 MB of RAM, and 2.21 GHz clock speed; and one PC had an Intel Celeron processor, 384 MB RAM, and 1.0 GHz clock speed. The data were analyzed using the Statistical Analysis System (SAS) for Windows Version 9.1 and Microsoft Excel 2003. All of the analysis was completed using a PC.

Table 4.3: List of Dependent Variable and Performance Measure Acronyms.

Acronym	Definition
BSF	The best solution found corresponding to the solution that generates the best objective function value.
BSF_Z	The objective function value of the best solution found.
Count	The count of the total solutions examined during the method.
CPU_BS	The central processing unit time required to find the best solution.
CPU_CM	The central processing unit time required to complete the simulation optimization method.
Init_Z	The objective function value of the randomly generated initial solution.
Init_Zdif	The difference between the objective function values of the best solution found and the initial solution.
ROCBS	The representative operation count of the number of calls to the simulation model required to find the best solution.
ROCCM	The representative operation count of the number of calls to the simulation model required to complete the simulation optimization method.
TQE	The time-quality estimator of the overall performance of the simulation optimization problem.

4.3 Computational Results

The results of the computational study were analyzed with respect to the computational speed, the quality of solution, and the overall performance for the simulation optimization method and the domain. The tools used to analyze the data included the analysis of variance (ANOVA), Duncan's Multiple Range Test, and correlation analysis. The complete SAS results are presented in Appendix C. Additionally, the robustness of the methods and test problems was evaluated based on the results of the computational study.

The null hypotheses of this study were that the main effects (independent variables) and the interaction effects (combinations of independent variables) were not statistically significant with respect to differences in each of the test measures (the dependent variables). Through analysis of variance (ANOVA), an F value was computed for each of the effects. The probability that this value will occur by chance given that the null hypothesis is true was calculated. If this probability is less than the level of significance, $\alpha = 0.05$, the null hypothesis is rejected and the effect is deemed statistically significant for the particular dependent variable.

Duncan's Multiple Range Test is a method that compares the means of different levels of a factor for each of the test measures. Means that are not significantly different are represented by the same letter. All of the tests use a level of significance of $\alpha = 0.05$.

4.3.1 Test Data

A summary of the test results described by the test factors (independent variables), test measures (dependent variables), and additional measures are given in Table 4.4. The average data, across twenty replications, is presented for each method-domain combination. A breakdown of each run is provided in Table 4.5 through Table 4.9. These tables are provided as a reference source for the rest of this chapter. The complete summary data from all 300 runs can be found in Appendix D.

Table 4.4: Average Output Summary.

Method	Domain	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	Init_Zdif
GA-SO	INV	1947.35	159.85	1035.75	103.50	644.40	470.81	4399.13
GA-SO	LOG	2020.05	534.95	3241.80	408.60	2413.05	211.25	4.21
GA-SO	PERT	1943.15	155.95	366.50	118.30	246.85	0.13	0.0146
GA-SO	PROD	2071.15	533.70	7672.25	302.95	4315.20	17.86	33.80
GA-SO	RELI	1984.10	335.05	3001.55	225.65	2057.15	3.16	30234.18
TS-SO	INV	1188.00	357.10	2900.14	253.70	2050.07	687.19	4211.37
TS-SO	LOG	1980.00	190.60	1644.38	89.95	762.12	301.93	167.80
TS-SO	PERT	1188.00	72.15	891.21	33.05	400.75	0.13	0.0130
TS-SO	PROD	2376.00	262.00	3390.35	110.70	1410.82	26.08	33.30
TS-SO	RELI	1584.00	151.20	1924.69	75.50	948.33	823.91	21636.89
LNM-SO	INV	879.60	134.90	881.44	75.10	498.77	386.43	4311.69
LNM-SO	LOG	891.95	34.10	155.45	23.40	102.18	278.90	197.93
LNM-SO	PERT	693.95	25.25	111.53	17.65	74.75	0.13	0.0188
LNM-SO	PROD	980.70	55.90	274.01	30.65	138.86	17.66	35.12
LNM-SO	RELI	867.95	40.95	196.90	30.75	143.72	45.22	18763.04

4.3.2 Computational Speed

For this study, five test measures were examined to characterize the speed of computation. These measures were: the time required to complete the simulation optimization method (CPU_CM), the number of calls to the simulation model required to complete the simulation optimization method (ROCCM), the time required to find the best solution (CPU_BS), the representative operation count of the number of calls to the simulation model required to find the best solution (ROCBS), and the count of the total number of solutions evaluated (Count).

Next, the comparison of the CPU time and representative operation counts of the number of calls to the simulation model required to complete the method and to find the best solution will be discussed.

Table 4.5: Inventory Domain Output Summary.

Method	Domain	Run	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z
GA-SO	INV	1	1956	196	1283	166	1056	640.29
GA-SO	INV	2	1956	151	978	62	383	1075.70
GA-SO	INV	3	1968	157	1035	128	810	782.91
GA-SO	INV	4	1905	131	860	97	605	114.41
GA-SO	INV	5	1859	163	1055	138	857	114.41
GA-SO	INV	6	1945	148	974	119	748	672.56
GA-SO	INV	7	1965	184	1196	132	826	640.29
GA-SO	INV	8	1958	152	985	68	421	114.41
GA-SO	INV	9	1957	131	867	95	597	640.29
GA-SO	INV	10	1932	211	1360	150	934	640.29
GA-SO	INV	11	1917	136	873	46	286	640.29
GA-SO	INV	12	1976	116	758	78	480	114.41
GA-SO	INV	13	1973	167	1075	126	777	782.91
GA-SO	INV	14	1983	205	1308	168	1037	672.56
GA-SO	INV	15	1941	136	877	51	313	672.56
GA-SO	INV	16	1897	139	900	95	586	114.41
GA-SO	INV	17	1919	189	1217	161	1002	114.41
GA-SO	INV	18	1974	176	1122	12	74	114.41
GA-SO	INV	19	1965	155	1003	122	752	640.29
GA-SO	INV	20	2001	154	989	56	344	114.41
LNM-SO	INV	1	790	145	914	60	375	348.68
LNM-SO	INV	2	940	197	1243	108	674	234.18
LNM-SO	INV	3	1084	145	918	117	729	337.65
LNM-SO	INV	4	730	101	638	7	44	172.41
LNM-SO	INV	5	845	73	779	69	746	148.83
LNM-SO	INV	6	743	42	268	37	231	690.13
LNM-SO	INV	7	848	181	1140	75	467	540.71
LNM-SO	INV	8	730	35	224	15	94	148.83
LNM-SO	INV	9	994	195	1226	125	778	271.15
LNM-SO	INV	10	735	198	1240	14	91	1131.05
LNM-SO	INV	11	1009	189	1184	115	712	242.03
LNM-SO	INV	12	715	24	155	5	31	706.01
LNM-SO	INV	13	1028	226	1430	74	460	395.54
LNM-SO	INV	14	967	195	1227	170	1059	185.15
LNM-SO	INV	15	1037	185	1474	182	1443	234.25
LNM-SO	INV	16	713	35	222	8	50	549.91
LNM-SO	INV	17	707	29	186	28	174	493.62
LNM-SO	INV	18	918	199	1250	31	192	526.24
LNM-SO	INV	19	1184	195	1228	177	1100	228.35
LNM-SO	INV	20	875	109	683	85	525	143.98
TS-SO	INV	1	1188	265	1707	116	724	521.68
TS-SO	INV	2	1188	398	2566	329	2091	522.60
TS-SO	INV	3	1188	641	4052	496	3131	480.64
TS-SO	INV	4	1188	245	1568	146	914	428.10
TS-SO	INV	5	1188	344	4841	264	3702	1308.63
TS-SO	INV	6	1188	303	4220	204	2841	451.70
TS-SO	INV	7	1188	542	7303	415	5606	436.71
TS-SO	INV	8	1188	236	3127	88	1161	347.08
TS-SO	INV	9	1188	380	4969	316	4125	470.09
TS-SO	INV	10	1188	241	1542	207	1289	1602.96
TS-SO	INV	11	1188	161	1036	83	514	296.62
TS-SO	INV	12	1188	471	2960	358	2249	516.88
TS-SO	INV	13	1188	346	2191	226	1404	303.77
TS-SO	INV	14	1188	573	3612	452	2837	1239.72
TS-SO	INV	15	1188	372	2388	318	2015	1040.54
TS-SO	INV	16	1188	273	1664	125	747	516.27
TS-SO	INV	17	1188	136	840	83	505	634.93
TS-SO	INV	18	1188	138	849	125	756	448.41
TS-SO	INV	19	1188	575	3504	300	1817	1150.85
TS-SO	INV	20	1188	502	3065	423	2573	1025.57

Table 4.6: Logistic Domain Output Summary.

Method	Domain	Run	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z
GA-SO	LOG	1	2022	544	9378	459	7666	211.25
GA-SO	LOG	2	2060	550	8882	407	6382	211.25
GA-SO	LOG	3	1994	580	7754	369	4829	211.25
GA-SO	LOG	4	1964	642	8223	544	6940	211.25
GA-SO	LOG	5	2049	488	6187	341	4278	211.25
GA-SO	LOG	6	2031	604	3001	461	2235	211.25
GA-SO	LOG	7	2027	509	1645	356	1114	211.25
GA-SO	LOG	8	1973	542	1760	450	1424	211.25
GA-SO	LOG	9	2047	456	1461	265	811	211.25
GA-SO	LOG	10	2037	620	1997	483	1520	211.25
GA-SO	LOG	11	2048	564	1532	414	1083	211.25
GA-SO	LOG	12	2068	501	1407	333	895	211.25
GA-SO	LOG	13	2027	436	1242	339	926	211.25
GA-SO	LOG	14	2028	580	1634	357	972	211.25
GA-SO	LOG	15	1990	594	1675	523	1458	211.25
GA-SO	LOG	16	2006	560	1575	510	1408	211.25
GA-SO	LOG	17	2032	505	1438	435	1207	211.25
GA-SO	LOG	18	2058	467	1324	402	1113	211.25
GA-SO	LOG	19	1978	525	1487	356	969	211.25
GA-SO	LOG	20	1962	432	1234	368	1031	211.25
LNM-SO	LOG	1	894	28	130	14	63	278.90
LNM-SO	LOG	2	891	32	146	14	61	278.90
LNM-SO	LOG	3	890	39	177	31	136	278.90
LNM-SO	LOG	4	893	43	193	15	65	278.90
LNM-SO	LOG	5	886	39	176	38	165	278.90
LNM-SO	LOG	6	884	48	215	43	186	278.91
LNM-SO	LOG	7	894	23	106	14	61	278.90
LNM-SO	LOG	8	893	21	97	19	82	278.90
LNM-SO	LOG	9	892	30	137	28	121	278.90
LNM-SO	LOG	10	891	26	120	20	88	278.90
LNM-SO	LOG	11	892	38	171	9	39	278.90
LNM-SO	LOG	12	893	28	128	24	104	278.90
LNM-SO	LOG	13	897	39	176	14	61	278.90
LNM-SO	LOG	14	893	30	136	28	121	278.90
LNM-SO	LOG	15	886	39	176	38	164	278.90
LNM-SO	LOG	16	891	33	152	28	123	278.90
LNM-SO	LOG	17	905	53	241	14	62	278.90
LNM-SO	LOG	18	893	27	126	26	115	278.90
LNM-SO	LOG	19	894	29	135	15	66	278.90
LNM-SO	LOG	20	887	37	170	36	159	278.90
TS-SO	LOG	1	1980	181	860	107	478	290.64
TS-SO	LOG	2	1980	117	577	26	115	289.84
TS-SO	LOG	3	1980	202	949	67	294	288.55
TS-SO	LOG	4	1980	190	899	109	483	292.23
TS-SO	LOG	5	1980	175	833	48	213	308.03
TS-SO	LOG	6	1980	162	771	93	413	292.45
TS-SO	LOG	7	1980	164	784	76	338	313.73
TS-SO	LOG	8	1980	153	736	50	223	316.93
TS-SO	LOG	9	1980	179	848	102	452	294.77
TS-SO	LOG	10	1980	121	592	38	169	308.97
TS-SO	LOG	11	1980	208	2288	129	1408	311.64
TS-SO	LOG	12	1980	190	2048	111	1180	294.84
TS-SO	LOG	13	1980	374	4553	167	1807	304.51
TS-SO	LOG	14	1980	221	2415	161	1724	312.28
TS-SO	LOG	15	1980	184	2061	106	1165	294.28
TS-SO	LOG	16	1980	185	2114	45	492	287.11
TS-SO	LOG	17	1980	172	2013	49	570	313.71
TS-SO	LOG	18	1980	210	2489	99	1184	302.63
TS-SO	LOG	19	1980	226	2717	135	1594	313.97
TS-SO	LOG	20	1980	198	2343	81	937	307.51

Table 4.7: PERT Domain Output Summary.

Method	Domain	Run	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z
GA-SO	PERT	1	1955	123	277	64	128	0.1324
GA-SO	PERT	2	1926	170	415	122	266	0.1324
GA-SO	PERT	3	1906	115	308	87	194	0.1324
GA-SO	PERT	4	1898	160	350	123	238	0.1324
GA-SO	PERT	5	1955	123	280	123	280	0.1324
GA-SO	PERT	6	1950	163	360	135	262	0.1324
GA-SO	PERT	7	1879	158	346	126	245	0.1324
GA-SO	PERT	8	1961	140	317	98	192	0.1324
GA-SO	PERT	9	1975	170	370	119	231	0.1324
GA-SO	PERT	10	1871	162	350	98	191	0.1324
GA-SO	PERT	11	1945	140	310	99	192	0.1324
GA-SO	PERT	12	2046	186	410	135	265	0.1324
GA-SO	PERT	13	1997	143	324	115	225	0.1324
GA-SO	PERT	14	1957	134	315	109	217	0.1324
GA-SO	PERT	15	1928	163	411	127	281	0.1324
GA-SO	PERT	16	1935	195	488	158	352	0.1324
GA-SO	PERT	17	2005	195	490	166	372	0.1324
GA-SO	PERT	18	2007	172	428	126	279	0.1324
GA-SO	PERT	19	1839	133	342	99	222	0.1324
GA-SO	PERT	20	1928	174	439	137	305	0.1324
LNM-SO	PERT	1	696	17	78	10	44	0.1324
LNM-SO	PERT	2	698	11	51	7	30	0.1324
LNM-SO	PERT	3	699	9	43	7	30	0.1324
LNM-SO	PERT	4	695	19	86	10	43	0.1324
LNM-SO	PERT	5	696	17	76	10	42	0.1324
LNM-SO	PERT	6	673	55	237	19	81	0.1324
LNM-SO	PERT	7	698	11	51	7	29	0.1324
LNM-SO	PERT	8	698	11	51	7	30	0.1324
LNM-SO	PERT	9	695	22	98	10	42	0.1324
LNM-SO	PERT	10	697	13	59	7	29	0.1324
LNM-SO	PERT	11	676	154	653	153	644	0.1324
LNM-SO	PERT	12	695	22	98	15	64	0.1324
LNM-SO	PERT	13	698	11	51	7	29	0.1324
LNM-SO	PERT	14	695	17	77	9	38	0.1324
LNM-SO	PERT	15	692	25	110	9	38	0.1324
LNM-SO	PERT	16	694	24	106	21	89	0.1324
LNM-SO	PERT	17	697	16	72	12	50	0.1324
LNM-SO	PERT	18	696	15	68	7	30	0.1324
LNM-SO	PERT	19	694	21	98	16	72	0.1324
LNM-SO	PERT	20	697	15	67	10	42	0.1324
TS-SO	PERT	1	1188	58	700	32	378	0.1341
TS-SO	PERT	2	1188	106	1256	53	609	0.1325
TS-SO	PERT	3	1188	94	1108	47	564	0.1336
TS-SO	PERT	4	1188	89	1070	52	616	0.1337
TS-SO	PERT	5	1188	51	630	22	261	0.1334
TS-SO	PERT	6	1188	78	978	24	289	0.1328
TS-SO	PERT	7	1188	76	883	35	398	0.1331
TS-SO	PERT	8	1188	60	712	23	266	0.1344
TS-SO	PERT	9	1188	64	774	21	249	0.1338
TS-SO	PERT	10	1188	62	750	27	314	0.1337
TS-SO	PERT	11	1188	63	780	27	330	0.1348
TS-SO	PERT	12	1188	55	695	16	201	0.1338
TS-SO	PERT	13	1188	47	601	14	171	0.1348
TS-SO	PERT	14	1188	70	860	19	230	0.1331
TS-SO	PERT	15	1188	86	1083	46	569	0.1336
TS-SO	PERT	16	1188	95	1209	54	682	0.1342
TS-SO	PERT	17	1188	73	942	32	400	0.1339
TS-SO	PERT	18	1188	52	669	14	174	0.1344
TS-SO	PERT	19	1188	73	953	52	670	0.1330
TS-SO	PERT	20	1188	91	1170	51	647	0.1341

Table 4.8: Production Domain Output Summary.

Method	Domain	Run	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z
GA-SO	PROD	1	2102	605	8103	5	84	17.86
GA-SO	PROD	2	2025	477	6338	296	3877	17.86
GA-SO	PROD	3	2061	525	7621	331	4951	17.86
GA-SO	PROD	4	2073	571	7665	359	4775	17.86
GA-SO	PROD	5	2004	461	6157	279	3686	17.86
GA-SO	PROD	6	2134	525	6683	347	4360	17.86
GA-SO	PROD	7	2069	556	7510	386	5146	17.86
GA-SO	PROD	8	2097	512	6863	186	2424	17.86
GA-SO	PROD	9	2106	538	7308	310	4135	17.86
GA-SO	PROD	10	2010	473	6331	262	3437	17.86
GA-SO	PROD	11	2071	549	7455	356	4723	17.86
GA-SO	PROD	12	2004	585	7371	424	5317	17.86
GA-SO	PROD	13	2082	513	6655	281	3585	17.86
GA-SO	PROD	14	2147	437	5848	199	2564	17.86
GA-SO	PROD	15	2126	538	7297	273	3666	17.86
GA-SO	PROD	16	2100	587	7825	351	4611	17.86
GA-SO	PROD	17	2107	455	6087	312	4124	17.86
GA-SO	PROD	18	2033	617	11427	334	5801	17.86
GA-SO	PROD	19	1970	522	11105	324	7167	17.86
GA-SO	PROD	20	2102	628	11796	444	7871	17.86
LNM-SO	PROD	1	986	47	232	36	165	17.50
LNM-SO	PROD	2	974	82	388	58	263	17.61
LNM-SO	PROD	3	980	61	292	53	240	17.21
LNM-SO	PROD	4	983	60	289	10	45	17.86
LNM-SO	PROD	5	981	45	232	45	205	17.86
LNM-SO	PROD	6	966	87	416	73	331	17.39
LNM-SO	PROD	7	982	53	263	45	204	17.21
LNM-SO	PROD	8	987	35	188	10	45	17.86
LNM-SO	PROD	9	981	60	288	10	44	17.86
LNM-SO	PROD	10	980	51	259	46	209	17.47
LNM-SO	PROD	11	981	59	294	53	240	17.39
LNM-SO	PROD	12	981	58	279	10	45	17.86
LNM-SO	PROD	13	980	57	275	10	45	17.86
LNM-SO	PROD	14	987	37	199	34	156	17.41
LNM-SO	PROD	15	984	52	252	10	44	17.86
LNM-SO	PROD	16	986	51	248	10	45	17.86
LNM-SO	PROD	17	975	46	237	35	159	17.71
LNM-SO	PROD	18	980	60	288	10	45	17.86
LNM-SO	PROD	19	979	59	284	10	45	17.86
LNM-SO	PROD	20	981	58	279	45	203	17.64
TS-SO	PROD	1	2376	218	2830	114	1466	28.70
TS-SO	PROD	2	2376	261	3360	139	1775	24.50
TS-SO	PROD	3	2376	339	4352	180	2270	24.53
TS-SO	PROD	4	2376	205	2633	111	1406	27.48
TS-SO	PROD	5	2376	167	2197	88	1132	26.36
TS-SO	PROD	6	2376	191	2497	85	1094	25.88
TS-SO	PROD	7	2376	341	4434	58	752	24.98
TS-SO	PROD	8	2376	396	5120	137	1760	24.37
TS-SO	PROD	9	2376	427	5462	161	2064	27.80
TS-SO	PROD	10	2376	257	3213	138	1695	24.50
TS-SO	PROD	11	2376	365	4683	136	1724	27.14
TS-SO	PROD	12	2376	166	2153	76	970	27.51
TS-SO	PROD	13	2376	258	3322	127	1602	25.98
TS-SO	PROD	14	2376	158	2095	33	433	25.52
TS-SO	PROD	15	2376	403	5262	131	1679	28.34
TS-SO	PROD	16	2376	257	3384	149	1938	27.66
TS-SO	PROD	17	2376	254	3358	107	1388	26.02
TS-SO	PROD	18	2376	186	2426	37	456	24.04
TS-SO	PROD	19	2376	171	2207	95	1203	26.77
TS-SO	PROD	20	2376	220	2820	112	1410	23.55

Table 4.9: Reliability Domain Output Summary.

Method	Domain	Run	Count	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z
GA-SO	RELI	1	1970	348	8056	261	6064	2.95
GA-SO	RELI	2	2013	353	6799	235	4839	2.95
GA-SO	RELI	3	1948	271	5077	188	3243	2.95
GA-SO	RELI	4	1986	238	3155	143	1851	2.95
GA-SO	RELI	5	1957	345	4561	223	2908	2.95
GA-SO	RELI	6	2007	351	4454	230	2850	3.54
GA-SO	RELI	7	1980	344	3419	298	2960	2.95
GA-SO	RELI	8	1984	397	3805	289	2764	2.95
GA-SO	RELI	9	1987	388	3586	203	1849	2.95
GA-SO	RELI	10	1989	341	3108	263	2351	2.95
GA-SO	RELI	11	1944	295	3350	227	2572	3.54
GA-SO	RELI	12	1995	376	4137	268	2903	2.95
GA-SO	RELI	13	1975	370	858	196	422	2.95
GA-SO	RELI	14	1958	342	777	252	533	3.54
GA-SO	RELI	15	1981	332	787	244	547	3.54
GA-SO	RELI	16	1937	340	797	191	418	2.95
GA-SO	RELI	17	1986	265	636	187	412	3.54
GA-SO	RELI	18	2007	370	1191	243	855	3.54
GA-SO	RELI	19	2020	315	730	173	372	2.95
GA-SO	RELI	20	2058	320	748	199	430	3.54
LNM-SO	RELI	1	830	27	121	17	73	54.01
LNM-SO	RELI	2	829	20	90	16	67	68.39
LNM-SO	RELI	3	860	28	124	25	105	15.75
LNM-SO	RELI	4	826	19	86	15	63	67.12
LNM-SO	RELI	5	794	19	86	18	76	137.78
LNM-SO	RELI	6	819	17	78	15	63	84.45
LNM-SO	RELI	7	1193	194	830	122	510	32.56
LNM-SO	RELI	8	830	25	112	17	72	73.70
LNM-SO	RELI	9	832	17	78	15	63	30.88
LNM-SO	RELI	10	1142	165	706	136	569	3.15
LNM-SO	RELI	11	806	35	157	32	137	38.88
LNM-SO	RELI	12	852	36	162	32	138	50.22
LNM-SO	RELI	13	836	27	122	18	77	34.82
LNM-SO	RELI	14	851	24	109	22	94	23.97
LNM-SO	RELI	15	844	33	147	19	81	27.55
LNM-SO	RELI	16	838	24	109	17	73	31.28
LNM-SO	RELI	17	845	30	135	21	89	69.16
LNM-SO	RELI	18	830	25	116	17	73	19.31
LNM-SO	RELI	19	846	24	327	19	275	25.74
LNM-SO	RELI	20	856	30	244	22	176	15.75
TS-SO	RELI	1	1584	114	1452	45	563	1536.71
TS-SO	RELI	2	1584	160	2030	59	728	1330.08
TS-SO	RELI	3	1584	140	1769	67	835	1286.31
TS-SO	RELI	4	1584	163	1998	98	1199	85.82
TS-SO	RELI	5	1584	124	1502	40	465	112.54
TS-SO	RELI	6	1584	113	1413	55	673	187.77
TS-SO	RELI	7	1584	172	2154	131	1624	1664.63
TS-SO	RELI	8	1584	166	2065	89	1084	654.79
TS-SO	RELI	9	1584	149	1889	79	983	1862.33
TS-SO	RELI	10	1584	150	1932	84	1083	883.76
TS-SO	RELI	11	1584	147	1846	88	1090	55.97
TS-SO	RELI	12	1584	179	2204	93	1133	327.28
TS-SO	RELI	13	1584	114	1405	40	483	945.94
TS-SO	RELI	14	1584	115	1450	40	499	1608.27
TS-SO	RELI	15	1584	240	3209	96	1313	187.89
TS-SO	RELI	16	1584	185	2406	83	1058	126.64
TS-SO	RELI	17	1584	124	1614	65	807	1718.93
TS-SO	RELI	18	1584	160	2076	76	980	1666.63
TS-SO	RELI	19	1584	154	2004	89	1137	233.39
TS-SO	RELI	20	1584	155	2074	93	1229	2.54

4.3.2.1 Comparing CPU Time and Number of Representative Operation Counts

One of the desires of this research was to be able to replace CPU time with the representative operation count (ROC) of the number of calls to the simulation model. Specifically, these values were calculated for the time required to **c**omplete the simulation optimization **m**ethod (CM) and the time required to find the best solution (BS). The intercorrelations of ROC and CPU time were calculated from one run examining the corresponding ROC and CPU time. These intercorrelation results are presented in Table 4.10. The closer the value is to 1.0 or –1.0, the closer the two measures are related. From the table, it is clear that CPU time and ROC were highly correlated.

Table 4.10: Correlation of ROC with CPU.

	ROC	CPU Time
ROC	---	0.99961
CPU Time	0.99961	---

Similarly, a scatter plot of CPU time versus ROC is presented in Figure 4.6. The figure shows an almost perfect forty-five degree line with its intercept occurring at zero. The figure also shows the trend line and the R-squared value of 0.9991. The trend line shows how closely the projected values are to the actual data. The closer the R-squared value is to one, the more reliable the trend line is. Thus, since the R-squared value equals 0.9991, the trend line is very accurate.

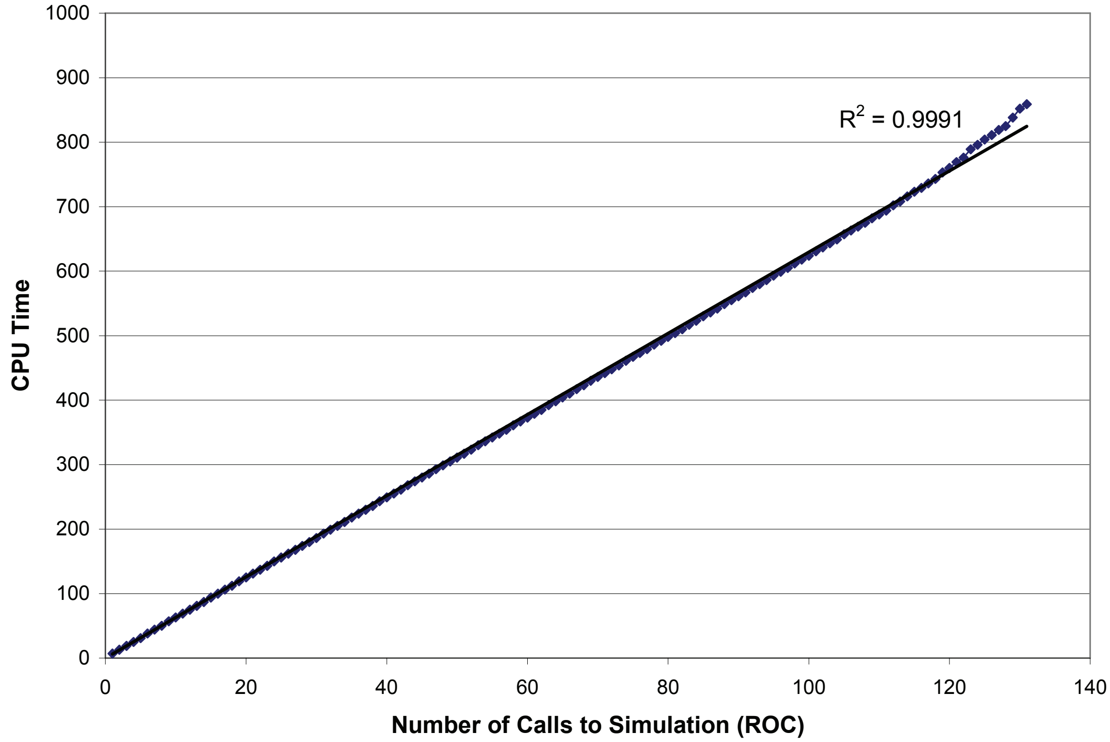


Figure 4.6: Scatter Plot of ROC and CPU Time.

These results support the goal of the study to be able to eliminate CPU time as one of the comparison measures and to replace it with the more independent ROC of the calls to the simulation model. Following these findings, the remainder of the discussion of the results will not include CPU time. Instead, the computational speed parameters will focus on ROCCM and ROCBS.

4.3.2.2 Analysis of ROCCM

ROCCM is one of the computational effort test measures that were used to compare the different runs and ultimately the different methods. These data are presented through graphs of the mean ROCCM for the interaction effects of the method versus the domain, an ANOVA table, and Duncan's Multiple Range tests.

The two-way interaction effects were plotted for ROCCM. Figure 4.7 shows the effect of the method and the domain on ROCCM. The ROCCM followed a very similar pattern for the logistic, PERT, production, and reliability domains. The number of ROCCM in all these cases was the greatest (by a wide margin) for the GA-SO method, then the TS-SO method (which was much greater than the LNM-SO method), and the least for the LNM-SO method. For the inventory domain, the LNM-SO method still required the least number of ROCCM, but by a substantial margin, TS-SO required the most. Ideally, the number of calls to the simulation should be as small as possible, so the computational speed would be faster. Thus, based on the discussion above, the LNM-SO method proved to be the best simulation optimization method in terms of ROCCM. Additionally, it can be seen that the PERT domain required the least amount of ROCCM for all three methods.

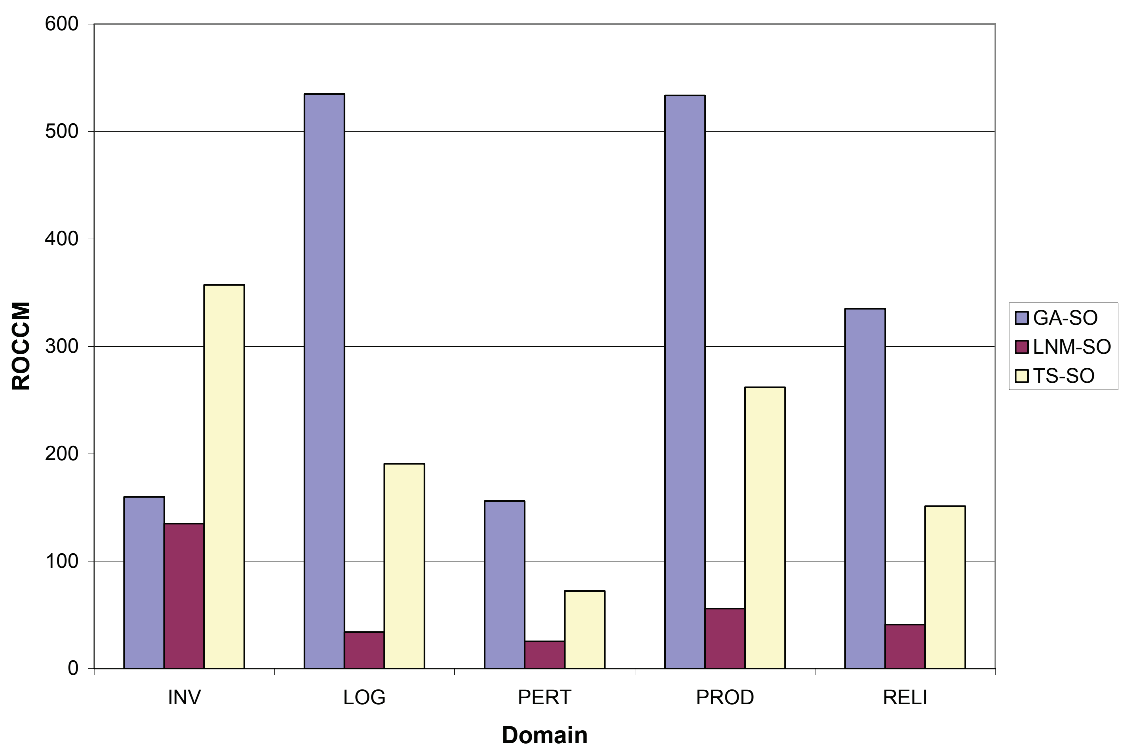


Figure 4.7: Mean ROCCM by Domain and Method.

The ANOVA results for ROCCM are presented in Table 4.11. The significant sources (as indicated by "Pr > F" having a value less than 0.05) are shown in boldface type. The domain, simulation optimization method, and the interaction of the domain and the simulation optimization method were all statistically significant. This is to be expected, since there is not a reliable way to compare across domains and there was a large difference between each of the methods.

Table 4.11: ANOVA Table for ROCCM.

Source	Degrees of Freedom	Type III Sum of Squares	Mean Square	F Value	Pr > F
Domain	4	1443760.687	360940.172	102.47	**<.0001**
Method	2	4082706.620	2041353.310	579.52	**<.0001**
Method*Domain	8	2491753.213	311469.152	88.42	**<.0001**

Table 4.12 shows the Duncan Multiple Range Test results for the independent variables with respect to ROCCM by domain and method. Recall that factors identified by the same letter are not significantly different. From Table 4.12, it is clear that there was not a significant difference between each domain or between each method. Again, based on Figure 4.7, this was expected.

Table 4.12: Duncan's Multiple Range Test for ROCCM.

Domain

Duncan Grouping	Mean	# of Observations	Domain
A	283.87	60	PROD
B	253.22	60	LOG
C	217.28	60	INV
D	175.73	60	RELI
E	84.45	60	PERT

Method

Duncan Grouping	Mean	# of Observations	Method
A	343.90	100	GA-SO
B	206.61	100	TS-SO
C	58.22	100	LNM-SO

It is also important to examine the percentage of the solution space size that was examined for each problem and method. This is calculated by dividing the ROCCM by the solution space size and multiplying this value by one hundred. Since the solution space size is so much larger than the ROCCM, this generates very small numbers. Thus a scaling factor of 1,000,000 was used. This was plotted in Figure 4.8. As can be seen from the figure, only the PERT domain had any real difference. The GA-SO examined the greatest percentage of the solution space, while LNM-SO examined the least. However, considering the extremely large scaling factor, it becomes clear that for all of the domains, the methods examined only covered a very small percentage of the solution space.

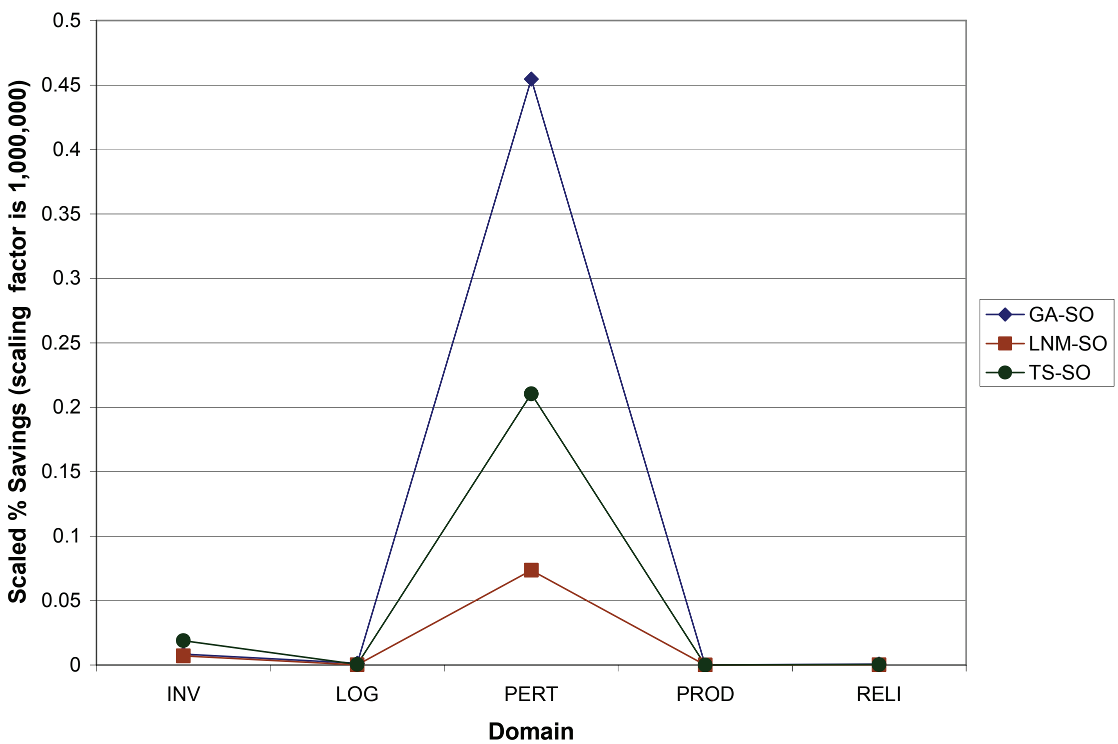


Figure 4.8: Scaled Percent Savings for Solution Space Size.

4.3.2.3 Analysis of ROCBS

ROCBS is another one of the computational speed test measures that were used to compare the different runs and ultimately the different methods. These data are presented through graphs of the mean ROCBS for the interaction effects of the method versus the domain, an ANOVA table, and Duncan's Multiple Range tests.

The two-way interaction effects were plotted for ROCBS. Figure 4.9 shows the effect of the method and the domain on the mean ROCBS. Consistent with ROCCM, the ROCBS followed a very similar pattern for the logistic, PERT, production, and reliability domains. The number of ROCBS in all these cases was the greatest (by a wide margin) for the GA-SO method, then the TS-SO method (which was much greater than the LNM-SO method), and the least for the LNM-SO method. For the inventory domain, the LNM-SO method still required the least number of ROCBS, but by a substantial margin, TS-SO required the most. Additionally, we see that the PERT domain required the least amount of ROCBS for all three methods.

Thus, it can be seen that the ranking of the methods for each domain for ROCBS were the same as the rankings of ROCCM. Each of the simulation optimization methods was terminated once it converged to 0.99 of the best objective function value found from each iteration. Thus, it follows that the fewer number of calls required to find the best solution would generally result in the fewer number of calls required to complete the simulation optimization method. Thus, it makes logical sense that the results for ROCBS should mirror the results of ROCCM.

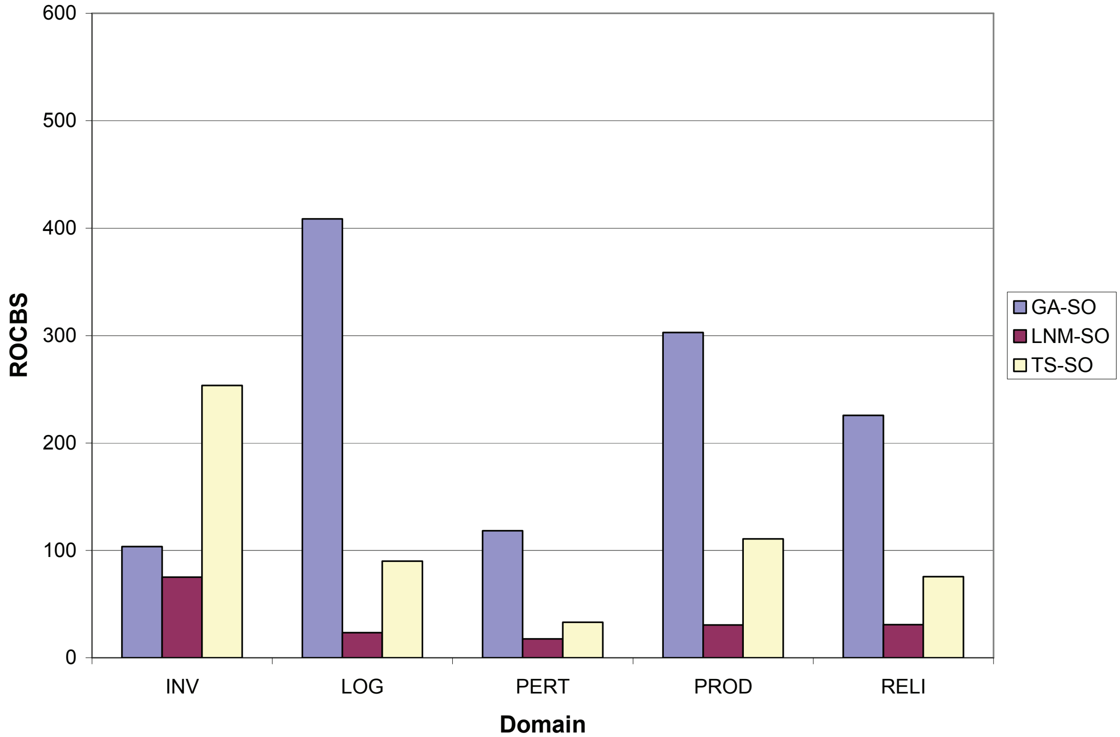


Figure 4.9: Mean ROCBS by Domain and Method.

The ANOVA results for ROCBS are presented in Table 4.13. The significant sources (as indicated by "Pr > F" having a value less than 0.05) are shown in boldface type. The domain, the simulation optimization method, and the interaction of the domain and the simulation optimization method were all statistically significant. This was to be expected, since there is not a reliable way to compare across domains and there was a large difference between each of the methods.

Table 4.13: ANOVA Table for ROCBS.

Source	Degrees of Freedom	Type III Sum of Squares	Mean Square	F Value	Pr > F
Domain	4	492360.880	123090.220	39.93	**<.0001**
Method	2	1956098.580	978049.290	317.27	**<.0001**
Method*Domain	8	1425864.220	178233.028	57.82	**<.0001**

Table 4.14 shows the Duncan Multiple Range Test results for the independent variables with respect to ROCBS by domain and method. Recall that factors identified by the same letter are not significantly different. From Table 4.14, it is clear that there was a significant difference between each of the domains, as expected. However, the means of the production and inventory domains were not significantly different. But after further breakdown, these appear not to be related. Examining Figure 4.9, this appears to be a statistical mirage, since the results are very different for each method within these domains. It appears that the overall average for these just happens to balance out to similar means.

Table 4.14: Duncan's Multiple Range Test for ROCBS.

Duncan Grouping	Mean	# of Observations	Domain
Domain			
A	173.98	60	LOG
B	148.10	60	PROD
B	144.10	60	INV
C	110.63	60	RELI
D	56.33	60	PERT

Duncan Grouping	Mean	# of Observations	Method
Method			
A	231.80	100	GA-SO
B	112.58	100	TS-SO
C	35.51	100	LNM-SO

To attempt to verify whether this is in fact a statistical mirage, the mean ROCBS for each method for the inventory and production domains were examined in Table 4.15. From the table it is obvious that the means for both domains are not similar at all for each method. It can further be verified that the overall means are very similar. Thus, this supports the hypothesis that the fact that the means do not appear to be statistically different for the inventory and production domains is just a statistical mirage. In fact, these domains are in essence statistically different.

Table 4.15: Breakdown of the Means for ROCBS for the Inventory and Production Domains.

	Method			
Domain	**GA-SO**	**LNM-SO**	**TS-SO**	**Overall Mean**
INV	103.5	75.1	253.7	144.1
PROD	302.95	30.65	110.7	148.1

4.3.2.3 Additional Analysis

Another variable of interest is the number of solutions examined to complete the simulation optimization method. This is referred to as Count. Note that each solution will only require one call to the simulation model, however, each solution maybe examined multiple times in the optimization method. These data are presented through graphs of the mean Count for the interaction effects of the method versus the domain, an ANOVA table, and Duncan's Multiple Range tests.

Recall from the discussion in section 3.1.2.1 that the maximum number of solutions that could be examined (the maximum Count value) for each method could be determined. The maximum values of Count based on the number of variables, n, are shown in Table 4.16. Recall that for the GA-SO method, the average number of solutions evaluated per iteration would be less than thirty, since solutions that are passed directly from one generation to the next (i.e., the best solution from that generation) were not re-evaluated). Since the maximum number of variables examined in this study was six, it would be expected that GA-SO would have the largest value of Count for all domains, followed by TS-SO, and then by LNM-SO.

Table 4.16: Maximum Value of Count per Iteration.

Method	Max Count per Iteration
GA-SO	30
LNM-SO	$n+4$
TS-SO	$4n+1$

The two-way interaction effects were plotted for Count. Figure 4.10 shows the effect of the method and the domain on the mean Count. In each case, the LNM-SO method required the smallest value of Count, while TS-SO required the second smallest value for all but the production domain. The GA-SO method required the largest value of Count for all but the production domain. These results support the predictions made based on Table 4.16

The variability of the values of Count between domains for each method can also be evaluated by examining Figure 4.10. The GA-SO method required approximately an average Count of 2000 across all domains. There was some variability for the LNM-SO method,. while, the TS-SO method had the most variability. This was expected, since the maximum number of Count per iteration for the GA-SO method was not dependent on the number of variables, while LNM-SO was dependent on the number of variables, and TS-SO was the most dependent method on the number of variables.

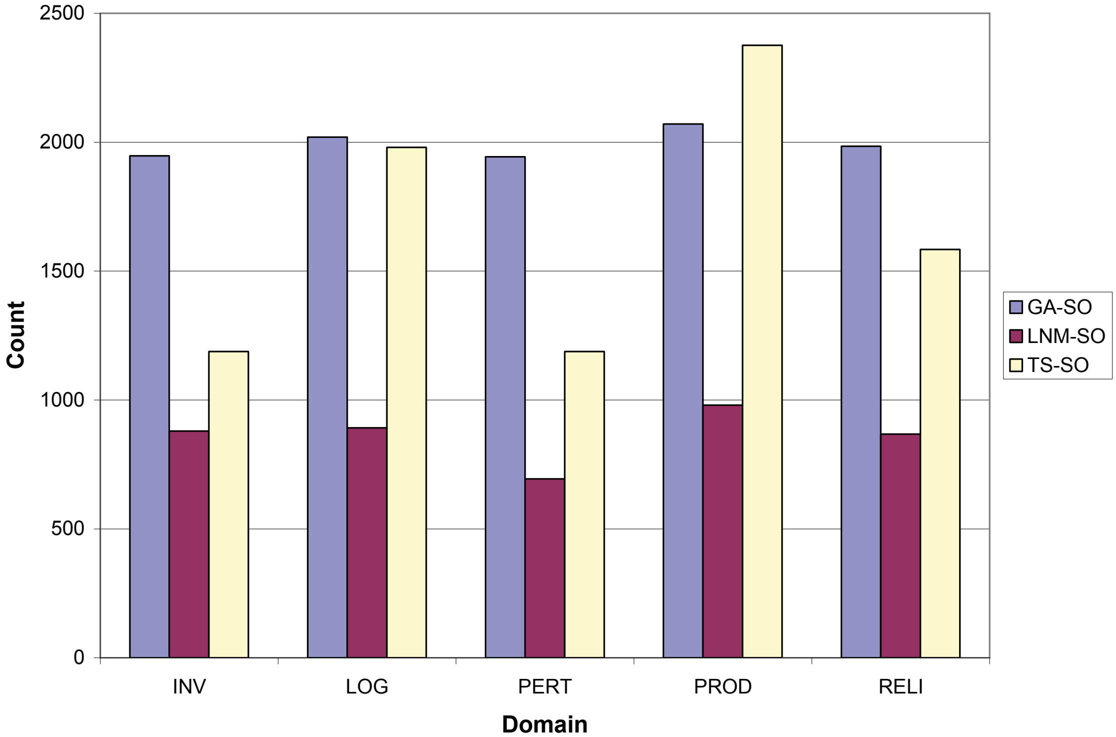


Figure 4.10: Mean Count by Domain and Method.

The TS-SO method required the exact same value of Count for each run within a domain (see Tables 4.5 – 4.9). This was because of the way the neighborhoods were setup within the method and the fact that each run required the same number of iterations. Thus, the Count required was strictly determined by the number of variables examined in the domain. The number of iterations varied for LNM-SO, thus the Count values were similar but not identical for each run within a domain.

The ANOVA results for Count are presented in Table 4.17. The significant sources (as indicated by “Pr > F” having a value less than 0.05) are shown in boldface type. The domain, the simulation optimization method, and the interaction of the domain and the simulation optimization method were all statistically significant. This is to be expected, since there is not a reliable way to compare across domains and there was a large difference between each of the methods.

Table 4.17: ANOVA Table for Count.

Source	Degrees of Freedom	Type III Sum of Squares	Mean Square	F Value	Pr > F
Domain	4	11383876.38	2845969.10	1064.95	**<.0001**
Method	2	67570388.25	33785194.12	12642.3	**<.0001**
Method*Domain	8	11044226.92	1380528.37	516.59	**<.0001**

Table 4.18 shows the Duncan Multiple Range Test results for the independent variables with respect to Count by domain and method. Recall that factors identified by the same letter are not significantly different. From Table 4.18, it is clear that there was a significant difference between each of the domains and each method as expected.

Table 4.18: Duncan's Multiple Range Test for Count.

Domain

Duncan Grouping	Mean	# of Observations	Domain
A	1809.28	60	PROD
B	1630.67	60	LOG
C	1478.68	60	RELI
D	1338.32	60	INV
E	1275.03	60	PERT

Method

Duncan Grouping	Mean	# of Observations	Method
A	1993.16	100	GA-SO
B	1663.20	100	TS-SO
C	862.83	100	LNM-SO

Count provides an interesting comparison measure. However, Count by itself is not very useful since it is completely dependent on the particular simulation optimization method. The number of solutions that are examined in each iteration and how many duplicate points are examined in each iteration varies depending on the particular method used and how the method is setup. A more interesting and useful comparison is the savings that can be found based on the number of calls to the simulation. This value is referred to as the potential total solution percent savings and can be represented by the following equation:

$$\textit{Potential Total Solution Percent Savings} = \left(\frac{Count - ROCCM}{Count}\right) \cdot 100 \tag{4.2}$$

Thus, if the value of Count of is 2000 and the value of ROCCM is 200, there is a potential savings of 1800 calls to the simulation. This is a 90% savings. Thus, if the methods did not incorporate memory to ensure that each solution is only simulated once, it is clear that it would take a great deal longer to complete each simulation optimization run. The average percent savings for each domain by method are presented in Figure 4.11. It can be seen that LNM-SO had the greatest savings for all but the inventory domain, while TS-SO had the second greatest savings for all but the inventory domain. GA-SO had the lowest savings for all but the inventory domain, where it had the greatest savings. In general, all the domains for each method had large percent savings.

A similar measure can be derived by comparing ROCCM and ROCBS. This is the potential savings that could be obtained if the program was stopped once the ROCBS was obtained. Of course this is not feasible, since a solution cannot be declared the best solution until a sufficient number of evaluations have been made (the actual number of evaluations required is determined by the simulation optimization method). However, this is still an interesting measure. This potential percent savings is calculated by taking the difference of ROCCM and ROCBS and dividing it by ROCCM. This expression is referred to as the potential best solution percent savings. It can be expressed as:

$$\textit{Potential Best Solution Percent Savings} = \left(\frac{ROCCM - ROCBS}{ROCCM}\right) \cdot 100 \tag{4.3}$$

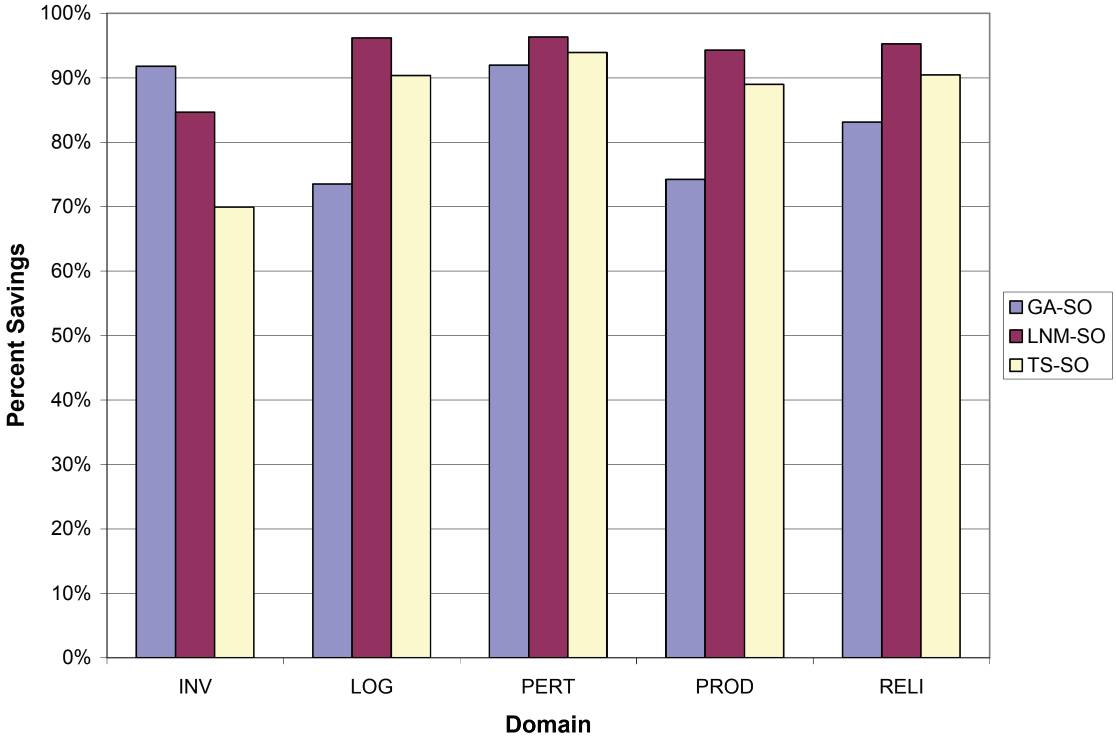


Figure 4.11: Mean Total Solution Percent Savings between Count and ROCCM by Domain and Method.

This data is presented in Figure 4.12. From the figure, it is observed that TS-SO had the largest potential percent savings for all domains except the inventory domain. LNM-SO had the highest potential savings for the inventory domain and the second highest for the logistics, PERT, and production domains. The potential best solution savings for each domain by the method ranged from approximately 20% to 60%.

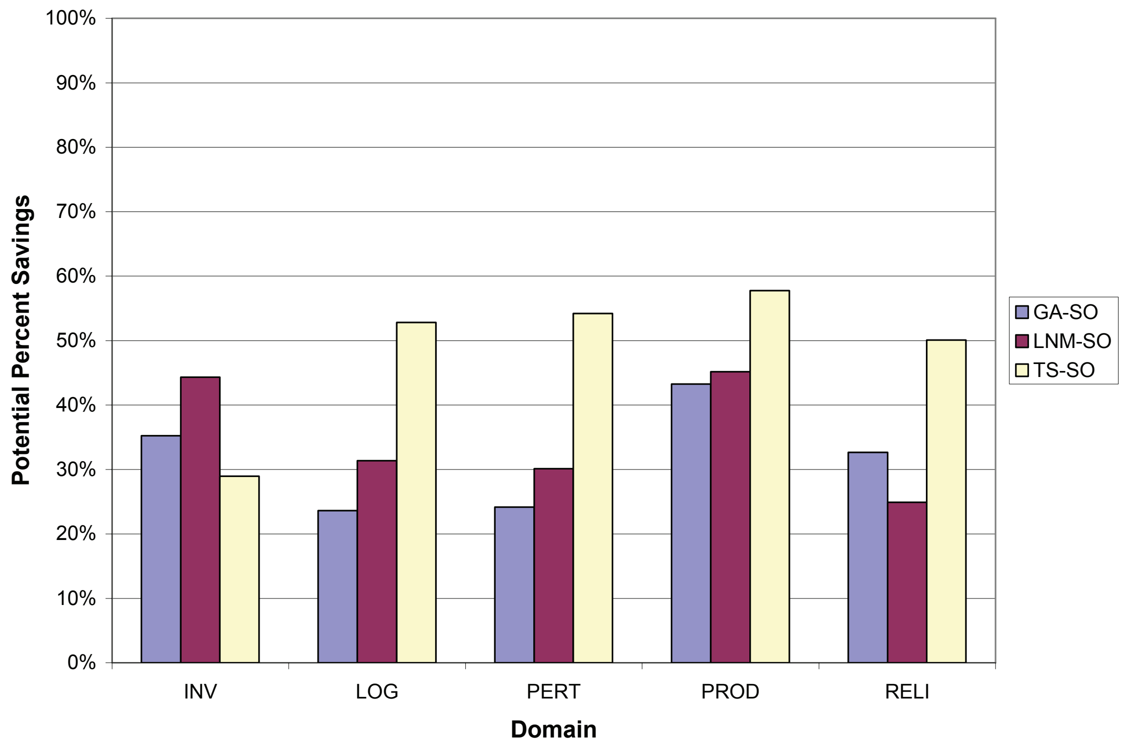


Figure 4.12: Mean Potential Best Solution Percent Savings between ROCCM and ROCBS by Domain and Method.

4.3.3 Quality of Solution

The quality of solution can be defined as the best solution found (BSF) and the corresponding objective function value (BSF_Z). However, it is also of interest to examine the BSF in relation to the initial solution examined (Init_Zdif).

4.3.3.1 Analysis of BSF

For this study, one performance measure examined quality of solution. This test measure was the best solution found (BSF). This was the solution that yielded the minimum objective function value (BSF_Z) found from each test run. This will provide a clear indication as to which method, if any, consistently found higher quality solutions.

However, since there is no reliable way to compare across domains, the best solution found was not used to compare the BSF from one domain to another. The data are presented through graphs of the mean BSF_Z and the minimum best solution found for the interaction effects of the method versus the domain (this provides the overall BSF_Z for each domain), an ANOVA table, and Duncan's Multiple Range tests.

The two-way interaction effects were plotted for BSF. Figures 4.13, 4.14, and 4.15 show the effect of the method and the domain on the mean BSF, the best solution found overall (the minimum BSF), and the standard deviation of BSF, respectively. From Figure 4.13, it can be seen that for the logistic, PERT, and production domains, all three methods have a similar mean BSF_Z. In general, GA-SO and LNM-SO have similar values for all the domains. The TS-SO method is clearly worse for the inventory and the reliability domains. From Figure 4.14, it can be seen that all three methods find a very similar best overall solution for the PERT, production, and reliability domains. For both the inventory and the logistic domains, GA-SO had the best overall solution followed by LNM-SO, and then TS-SO. Figure 4.15 shows that all three methods had a very low standard deviation for the logistic, PERT, and production domains. All the methods had substantial variability for the inventory domain, while the TS-SO method had extremely high variability for the reliability domain.

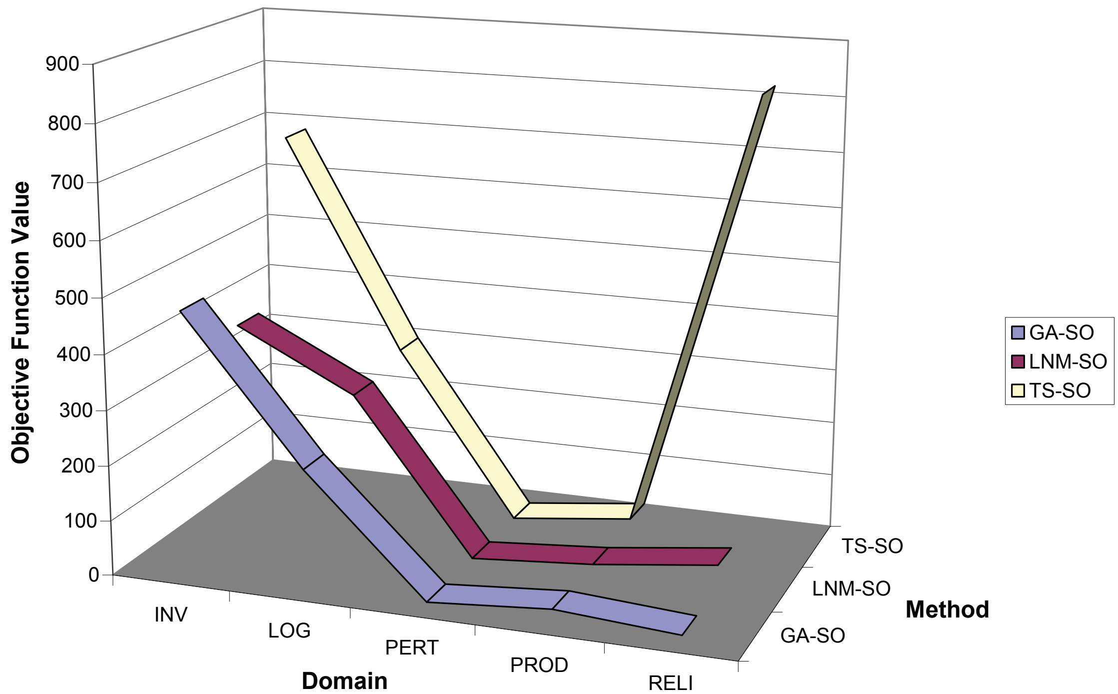


Figure 4.13: Mean Best Solution Found (Minimum) by Domain and Method.

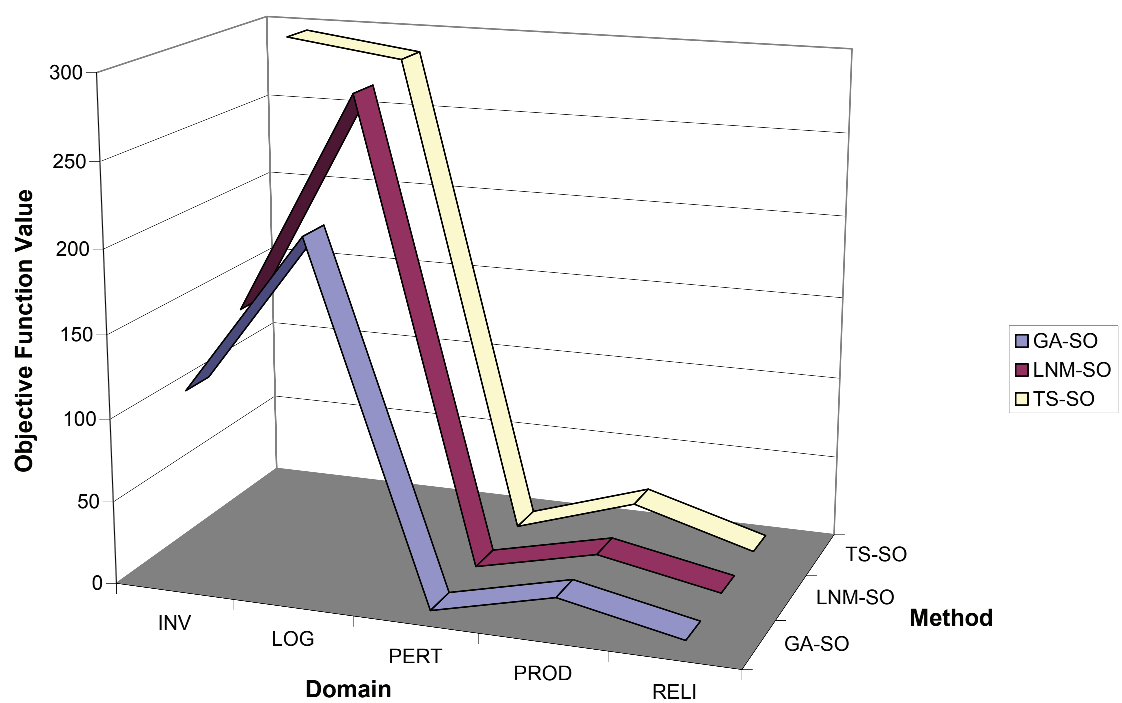


Figure 4.14: Overall BSF (Minimum) by Domain and Method.

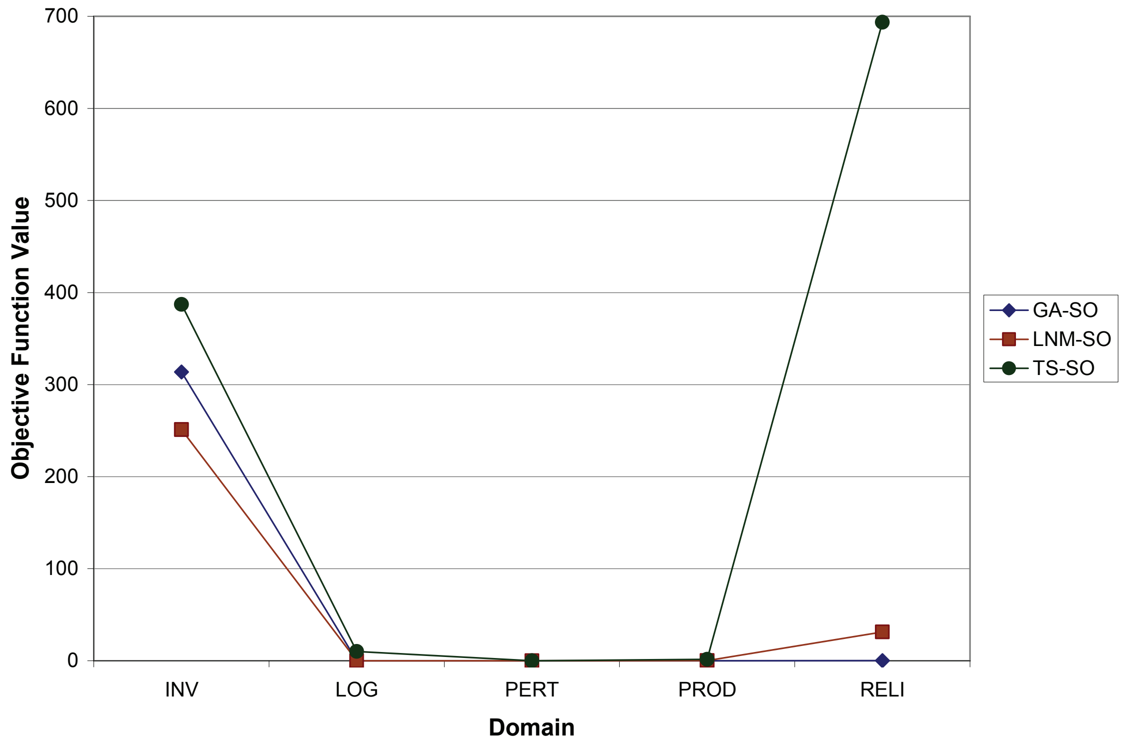


Figure 4.15: Standard Deviation of the BSF by Domain and Method.

The ANOVA results for BSF are presented in Table 4.19. The significant sources (as indicated by "Pr > F" having a value less than 0.05) are shown in boldface type. The domain, the simulation optimization method, and the interaction of the domain and the simulation optimization method were all statistically significant. This is to be expected, since there is not a reliable way to compare across domains and there was a large difference between each of the methods.

Table 4.19: ANOVA Table for BSF.

Source	Degrees of Freedom	Type III Sum of Squares	Mean Square	F Value	Pr > F
Domain	4	10918085.32	2729521.33	51.59	**<.0001**
Method	2	3367063.10	1683531.55	31.82	**<.0001**
Method*Domain	8	6230427.29	778803.41	14.72	**<.0001**

Table 4.20 shows the Duncan Multiple Range Test results for the independent variables with respect to BSF. Recall that factors identified by the same letter are not significantly different. From Table 4.20, it is clear LNM-SO and GA-SO were not statistically different. However, TS-SO was statistically different from the other two methods. These results were expected based on Figures 4.13 to 4.15.

Table 4.20: Duncan's Multiple Range Test for BSF.

Method			
Duncan Grouping	Mean	# of Observations	Method
A	367.85	100	TS-SO
B	145.67	100	LNM-SO
B	140.64	100	GA-SO

4.3.3.2 Additional Analysis

As previously discussed, the quality of solution was determined by BSF. However, it is also important to take into account the initial parameters or the starting solution (Initial X). In this way, it can be better determined how much impact the starting solution had on generating the best solution for each run. This will provide a level of robustness for each model and each method. The initial objective function value (Init_Z) was calculated and its distance from the BSF_Z for each domain regardless of method was determined. This value is referred to as Init_Zdif.

Table 4.21 presents up to the best five different solutions for each method and domain combination (note, for some cases, there were not five different solutions). Thus, if the same solution was obtained in several runs, it was only recorded one time. The frequency (Freq) parameter recorded the number of times this particular solution was found for that method and domain combination. Similarly to Init_Zdif, the BSF_Zdif was determined by taking the difference from the current Init_Z and the best objective function value found over all instances of that domain (note that if frequency was greater than one, only one initial solution was presented, even though each of the solutions had a different initial solution. The BSF over all runs for each domain are in boldfaced type in the table. A complete listing of starting and ending solutions for each run is presented in Appendix E. Additionally, the complete output from the first run that produced the BSF for each method-domain combination is presented in Appendix F.

The average, maximum, and standard deviation of Init_Zdif are presented in Figures 4.16 to 4.18, respectively. However, examining these figures, it is obvious that it is not possible to evaluate each domain due to the extreme difference in the objective function values. Thus, these figures will be used to compare the inventory and reliability domains. The logistic and production domains will be compared in Figures 4.19 to 4.21. The PERT domain is not plotted, since its objective function value is very close to zero for all runs. In fact, from Table 4.21, it can be seen that for the PERT domain the GA-SO and LNM-SO methods found the best overall solution found in all 40 runs (twenty runs each). The top five TS-SO method PERT solutions found were within 0.0008 of the overall best solution found.

Table 4.21: Summary of the Top 5 Different Solutions for Each Method-Domain Combination.

Method	Domain	Initial Solution	Init_Z	BSF	BSF_Z	BSF_Zdif	Init_Zdif	Freq
GA-SO	**INV**	[67.33, 30.00, 5.80]	1282.95	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	1168.54	**8**
GA-SO	INV	[53.33, 0.00, 7.60]	15364.65	[95.33, 28.00, 2.80]	640.29	525.88	15250.24	6
GA-SO	INV	[86.00, 16.00, 3.40]	2738.43	[86.00, 28.00, 4.00]	672.56	558.15	2624.02	2
GA-SO	INV	[76.67, 8.00, 2.20]	4256.10	[48.67, 28.00, 3.40]	672.56	558.15	4141.69	1
GA-SO	INV	[62.67, 22.00, 5.80]	2060.34	[100.00, 26.00, 4.00]	782.91	668.50	1945.93	2
GA-SO	**LOG**	[3, 2.87, 8.20, 9.33, 9.67]	213.22	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	1.97	**17**
GA-SO	LOG	[4, 5.00, 4.00, 8.33, 7.33]	215.62	[13, 5.00, 10.00, 10.00, 7.00]	211.25	0.00	4.37	2
GA-SO	LOG	[6, 1.27, 7.60, 7.67, 9.33]	214.37	[15, 4.73, 10.00, 10.00, 7.00]	211.25	0.00	3.12	1
GA-SO	**PERT**	[7.80, 7.53, 6.60]	0.1559	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	0.0235	**20**
GA-SO	PROD	[8.33, 14.00, 7.00, 4.33, 3.60, 6.00]	51.25	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	34.03	20
GA-SO	RELI	[15.00, 4.00, 8.80, 7.20]	36047.01	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	36044.47	13
GA-SO	RELI	[7.20, 2.80, 6.40, 6.33]	13941.52	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	13938.98	7
TS-SO	INV	[89.74, 28.87, 5.31]	1969.29	[96.74, 29.87, 5.31]	296.62	182.21	1854.88	1
TS-SO	INV	[93.47, 25.16, 8.04]	1686.18	[90.47, 29.16, 5.04]	303.77	189.35	1571.77	1
TS-SO	INV	[91.98, 18.92, 3.58]	3523.85	[95.98, 25.92, 2.58]	347.08	232.67	3409.44	1
TS-SO	INV	[95.78, 19.46, 3.20]	2375.05	[94.78, 25.46, 4.20]	428.10	313.69	2260.64	1
TS-SO	INV	[97.71, 2.31, 6.86]	4132.60	[96.71, 26.31, 3.86]	436.71	322.29	4018.19	1
TS-SO	LOG	[9, 4.54, 4.92, 6.08, 11.84]	316.16	[15, 4.54, 4.92, 9.08, 11.84]	287.11	75.86	104.91	1
TS-SO	LOG	[7, 2.69, 3.95, 5.47, 9.35]	379.21	[9, 4.69, 4.95, 9.47, 11.35]	288.55	77.31	167.96	1
TS-SO	LOG	[12, 4.85, 4.67, 6.66, 11.97]	314.77	[14, 4.85, 4.67, 9.66, 11.97]	289.84	78.59	103.53	1
TS-SO	LOG	[6, 2.08, 4.84, 7.54, 10.40]	312.71	[14, 4.08, 4.84, 9.54, 11.40]	290.64	79.39	101.47	1
TS-SO	LOG	[6, 3.10, 4.70, 7.96, 7.27]	345.31	[14, 4.10, 4.70, 9.96, 11.27]	292.23	80.98	134.07	1
TS-SO	PERT	[10.07, 4.01, 5.04]	0.1480	[5.07, 1.01, 1.04]	0.1325	0.0001	0.0156	1
TS-SO	PERT	[5.11, 3.23, 7.11]	0.1440	[5.11, 1.23, 1.11]	0.1328	0.0005	0.0116	1
TS-SO	PERT	[10.06, 6.09, 3.58]	0.1506	[5.06, 1.09, 1.58]	0.1330	0.0007	0.0182	1
TS-SO	PERT	[6.27, 3.15, 6.35]	0.1432	[5.27, 1.15, 1.35]	0.1331	0.0007	0.0109	1
TS-SO	PERT	[5.07, 2.53, 5.08]	0.1391	[5.07, 1.53, 1.08]	0.1331	0.0008	0.0067	1
TS-SO	PROD	[5.42, 15.14, 7.24, 6.42, 3.31, 5.91]	55.53	[5.42, 10.14, 5.24, 3.42, 1.31, 2.91]	23.55	6.34	38.32	1
TS-SO	PROD	[5.54, 10.16, 5.77, 5.12, 3.71, 4.05]	40.19	[5.54, 10.16, 5.77, 3.12, 1.71, 3.05]	24.04	6.83	22.98	1
TS-SO	PROD	[8.64, 17.41, 9.21, 6.43, 1.52, 2.35]	56.13	[5.64, 10.41, 5.21, 3.43, 1.52, 2.35]	24.37	7.15	38.91	1
TS-SO	PROD	[7.20, 19.55, 8.95, 5.25, 3.20, 2.88]	58.74	[5.20, 10.55, 5.95, 3.25, 1.20, 2.88]	24.50	7.28	41.53	1
TS-SO	PROD	[8.37, 15.96, 11.11, 5.35, 1.09, 3.42]	52.72	[5.37, 10.96, 5.11, 3.35, 1.09, 2.42]	24.50	7.28	35.51	1
TS-SO	**RELI**	[8.44, 3.08, 2.61, 10.28]	23843.71	**[2.44, 1.08, 9.61, 6.28]**	**2.54**	**0.00**	23841.17	**1**
TS-SO	RELI	[6.01, 3.55, 2.14, 2.65]	32232.39	[2.01, 1.55, 7.14, 10.65]	55.97	53.43	32229.85	1
TS-SO	RELI	[5.07, 4.21, 4.35, 3.87]	17806.98	[2.07, 1.21, 9.35, 9.87]	85.82	83.28	17804.44	1
TS-SO	RELI	[3.10, 2.27, 9.17, 6.99]	3507.21	[2.10, 1.27, 7.17, 10.99]	112.54	110.00	3504.67	1
TS-SO	RELI	[10.21, 4.10, 2.29, 11.56]	35079.54	[2.21, 1.10, 8.29, 11.56]	126.64	124.10	35077.00	1
LNM-SO	INV	[42.75, 3.57, 4.12]	7021.67	[99.66, 29.88, 4.53]	143.98	29.56	6907.25	1
LNM-SO	INV	[70.49, 23.06, 1.54]	2357.24	[100.00, 30.00, 4.25]	148.83	34.42	2242.83	2
LNM-SO	INV	[72.97, 4.68, 8.01]	4583.64	[100.00, 30.00, 5.90]	172.41	57.99	4469.23	1
LNM-SO	INV	[87.78, 11.54, 8.47]	3227.48	[95.27, 29.40, 3.70]	185.15	70.73	3113.07	1
LNM-SO	INV	[49.18, 18.98, 8.60]	4956.10	[97.51, 28.93, 4.27]	228.35	113.93	4841.69	1
LNM-SO	LOG	[6, 2.81, 3.89, 9.79, 9.24]	339.21	[13, 5.00, 5.00, 10.00, 12.00]	278.90	67.65	127.96	2
LNM-SO	LOG	[6, 1.22, 2.78, 6.40, 7.19]	462.56	[13, 4.68, 5.00, 10.00, 12.00]	278.90	67.65	251.32	1
LNM-SO	LOG	[2, 2.23, 3.39, 7.96, 8.30]	386.93	[13, 4.58, 5.00, 10.00, 12.00]	278.90	67.65	175.68	1
LNM-SO	LOG	[13, 3.43, 2.81, 6.65, 8.84]	436.46	[13, 4.56, 5.00, 10.00, 12.00]	278.90	67.65	225.22	1
LNM-SO	LOG	[7, 1.56, 3.24, 6.70, 11.41]	383.21	[13, 4.52, 5.00, 10.00, 12.00]	278.90	67.65	171.96	2
LNM-SO	**PERT**	[9.10, 6.75, 7.35]	0.1561	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	0.0237	**20**
LNM-SO	**PROD**	[8.11, 18.48, 13.67, 3.88, 3.24, 2.28]	57.92	**[5.00, 10.00, 5.00, 3.00, 1.00, 2.18]**	**17.21**	**0.00**	40.71	**2**
LNM-SO	PROD	[5.52, 14.88, 7.47, 3.69, 1.31, 5.28]	44.11	[5.01, 10.00, 5.00, 3.00, 1.00, 2.00]	17.39	0.17	26.90	2
LNM-SO	PROD	[6.29, 12.37, 10.51, 4.90, 3.35, 2.17]	46.63	[5.00, 10.00, 5.00, 3.00, 1.00, 2.06]	17.41	0.19	29.42	1
LNM-SO	PROD	[6.03, 12.10, 10.68, 4.59, 3.19, 4.57]	47.04	[5.00, 10.00, 5.04, 3.02, 1.00, 2.00]	17.47	0.26	29.82	1
LNM-SO	PROD	[6.60, 10.93, 11.27, 6.12, 2.58, 4.20]	47.28	[5.02, 10.00, 5.00, 3.00, 1.03, 2.03]	17.50	0.29	30.07	1
LNM-SO	RELI	[12.54, 4.84, 9.92, 4.88]	33499.59	[2.01, 1.13, 9.57, 10.70]	3.15	0.61	33497.05	1
LNM-SO	RELI	[4.34, 1.05, 3.77, 10.74]	4250.57	[2.00, 1.00, 10.00, 11.74]	15.75	13.21	4248.03	1
LNM-SO	RELI	[5.09, 2.95, 2.86, 11.88]	13660.24	[2.00, 1.00, 10.00, 11.72]	15.75	13.21	13657.70	1
LNM-SO	RELI	[3.73, 1.36, 5.21, 10.55]	2876.39	[2.00, 1.00, 10.00, 11.66]	19.31	16.77	2873.85	1
LNM-SO	RELI	[5.39, 4.17, 9.93, 5.91]	13405.15	[2.00, 1.00, 10.00, 11.75]	23.97	21.43	13402.61	1

The differences in the objective function values are a product of the particular problems examined at each domain. This involves the number of goals in the problem and the target values for each goal. The inventory and reliability domains each involve three goals, while the other domains involve only two goals. Recall that a greater penalty is assessed for not satisfying higher priority goals. Thus, the more goals that a problem had, the greater the potential penalty and consequently the greater the potential objective function value was. Additionally, the PERT network satisfies goal 1 for all runs and nearly satisfies goal 2 for all runs. Thus, the result is a very small objective function value. This suggests that the specific problem for the PERT domain is very robust.

Examining Figures 4.16 to 4.18 for the inventory domain, it can be seen that all three methods had a very similar mean Init_Zdif, and the LNM-SO method had the least standard deviation. For the Reliability domain, the GA-SO has on average the worst starting solutions and the least variability of all the methods. Recall from the previous section, that GA-SO also had the average best solution found with the minimum standard deviation. This suggests that the GA-SO method is very robust and the most effective for the reliability domain.

From Figures 4.19 to 4.21, it is observed that the production domain had very similar results for the mean Init_Zdif, the maximum Init_Zdif, and the standard deviation of the Init_Zdif. From the previous section, the same was observed to be true for BSF. This suggests that the specific problem for the production domain is very robust.

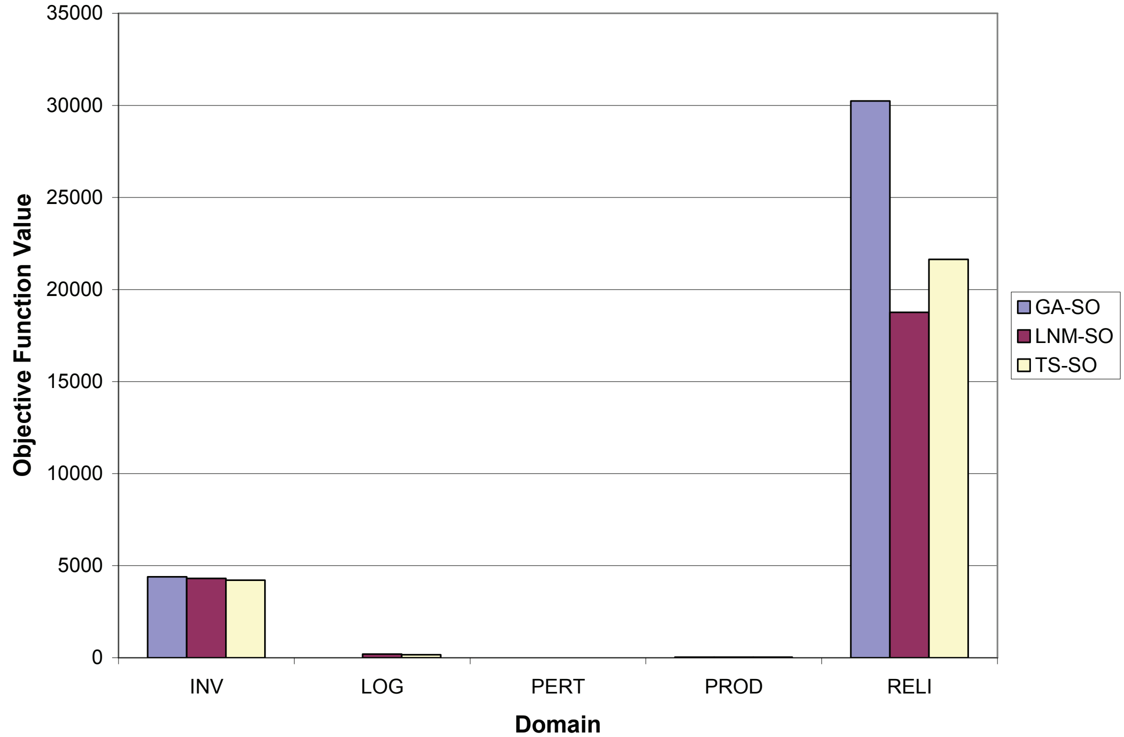


Figure 4.16: Mean Init_Zdif by Domain and Method.

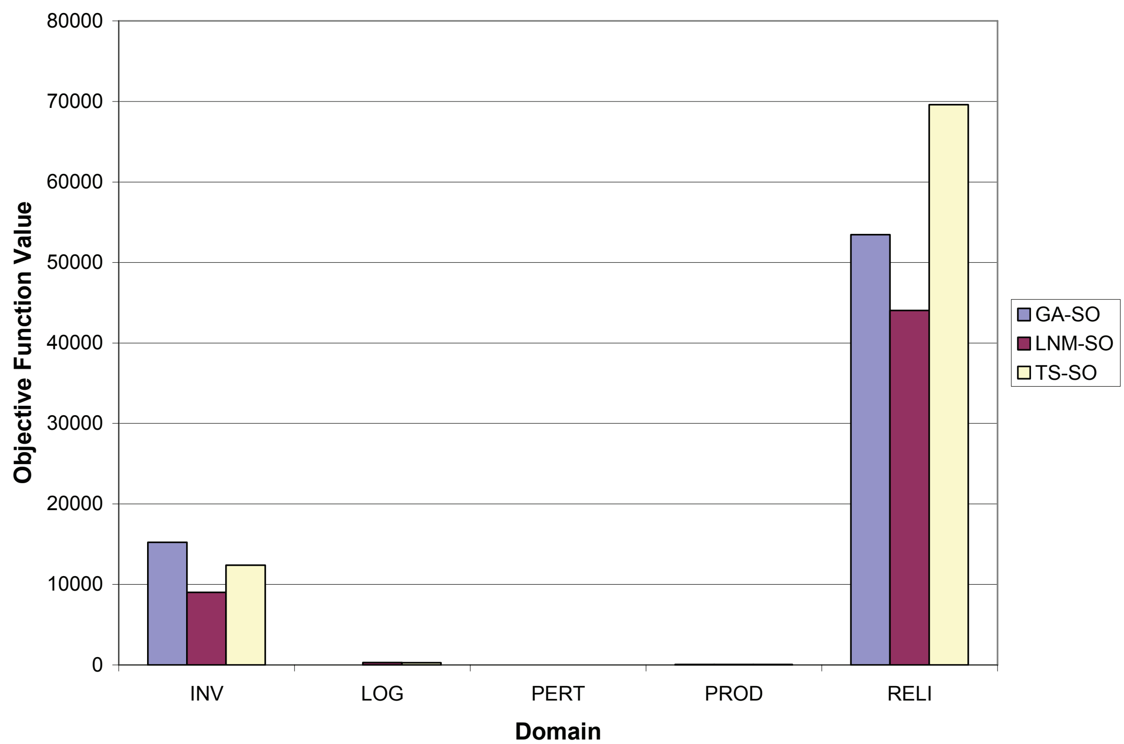


Figure 4.17: Maximum Init_Zdif by Domain and Method.

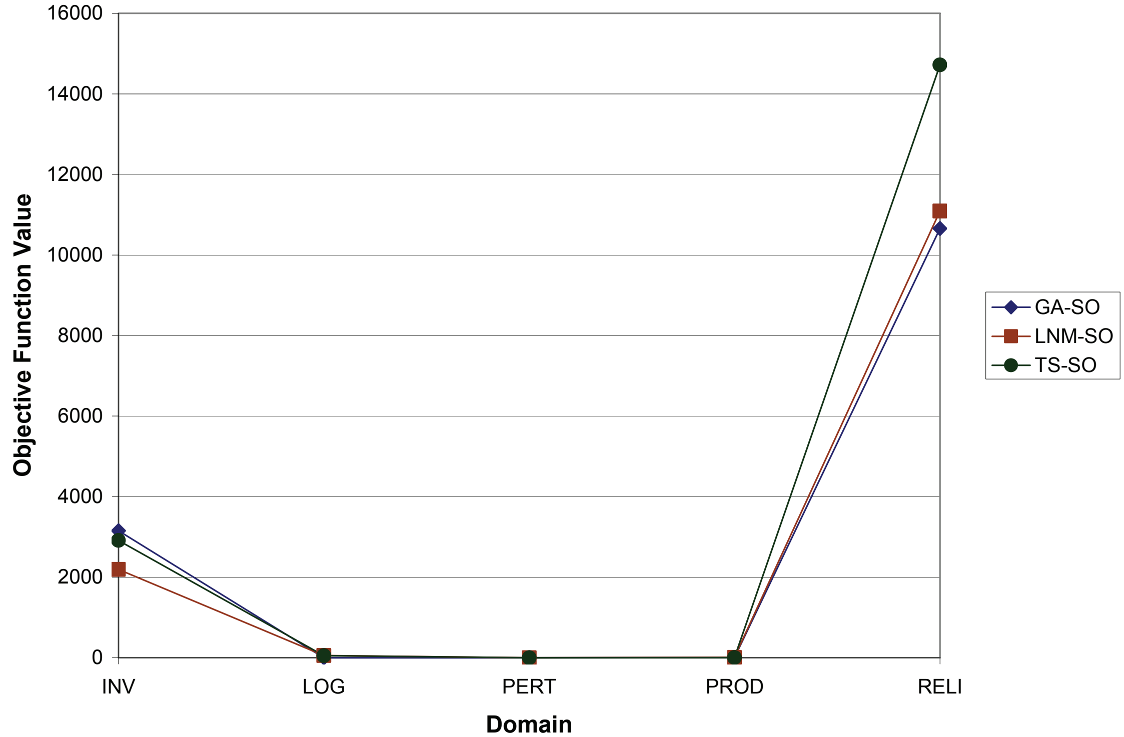


Figure 4.18: Standard Deviation of Init_Zdif by Domain and Method.

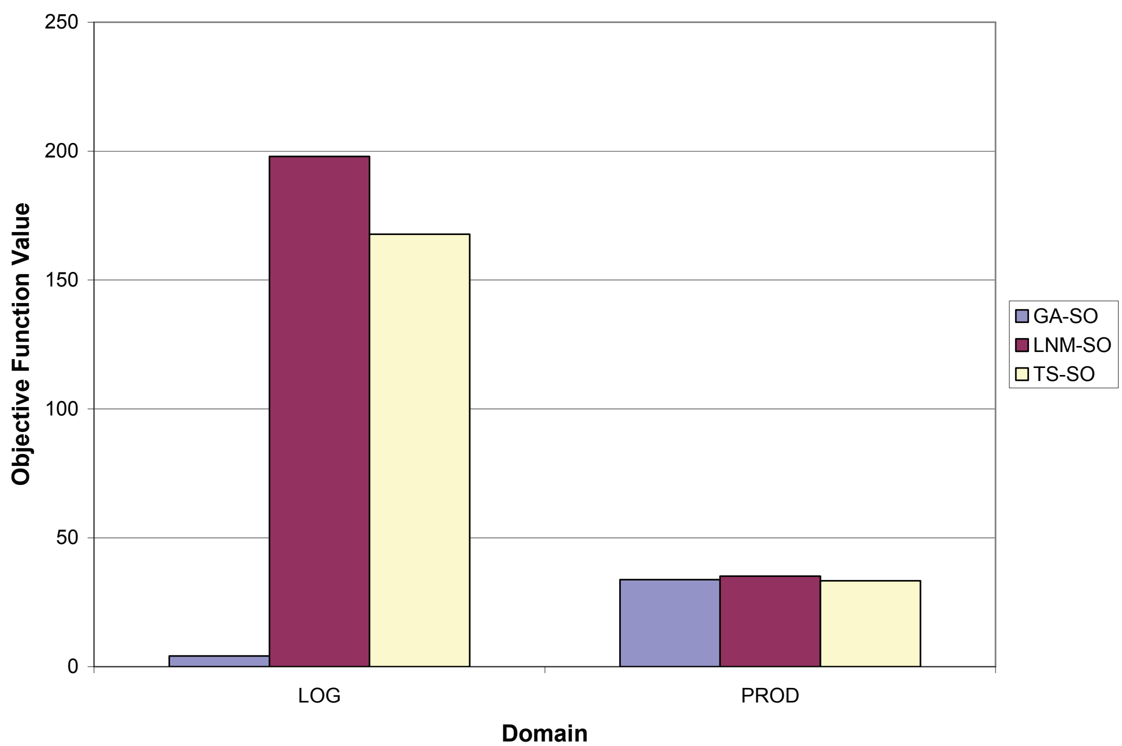


Figure 4.19: Mean Init_Zdif by Logistic and Production Domains and Method.

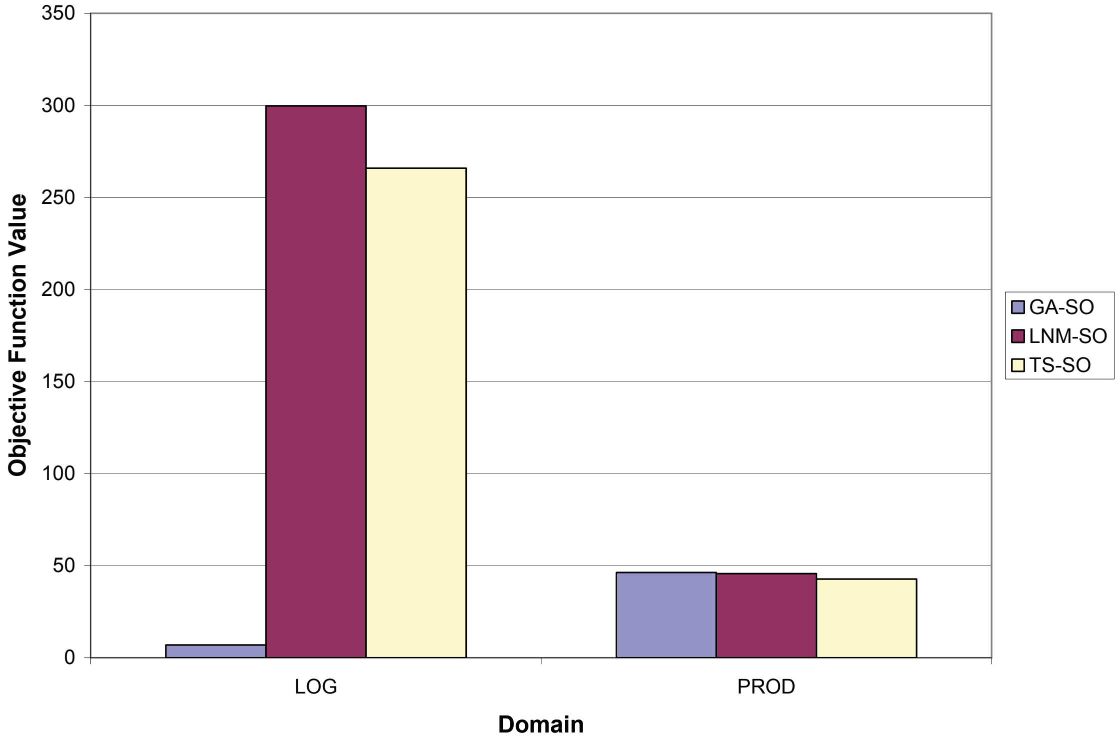


Figure 4.20: Maximum Init_Zdif by Logistic and Production Domains and Method.

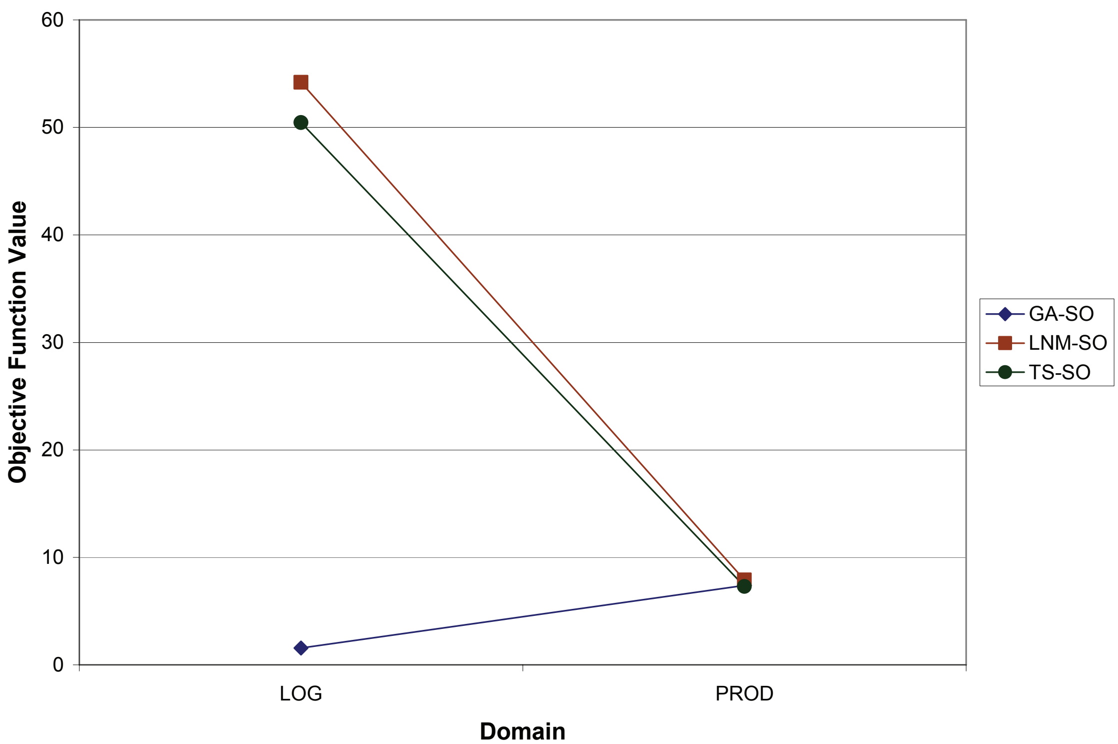


Figure 4.21: Standard Deviation of Init_Zdif by the Logistic and Production Domains and Method.

For the logistics domain, it is observed that the GA-SO had much lower Init_Zdif than the other two methods, and it had very little variability. Examining the results from the previous section, all the methods had similar mean BSF objective function values for the logistics domain. The best overall solution was found by GA-SO and it had the least variability. However, since the Init_Zdif was so small for GA-SO, it is hard to evaluate how well the method performed. The other two methods appear to perform well for this domain.

The ANOVA results for Init_Zdif are presented in Table 4.22. The significant sources (as indicated by "Pr > F" having a value less than 0.05) are shown in boldface type. The domain, the simulation optimization method, and the interaction of the domain and the simulation optimization method were all statistically significant. This is to be expected, since there is not a reliable way to compare across domains and there was a large difference between each of the methods.

Table 4.22: ANOVA Table for Init_Zdif.

Source	Degrees of Freedom	Type III Sum of Squares	Mean Square	F Value	Pr > F
Domain	4	24960996195	6240249049	196.43	**<.0001**
Method	2	281312487	140656244	4.43	**0.0128**
Method*Domain	8	1144537259	143067157	4.50	**<.0001**

Table 4.23 shows the Duncan Multiple Range Test results for the independent variables with respect to Init_Zdif. Recall that factors identified by the same letter are not significantly different. From Table 4.23, it is clear that LNM-SO and TS-SO were not statistically different. However, GA-SO was statistically different from the other two methods. These results were expected based on Figures 4.16 to 4.21. One possible

explanation of this is how the initial solution was calculated. For LNM-SO and TS-SO, the same random function formula was used in MATLAB, while a different random function formula was used for GA-SO in C++.

Table 4.23: Duncan's Multiple Range Test for Init_Zdif.

Method			
Duncan Grouping	Mean	# of Observations	Method
A	6934.3	100	GA-SO
B	5209.9	100	TS-SO
B	4661.6	100	LNM-SO

Figures 4.22 through 4.26 show the convergence for the inventory, logistic, PERT, production, and reliability domains, respectively, based on the GA-SO, LNM-SO, and TS-SO methods. Essentially, the figures are plotting the best objective function found over the first 100 iterations or generations (depending on the method). For the inventory domain, all three methods initially fall quickly and then level off quickly. GA-SO finds its best solution after 43 iterations and LNM-SO finds its best after 100 iterations, and TS-SO finds its best solution after 14 iterations. The LNM-SO method finds the best overall solution, but it is only slightly better than the TS-SO best solution found, and the TS-SO finds its best solution 86 iterations sooner.

For the logistics domain, all three methods appear to find the BSF or a value very near the BSF within the first 2 iterations. The LNM-SO and TS-SO methods find similar solutions, while the GA-SO finds a much better solution. For the PERT domain, all three methods find a very similar objective function value. The LNM-SO method finds it in the first iteration, the GA-SO finds it in the fourth iteration, and the TS-SO finds it in the seventh iteration.

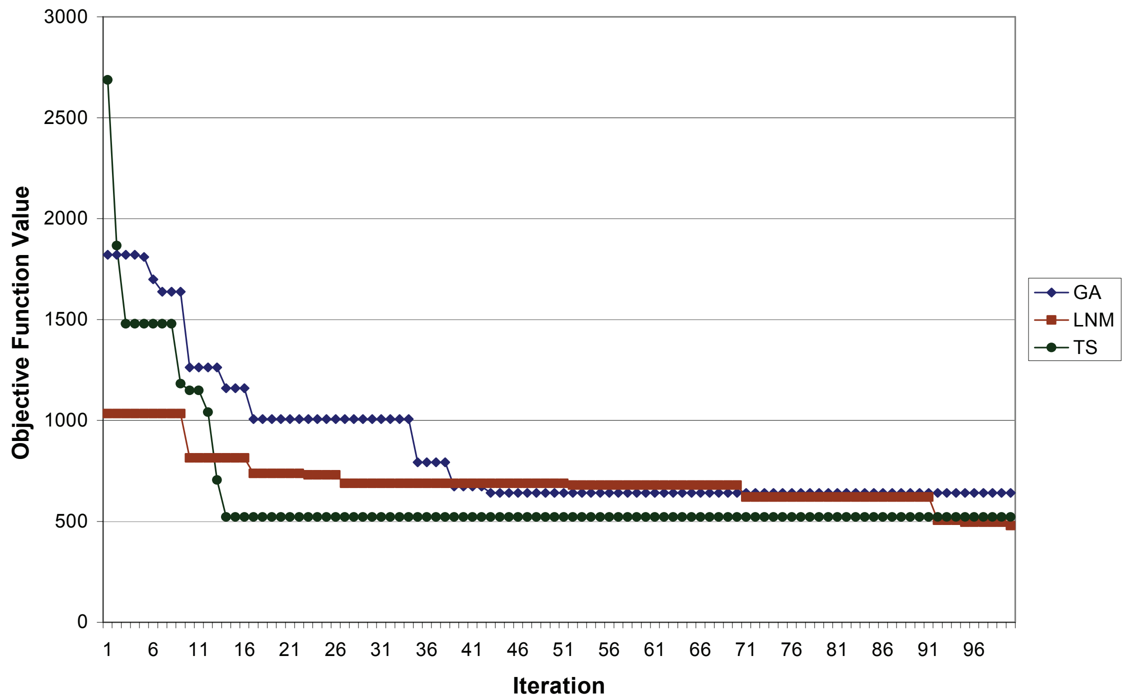


Figure 4.22: Inventory Convergence Plot by Method.

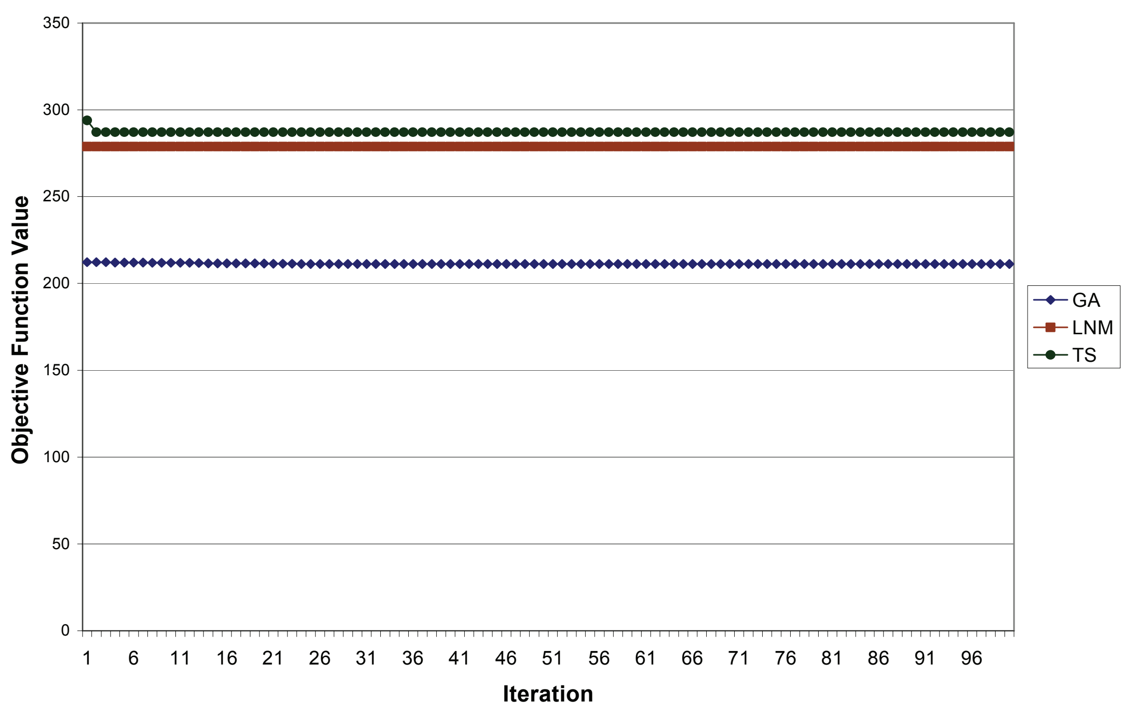


Figure 4.23: Logistics Convergence Plot by Method.

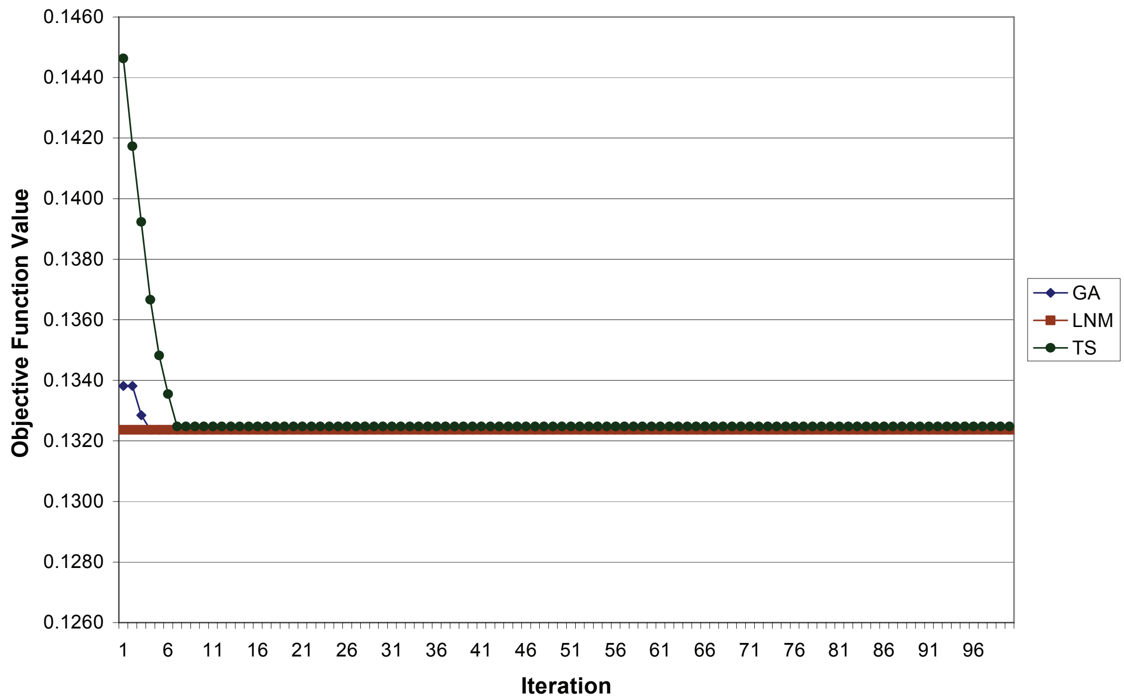


Figure 4.24: PERT Convergence Plot by Method.

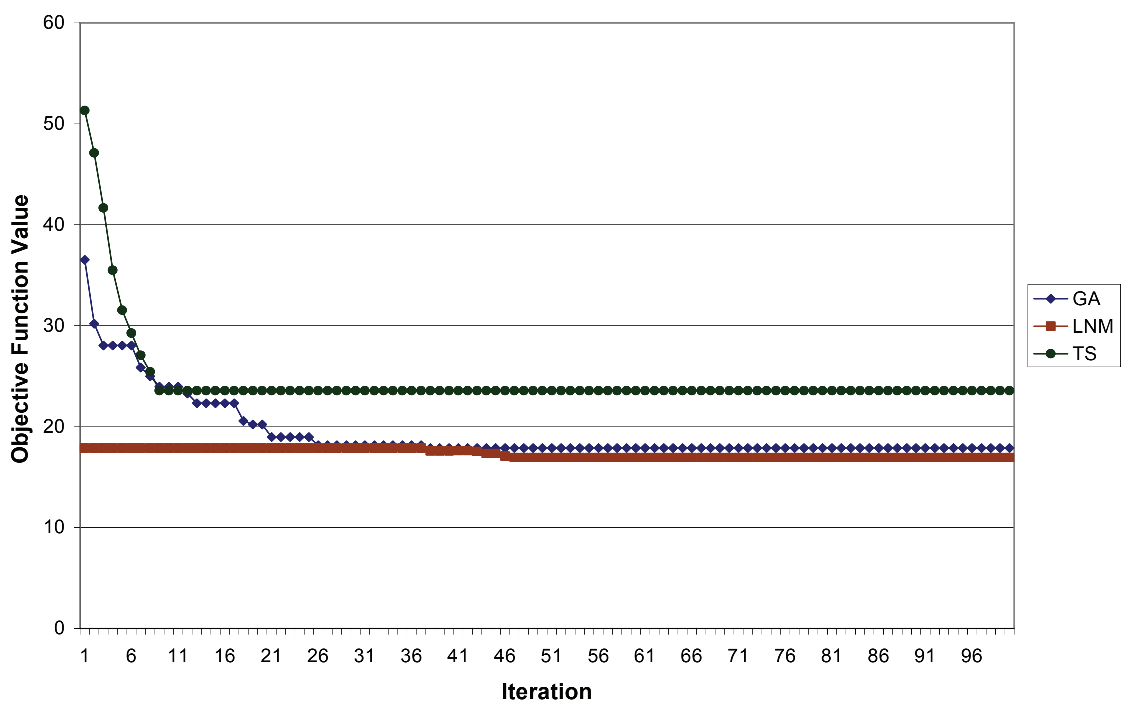


Figure 4.25: Production Convergence Plot by Method.

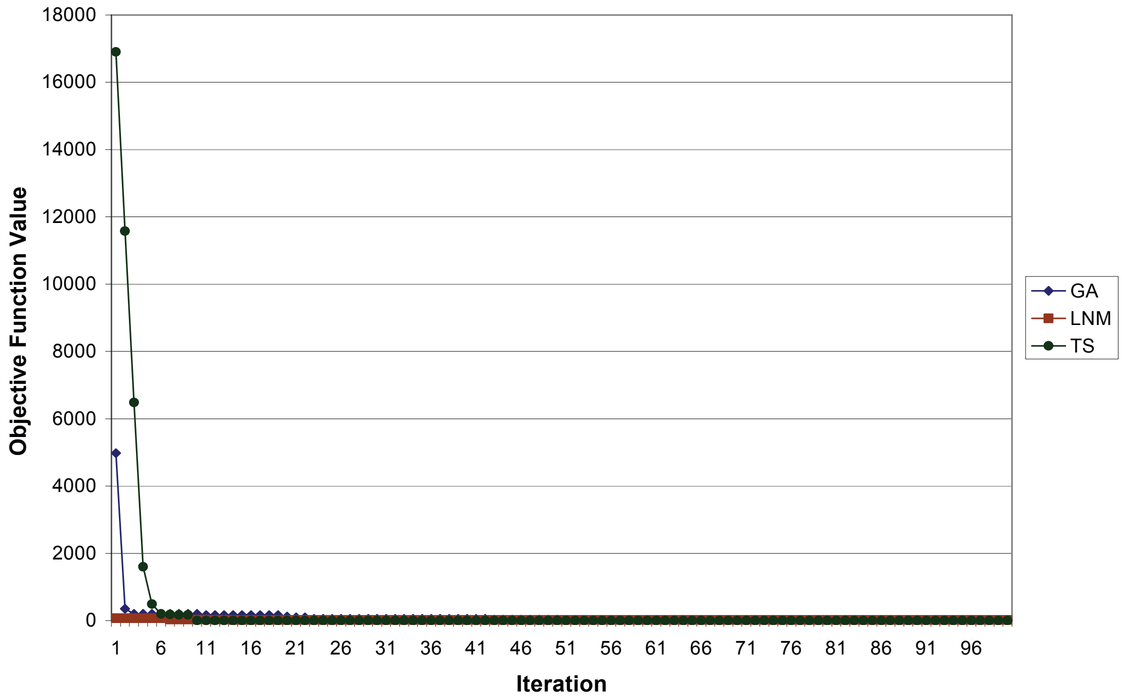


Figure 4.26: Reliability Convergence Plot by Method.

For the production domain, both the GA-SO and the LNM-SO simulation optimization methods find similar best objective function values. The LNM-SO method essentially finds it right away, while the GA-SO finds it after twenty-five generations. The TS-SO method finds its best objective function value after nine iterations; however, its value is not as good as the other two method's values.

Finally for the reliability domain, the LNM-SO method converges to an objective function value of around 16 after fourteen iterations. The GA-SO method drops from a value around 5000 to a value just under 3 after 51 generations. The TS-SO method falls dramatically from a solution of almost 17,000 after the first iteration a value of approximately 2.5 after just 10 iterations.

It is important to note that these plots were based only on one run for each domain. The particular run chosen was the run that generated the BSF across all runs for a given domain and method (i.e., the BSF out of 20 runs). Thus, it is possible and certainly likely for some of the cases that these do not present a representative sample. They do however provide a view of how each method progresses. Since for most of these plots, the methods reach the best solution found very quickly, it maybe possible to adjust the termination criteria to reduce the number of runs required before convergence can be checked. However, it is important to note that since duplicate solutions are not re-simulated, changing the termination criteria likely will not affect the number of calls required to complete the method (ROCCM) by very much, if at all. Reducing the minimum number of iterations required to complete the method must be done with caution, since if it is dropped too much, it could have a large impact on the quality of solution.

4.3.4 Best Overall Performance

The overall performance is a combination of both the computational speed and the best solution found. To compute this it is important to include the mean and variance of the best solution found and the mean and variance of ROCBS. The mean and standard deviations for the combined replications for each domain by the methods are provided in Table 4.24. These values were used to calculate the 95% confidence intervals (CI) on the ROCCM, ROCBS, and BSF_Z. The CI values are provided in Table 4.25.

Table 4.24: Mean and Standard Deviation Values.

Method	Domain	ROCCM		ROCBS		BSF_Z	
		Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
LNM-SO	INV	134.90	71.31	75.10	58.54	386.43	251.08
GA-SO	INV	159.85	26.38	103.50	44.57	470.81	313.83
TS-SO	INV	357.10	152.19	253.70	132.66	687.19	387.11
LNM-SO	LOG	34.10	8.27	23.40	10.05	278.90	0.00
TS-SO	LOG	190.60	51.69	89.95	40.24	301.93	10.24
GA-SO	LOG	534.95	60.31	408.60	73.33	211.25	0.00
LNM-SO	PERT	25.25	31.84	17.65	32.13	0.13	0.00
TS-SO	PERT	72.15	16.88	33.05	14.48	0.13	0.00
GA-SO	PERT	155.95	23.64	118.30	23.63	0.13	0.00
LNM-SO	PROD	55.90	12.39	30.65	20.89	17.66	0.24
TS-SO	PROD	262.00	86.78	110.70	39.24	26.08	1.55
GA-SO	PROD	533.70	54.99	302.95	94.84	17.86	0.00
LNM-SO	RELI	40.95	47.93	30.75	34.03	45.22	31.32
GA-SO	RELI	335.05	41.33	225.65	40.36	3.16	0.29
TS-SO	RELI	158.60	38.20	78.79	24.02	748.38	694.23

Table 4.25: Confidence Interval Values.

Method	Domain	ROCCM CI	ROCBS CI	BSF_Z CI
LNM-SO	INV	31.25	25.66	110.04
GA-SO	INV	11.56	19.53	137.54
TS-SO	INV	66.70	58.14	169.66
LNM-SO	LOG	3.62	4.40	0.00
TS-SO	LOG	22.65	17.64	4.49
GA-SO	LOG	26.43	32.14	0.00
LNM-SO	PERT	13.96	14.08	0.00
TS-SO	PERT	7.40	6.35	0.00
GA-SO	PERT	10.36	10.35	0.00
LNM-SO	PROD	5.43	9.15	0.11
TS-SO	PROD	38.03	17.20	0.68
GA-SO	PROD	24.10	41.57	0.00
LNM-SO	RELI	21.01	14.92	13.73
GA-SO	RELI	18.12	17.69	0.13
TS-SO	RELI	16.74	10.53	304.25

It is desirable to calculate the speed and quality weights (w_S and w_Q) that will be used to determine the overall performance calculated as the Time-Quality Estimator (TQE). In order to do this, the TQE values were calculated for w_S and w_Q ranging from one to ten, using the sum of all the means and standard deviations as the combined data set used to determine the appropriate values for the TQE equation. The results for w_S were plotted by w_Q. These plots are shown in Figure 4.27.

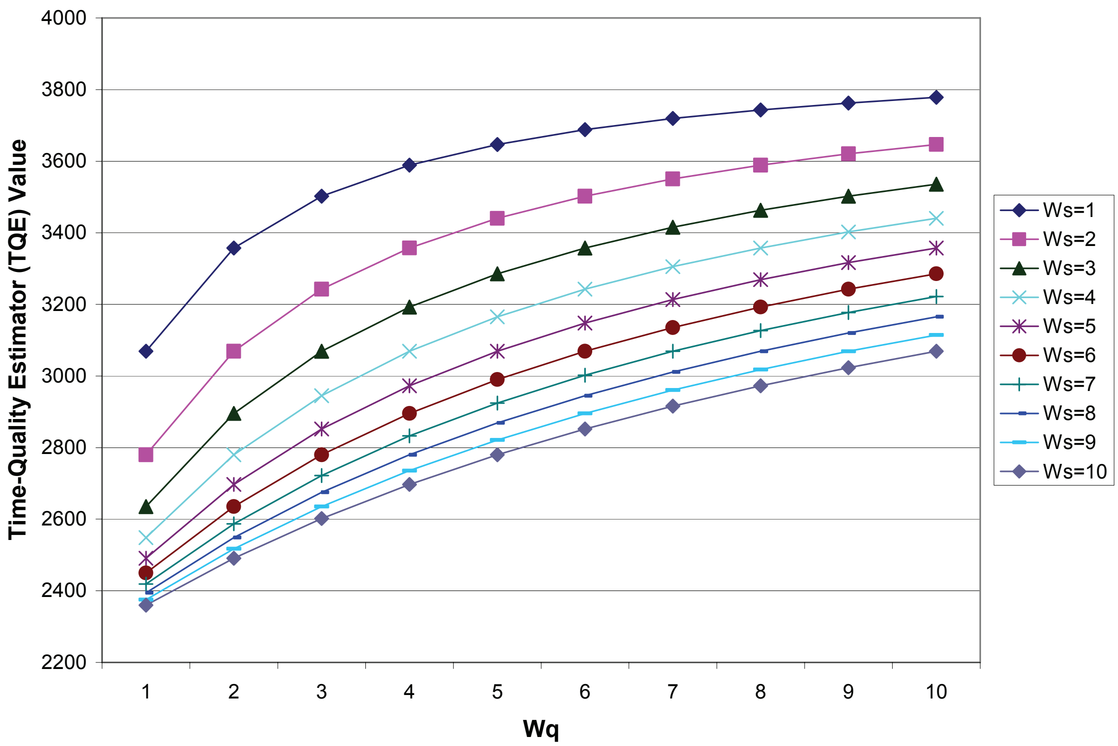


Figure 4.27: Evaluation of the Sum of the Overall Performance for All Cases.

Examining the above the figure, it appears that each of the plots begin to level off around a w_Q value of 5. Thus, a value of 5 was chosen as the time-to-quality ratio. Thus, the weights used for the Time-Quality Neutral Case, the Time Sensitive Case, and the Quality Sensitive Case were:

For the Time-Quality Neutral Case: $w_S = w_Q$ (4.4)

For the Time-Sensitive Case: $w_S = 5 \quad w_Q = 1$ (4.5)

For the Quality-Sensitive Case: $w_S = 1 \quad w_Q = 5.$ (4.6)

From the values calculated and shown in Tables 4.24 and 4.25, the upper confidence limit (UCL) can be determined. The UCL of ROCBS and BSF_Z, along with the speed and quality weights were used to calculate the Time-Quality Estimator for the neutral, time sensitive, and quality sensitive cases. All of these values are provided in Table 4.26. The best TQE score for each sensitivity level and each domain are in boldfaced type.

Table 4.26: Time-Quality Estimator Scores.

Method	Domain	UCL			Time-Quality Estimator (TQE)		
		ROCCM	ROCBS	BSF_Z	Neutral	Time	Quality
LNM-SO	INV	166.15	100.76	496.47	**298.61**	**166.71**	**430.52**
GA-SO	INV	171.41	123.03	608.35	365.69	203.92	527.47
TS-SO	INV	423.80	311.84	856.84	584.34	402.67	766.01
LNM-SO	LOG	37.72	27.80	278.90	**153.35**	**69.65**	**237.05**
TS-SO	LOG	213.25	107.59	306.42	207.00	140.73	273.28
GA-SO	LOG	561.38	440.74	211.25	325.99	402.49	249.50
LNM-SO	PERT	39.21	31.73	0.13	**15.93**	**26.46**	**5.40**
TS-SO	PERT	79.55	39.40	0.13	19.76	32.85	6.68
GA-SO	PERT	166.31	128.65	0.13	64.39	107.23	21.55
LNM-SO	PROD	61.33	39.80	17.76	**28.78**	**36.13**	**21.44**
TS-SO	PROD	300.03	127.90	26.76	77.33	111.04	43.62
GA-SO	PROD	557.80	344.52	17.86	181.19	290.07	72.31
LNM-SO	RELI	61.96	45.67	58.95	**52.31**	**47.88**	56.74
GA-SO	RELI	353.17	243.34	3.28	123.31	203.33	**43.29**
TS-SO	RELI	175.34	89.31	1052.63	570.97	249.86	892.08
Sum Total		**3368.42**	**2202.07**	**3935.88**	**3068.98**	**2491.04**	**3646.91**

The Time-Quality Estimator for each sensitivity level was broken down by the domain. These results are plotted in Figures 4.28 – 4.32. From these figures it can be observed that except for one case, the LNM-SO method provided the best TQE score for all sensitivity levels for all five domains. The only exception was for the reliability domain for quality-sensitive decision-makers. In this case GA-SO had the best TQE score, but LNM-SO was a very close second. The TS-SO method had the second best score for all sensitivity levels for the logistics and the PERT domains. TS-SO also had the second best score for the neutral and time-sensitive cases for the production domain. For the inventory and the reliability domains, the GA-SO had the second best TQE scores for all cases except the aforementioned case where it had the best score for the quality case of the reliability domain. GA-SO also had the second best score for the quality case of the production domain.

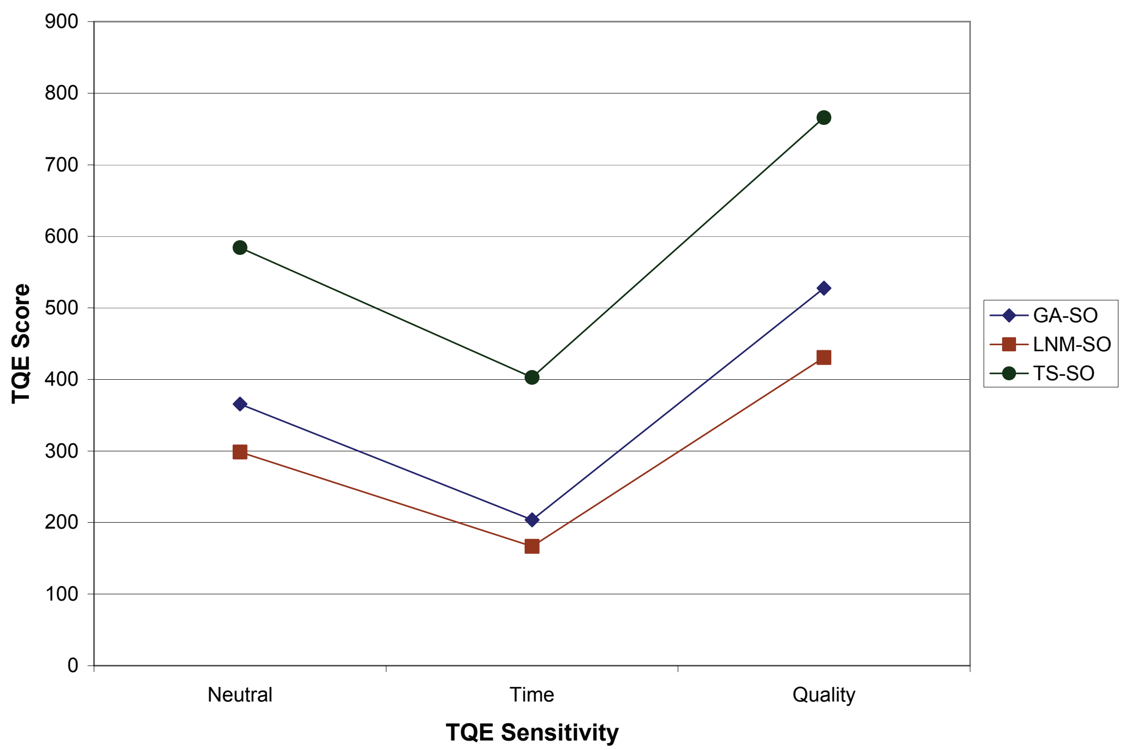


Figure 4.28: Evaluation of Method by TQE Sensitivity for the Inventory Domain.

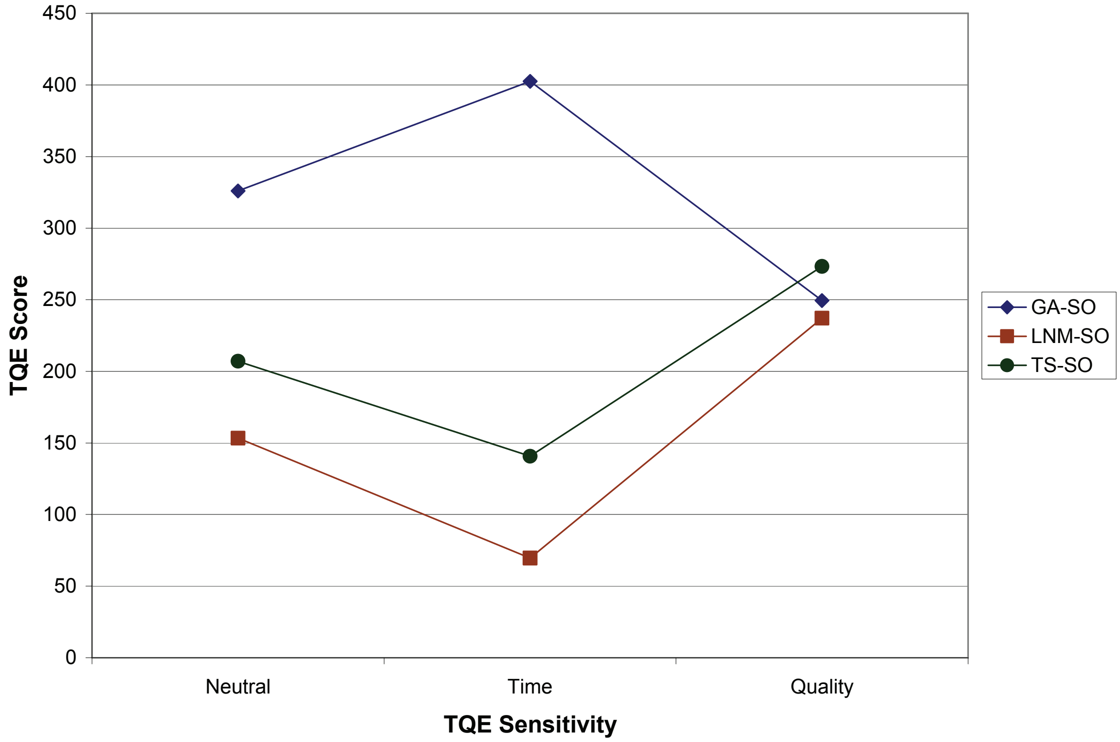


Figure 4.29: Evaluation of Method by TQE Sensitivity for the Logistics Domain.

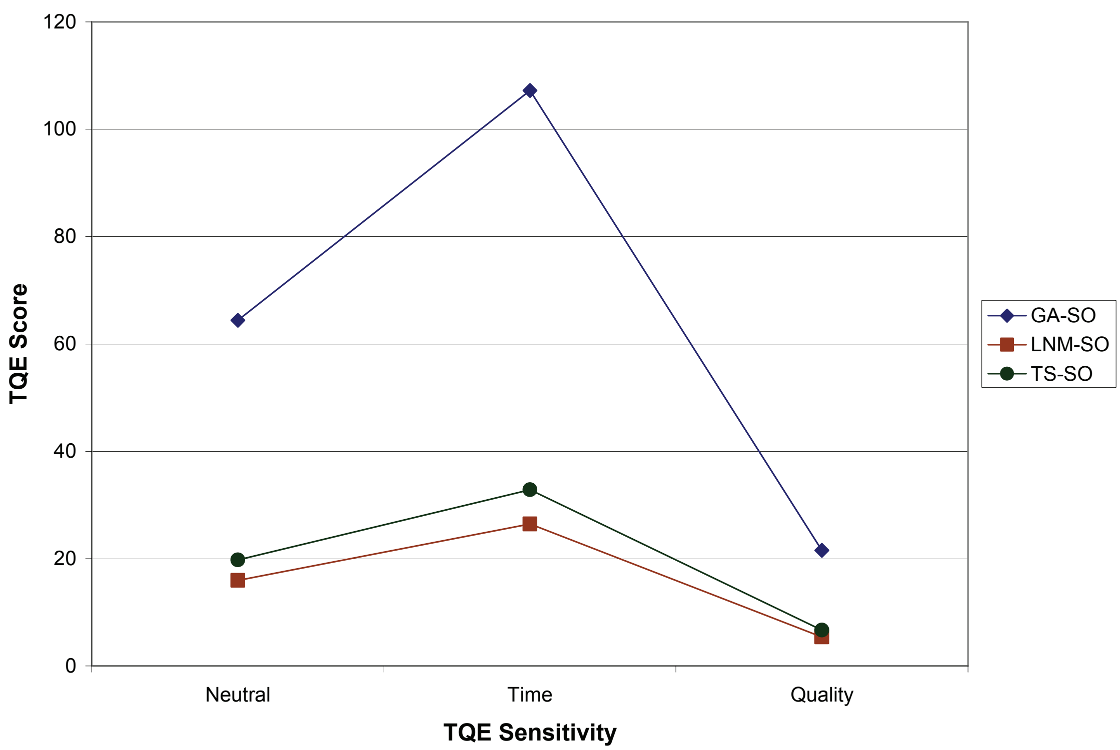


Figure 4.30: Evaluation of Method by TQE Sensitivity for the PERT Domain.

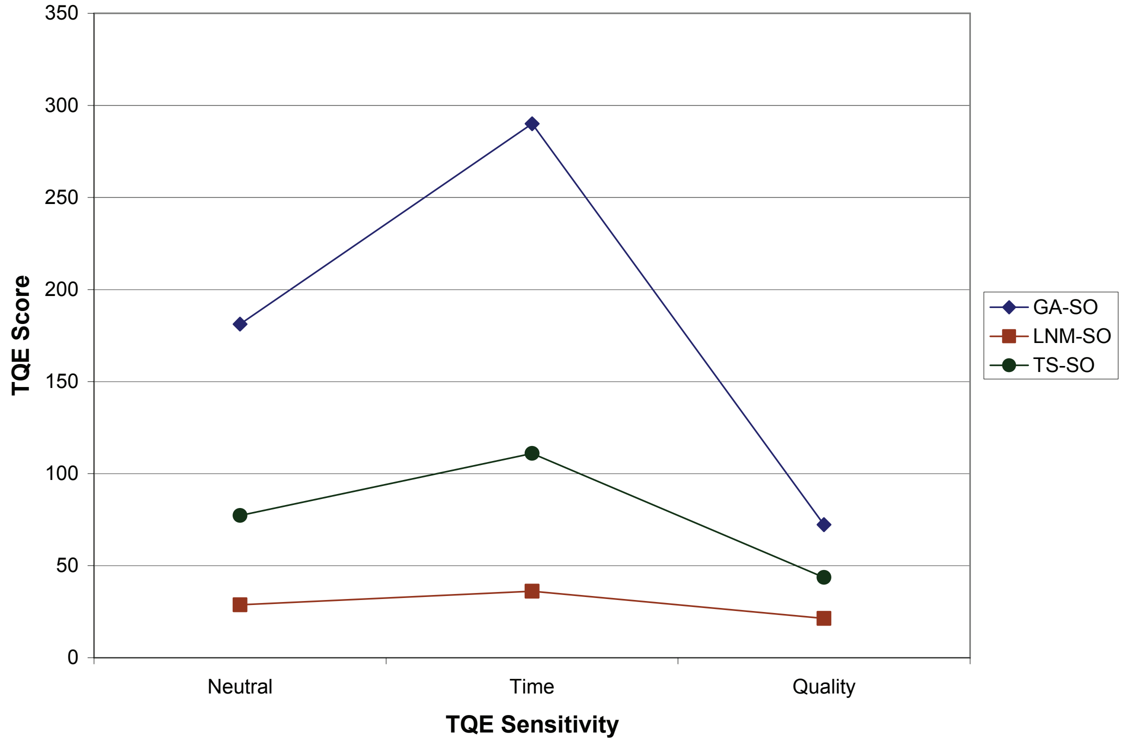


Figure 4.31: Evaluation of Method by TQE Sensitivity for the Production Domain.

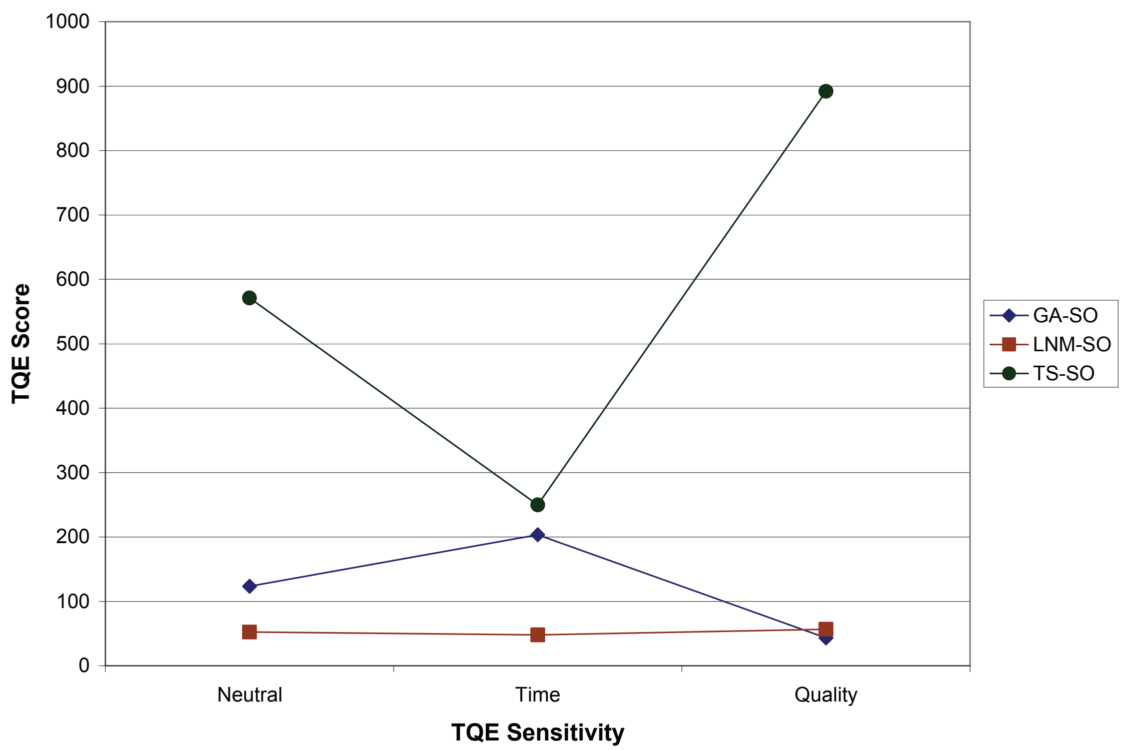


Figure 4.32: Evaluation of Method by TQE Sensitivity for the Reliability Domain.

The summary of the best, second best (or middle), and worst multi-criteria simulation optimization methods for all three time-quality sensitivity levels and for all five domains is summarized in Table 4.27. As was previously, discussed, LNM-SO proved to be the best method in fourteen out of the possible fifteen cases.

Table 4.27: Summary of the Best, Middle, and Worst TQE Scores by Sensitivity and Domain.

	Neutral			Time			Quality		
Domain	**Best**	**Middle**	**Worst**	**Best**	**Middle**	**Worst**	**Best**	**Middle**	**Worst**
INV	LNM-SO	GA-SO	TS-SO	LNM-SO	GA-SO	TS-SO	LNM-SO	GA-SO	TS-SO
LOG	LNM-SO	TS-SO	GA-SO	LNM-SO	TS-SO	GA-SO	LNM-SO	GA-SO	TS-SO
PERT	LNM-SO	TS-SO	GA-SO	LNM-SO	TS-SO	GA-SO	LNM-SO	TS-SO	GA-SO
PROD	LNM-SO	TS-SO	GA-SO	LNM-SO	TS-SO	GA-SO	LNM-SO	TS-SO	GA-SO
RELI	LNM-SO	GA-SO	TS-SO	LNM-SO	GA-SO	TS-SO	GA-SO	LNM-SO	TS-SO

4.3.5 Correlation and R-Squared Analysis

The intercorrelations of the dependent variables were presented in Table 4.28. The closer the value is to 1.0 or –1.0, the closer the two measures are related. From the table, it is clear that the only dependent variables that were highly correlated were ROCCM and ROCBS, with a correlation value of 0.9278. This is logical, since in general, the faster the method finds the best solution, the faster the method should be completed. The rest of the dependent variables are not correlated at all. This makes sense, because they are all very diverse measures and basically operate independently of each other.

Table 4.28: Correlations of the Dependent Variables.

	ROCCM	ROCBS	Count	BSF_Z	Init_Zdif
ROCCM	---	**0.9278**	-0.2975	0.0462	0.0148
ROCBS	---	---	-0.2793	0.0585	0.0377
Count	---	---	---	-0.2773	-0.0734
BSF_Z	---	---	---	---	0.1463
Init_Zdif	---	---	---	---	---

A scatter plot was also done to further examine the relationship between ROCCM and ROC BS. This is shown in Figure 4.33. The figure shows that the data approximately follows the trend line with its intercept occurring at zero. The figure also shows the R-squared value is 0.8329. The trend line shows how closely the projected values are to the actual data. The closer the R-squared value is to one, the more reliable the trend line is. Thus, since the R-squared value equals 0.8329, it suggests the trend line is fairly accurate. There are however, several outliers that skew the results. This suggests that ROCBS and ROCCM are related statistics, but that they still have some independence. This was expected, since having a shorter ROCBS might make it more likely that the problem will also have a shorter ROCCM, but it is not guaranteed.

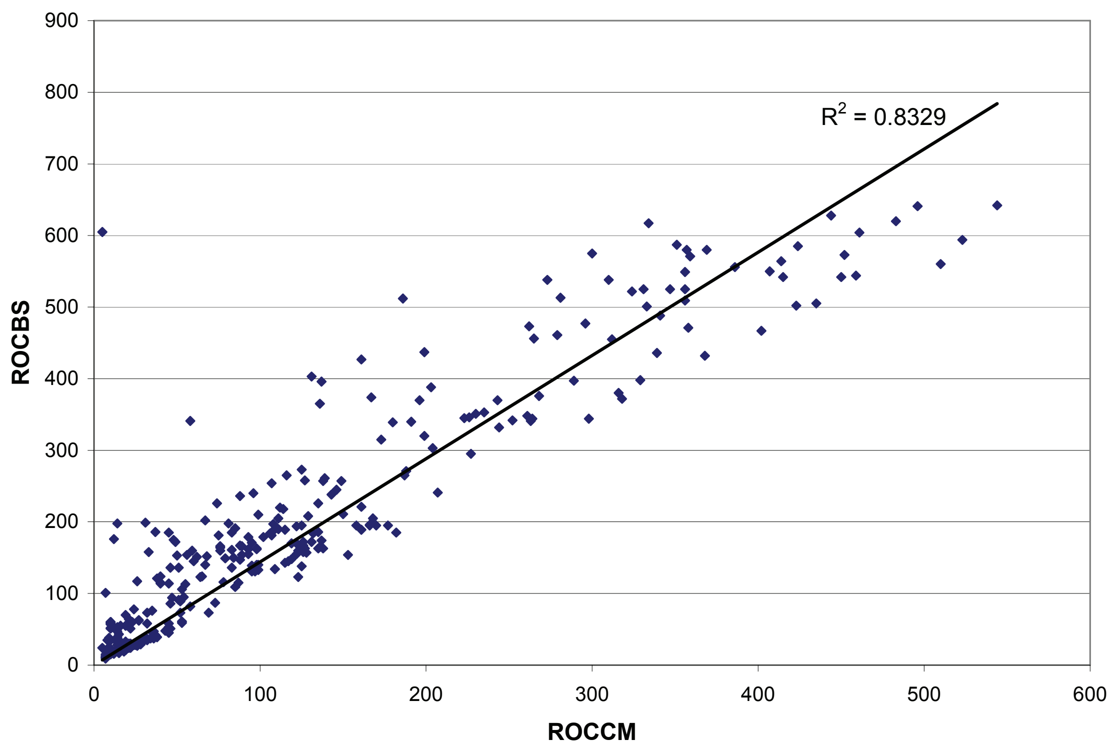


Figure 4.33: Scatter Plot of ROCCM and ROCBS.

Chapter 5

Conclusions and Future Research

This chapter is divided into general conclusions and recommendations for future research.

5.1 General Conclusions

The purpose of this research was to develop global performance measures to allow for comparison between multi-criteria simulation optimization methods. Additionally, this research developed three multi-criteria simulation optimization methods. Finally, this research developed a test bed of five test problems, representing five different domains. This test bed can be used by other researchers for comparison of other solution methods for these problems.

The global performance measures that were developed were intended to address the computational speed, the quality of solution, and a combined measure incorporating both speed and quality. In most simulation optimization methods, the main bottleneck is the number of calls to the simulation. Thus, one way to measure the computational effort is to determine the number of representative operation counts (ROC) of the number of calls to the simulation model. Therefore, computational effort performance measures were the number of calls to the simulation model required to complete the simulation optimization method (ROCCM) and the number of calls to the simulation model required to find the best solution (ROCBS). These values are not dependent on the particular computer or programming language used to implement the simulation optimization methods, as CPU time is. ROCCM and ROCBS proved very effective and were able to replace CPU time as an evaluator of the computational effort.

The quality of solution performance measure was the best solution found (BSF). This performance measure determines the solution examined during the implementation of the simulation optimization method that had the corresponding best objective function value. This is essentially the near-optimal solution found by the multi-criteria simulation optimization method.

The final performance measure quantified the overall performance of the multi-criteria simulation optimization method by combining the computational effort and the quality of the solution found. This measure was referred to as the time-quality estimator (TQE). The equation involved weights for both the quality of solution and the computational effort. Three classes of decision makers were identified based on what component, speed or quality, they were most concerned with. These decision-makers were classified as Time-Sensitive, Quality-Sensitive, or Time-Quality Neutral. Time-sensitive decision-makers desire to find a decent solution in the fastest time possible; quality-sensitive decision makers want to find a high quality solution in a moderate amount of time; while time-quality neutral decision-makers want to find the best solution in the fastest time possible.

The three multi-criteria simulation optimization methods developed through this research incorporated Genetic Algorithms (GA), Tabu Search (TS), and Lexicographic Nelder-Mead (LNM) optimization techniques, along with simulation and goal programming. The specific methods developed were the GA simulation optimization (GA-SO) method, the TS simulation optimization (TS-SO) method, and the LNM simulation optimization (LNM-SO) method. All of the methods used memory to ensure that a solution would not be simulated multiple times during one simulation optimization run. This reduced the potential amount of time required to complete each method. The methods were designed to be robust enough to handle a wide range of test problems and to perform well for both the quality of solution obtained and the speed of execution.

The test problems included five different domains. The domains examined were the inventory (INV) domain, the logistics (LOG) domain, the PERT domain, the production (PROD) domain, and the reliability (RELI) domain. These domains were selected because they have been well studied and are generally well understood. Future researchers should be able to implement these test problems without having to learn a completely new domain structure or specific problem. Thus, these test bed problems should allow for better comparison of future multi-criteria simulation optimization methods, especially when they are performed in conjunction with the proposed global performance measures.

An experimental study involving the test problems from the five domains and the three multi-criteria simulation optimization methods was conducted. The test measures used to analyze the results were ROCCM, ROCBS, Count, BSF, and Init_Zdif. Count was defined as the number of solutions evaluated before the simulation optimization method was completed. This included duplicate solutions. The Init_Zdif was the difference between the objective function value of the initial solution examined and the objective function value of the BSF. Twenty replications were performed, thus, a total of 300 runs were completed and analyzed using a 5×3 factorial design. Additionally, the percent savings between Count and ROCCM and between ROCCM and ROCBS were examined. Finally, TQE was used to evaluate each multi-criteria simulation optimization method.

5.1.1 Computational Effort Conclusions

The following conclusions or points are noteworthy for the problems examined in this dissertation regarding the computational effort:

- Representative operation count (ROC) of the number of calls to the simulation model can effectively replace CPU time as the performance measure for computation effort. ROC is not dependent on the particular computer or programming language used or the skill of the individual programmer as CPU time is; therefore, it is a more accurate measure for comparing problems and multi-criteria simulation optimization methods.
- ROC is more method dependent than problem dependent.
- The LNM-SO method required the fewest ROCCM for all five domains. GA-SO had the worst ROCCM for all but INV.
- PERT required the least ROCCM amongst all domains. On average, PROD was the worst, followed closely by LOG.
- The LNM-SO method also required the fewest ROCBS for all five domains. GA-SO had the worst ROCBS for all but INV.
- PERT required the least ROCBS amongst all domains.
- LNM-SO had the smallest value of Count for all methods. TS-SO required the second smallest value for all but the production domain. The GA-SO method required the largest value of Count for all but the production domain. However, Count was dependent on the structure and the setup of the particular multi-criteria

simulation optimization method, and thus it was determined not to be a very useful measure.

- The percentage savings between Count and ROCCM, however, was a useful, measure. This is because it showed the benefit of including the memory component into the simulation optimization methods. Thus, it shows the actual computational effort saved. LNM-SO had the largest percentage savings between Count and ROCCM, while GA-SO had the worst for all domains but INV. However, all the methods had large savings.
- Another useful statistic is the percentage savings between ROCCM and ROCBS. This shows the potential computational savings if the method could be stopped once ROCBS was obtained. Of course, it is difficult to know when to stop the method so that the value of ROCCM is nearly equal to ROCBS. TS-SO had the greatest percent savings for all but INV. LNM-SO had the best savings for INV. GA-SO had the worst savings for LOG, PERT, and PROD.
- The GA-SO method required the most ROCCM; this was good in that it examined more solutions which may provide more confidence in the final solution obtained. However, it also required the most time.

5.1.2 Quality of Solution Conclusions

The following conclusions or points are noteworthy for the problems examined in this dissertation regarding the quality of solution:

- The GA-SO method found the average BSF for LOG, PERT, and RELI. LNM-SO found the average BSF for INV, PERT (as well), and PROD. TS-SO found the worst average BSF for all domains.

- The GA-SO method found the best overall BSF, by domain, for INV, LOG, and PERT. LNM-SO found the best overall BSF for PERT and PROD. While, TS-SO found the best overall BSF for RELI.
- In general, GA-SO and LNM-SO had less variability than TS-SO. GA-SO and LNM-SO had similar levels of variability.
- The GA-SO method found the best solution the most often, and had the least variability of all the methods.
- The LNM-SO method performed very well and usually found the best or near the best solution.
- The TS-SO method found the best solution only once, and it had the greatest variability of all the methods.
- The TS-SO and LNM-SO methods were more sensitive to the starting solution than the GA-SO method.
- For RELI, GA-SO was very robust and most effective based on Init_Zdif.
- The PERT problem and the PROD problem were very robust based on BSF, since similar results were found by all three methods regardless of the initial solution.

5.1.3 Overall Performance Conclusions

The following conclusions or points are noteworthy for the problems examined in this dissertation regarding the overall performance:

- The LNM-SO method provided very good results and did so in the fewest ROCCM.
- The TS-SO method had a moderate level of ROCCM, but did not generate the best solution very often.

- The PERT domain and the LOG domain problems were very robust and each method solved these about equally well.
- The equation that was derived for TQE was:

$$TQE = \frac{w_S(UCL_{ROCBS}) + w_Q(UCL_{BSF})}{w_S + w_Q} \tag{3.9}$$

  where UCL is the upper confidence limit, w_S is the weight for the computational speed, and w_Q is the weight for the quality of solution.
- The speed and quality weights for determining TQE were determined to be :

  $w_S = w_Q$, for the Time-Quality Neutral Case,

  $w_S = 5$ and $w_Q = 1$, for the Time-Sensitive Case, and

  $w_S = 1$ and $w_Q = 5$, for the Quality-Sensitive Case.
- LNM-SO generated the best TQE score for 14 out of the 15 possible cases. The only case it did not generate the best score was for the quality-sensitive case for RELI. In this case, GA-SO had the best score and LNM-SO had the second best score.
- GA-SO had the best TQE score for 1 case, the second best score for 6 cases, and the worst score for 8 cases.
- TS-SO had the second best TQE score for 8 cases and the worst score for 7 cases.

5.1.3 Final Conclusions

Each of the test problems function well and make excellent test bed problems. Each problem provides an excellent means of comparison for both current and future multi-criteria simulation optimization methods. Additionally, these problems can be easily extended to create further test problems to add to the test bed.

All of the performance measures examined in this research function very well and provide a means of evaluating simulation optimization test problems and for comparing across multi-criteria simulation optimization methods. Additionally, they will work just as well for single-criteria problems and methods. TQE allows a decision-maker to decide whether time or quality is most important to him/her. Thus, the overall effectiveness of the simulation optimization method can thus be evaluated based on the decision-maker's preferences. These measures should all be used in future simulation optimization studies so that future methods can be properly compared to current methods.

In general all the simulation optimization methods performed well. If the quality of solution is the only concern, GA-SO would likely be the best option. If speed is the only concern, LNM-SO or TS-SO would likely be the best option. If however, the decision-maker would like some combination of speed and quality, then LNM-SO would clearly be the preferable method.

5.2 Future Research

There are several areas that could be considered for further study:

1. Examples involving the same number of variables and goals could be examined. This would allow for a more complex structure for the experimental design. It could also provide a more complete analysis of the test problems.
2. More complex problems could be created and examined. This could be done by simply extending the current problems to have additional goals or to have hard constraints. Other means could include adding more variability into the models.

3. Different parameter values for each of the multi-criteria simulation optimization methods could be examined to determine what impact their value has on the method. Examples of parameter values that could be adjusted include: the tabu list sizes, the mutation factor, and the initial simplex calculations. This could provide greater insight into each method.
4. Additionally, the termination criteria for each multi-criteria simulation optimization method could be adjusted to see what if any impact this would have on each method or the test problems. In particular, the minimum number of iterations that must be performed before convergence can be checked could be closely examined. It might be possible to reduce the value of ROCCM without negatively impacting the best solution found. This could provide greater insight into each method.
5. The goals within the problems could be rearranged to see how this affects the results. Changing the goals could also provide more insight into the characteristics of the methods and the test problems. Additionally, it would create in essence an additional set of test problems that could be used for evaluation of future simulation optimization methods.
6. The effect of changing the target values could be examined. It would be desirable to see how examining different levels of the target values would affect the test problems and methods. This again could provide a better understanding of the problems and the methods.
7. An Automatic Target Adjustment (ATA) technique that uses simulation optimization to automatically adjust the target levels for goal programming simulation optimization problems could be developed and evaluated. One possible

approach would be to use simulation optimization to automatically find the target levels that allow for the top "y" number of goals to be satisfied.

In order to accomplish the additional target sets, it would be necessary to determine appropriate target value levels. These values could be determined by adding the appropriate target values to the simulation optimization parameters. Thus, one proposed approach would be to use the simulation optimization methods to automatically find the target levels that will allow all the goals to be satisfied.

It is expected that there will be several possible target values that would satisfy this proposed target set, since all the parameters and not just the target levels would be adjusted multiple times during each iteration. The best target set could then be selected. One proposed formula using the current target value (TVc), the upper target value (TVu), the current solution's objective function value (Zc), and a weight (W) to determine the best target set (BTS) is:

$$BTS = \max\left\{\left(\sum_{i=1}^{g} \frac{TVc_{si}}{TVu_{si}}\right) - W(Zc_s)\right\} \quad \forall s$$

where g is the number of goals and s is the number of solutions examined. TVu could be defined as the maximum value of the absolute value of either the upper or lower range of each target value. The value of W would be set to a significantly large value to ensure that only solutions that satisfied all target values (i.e., Z=0) would be selected. Once the target values are determined, the problem could be run for each simulation optimization method at the newly defined target values.

8. More work could be performed on combining the computational speed and the quality of solution into an overall performance measure. The results for TQE look promising, but more research could provide additional information on its function.

9. The objective function formula could be made to be more adaptable, so that it could be adjusted to best solve each particular problem. One approach for handling this is through the use of neural networks. This is an exciting area that could potentially make these multi-criteria simulation optimization methods more flexible.

These additional areas of research could provide a more complete and potentially a more accurate picture of the effectiveness and the nature of each of the three methods, the five test bed problems, the performance measures, and the state of multi-criteria simulation optimization in general.

Bibliography

1. Abspoel, S.J.; Etman, L.F.; Vervoort, J.; van Rooij, R.A.; Schoofs, A.J.; and Rooda, J.E. (2001). Simulation based optimization of stochastic systems with integer design variables by sequential multipoint linear approximation, *Structural and Multidisciplinary Optimization*, 22, 125-138.

2. Adcock, S. (2006). GAUL (The Genetic Algorithm Utility Library). Available at \url{http://gaul.sourceforge.net/} (10-30-2006). Copyright © 2000-2005 Stewart Adcock.

3. Ahuja, R.K., and Orlin, J.B. (1996). Use of representative operation counts in computational testing of algorithms. *INFORMS Journal of Computing*. 8(3), 318-330.

4. Al-Aomar, R., (2002). A robust simulation-based multicriteria optimization methodology, *Proceedings of the 2002 Winter Simulation Conference*, 1931-1939.

5. Anderson, N.P. (2004). *Simulation Optimization of Logistics Systems Through the Use of Criterion Models*, Ph.D. Dissertation, Department of Industrial Engineering, University of Louisville, Louisville, KY.

6. Anderson, N., Evans, G., and Biles, W. (2006) Simulation optimization of logistics systems through the use of variance reduction techniques and criterion models. *Engineering Optimization,* 38(4), 441-460.

7. Andradottir, Sigrun, (1998). A review of simulation optimization techniques, *Proceedings of the 1998 Winter Simulation Conference,* 151-158.

8. April, J., Better, M., Glover, F., and Kelly, J., (2004). New advances and applications for marrying simulation and optimization, *Proceedings of the 2004 Winter Simulation Conference,* 80-86.

9. April, J., Glover, F., Kelly, J., and Laguna, M., (2003). Practical introduction to simulation optimization, *Proceedings of the 2003 Winter Simulation Conference,* 71-78.

10. April, J., Glover, F., Kelly, J., and Laguna, M., (2001). Simulation/optimization using "real-world" applications, *Proceedings of the 2001 Winter Simulation Conference,* 134-138.

11. Arthur, J.L., and Ravindran, A., (1980a). A branch and bound algorithm with constraint partitioning for integer goal programming problems, *ACM Transactions on Mathematical Software*, 6(3), 378-386.

12. Arthur, J.L., and Ravindran, A., (1980b). PAGP, a partitioning algorithm for (linear) goal programming problems, *European Journal of Operational Research*, 4(6), 421-425.

13. Aurdal, L. (2006). Tabuks.m. Available at \url{http://www.aurdalweb.com/tabuks.m} (10-31-2006). Copyright Lars Aurdal/Rikshospitalet.

14. Avello, E.A., Baesler, F.E., and Moraga, R.J., (2004). A meta-heuristic based on simulated annealing for solving multiple-objective problems in simulation optimization, *Proceedings of the 2004 Winter Simulation Conference*, 508-513.

15. Azadivar, F., and Lee, Y.H., (1998). Optimization of discrete variable stochastic systems by computer simulation, *Mathematics and Computers in Simulation,* 30, 331-345.

16. Azadivar, F., (1999). Simulation optimization methodologies, *Proceedings of the 1999 Winter Simulation Conference,* 93-100.

17. Baesler, F.F., (2000). *Multi-response simulation optimization using stochastic genetic search within a goal programming framework*, Ph.D. Dissertation, Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL.

18. Baesler, F.F., and Sepulveda, J.A. (2001). Multi-objective simulation optimization for a cancer treatment center. *Proceedings of the 2001 Winter Simulation Conference*, 1405-1411.

19. Baesler, F.F., and Sepulveda, J.A., (2000). Multi-response simulation optimization using stochastic genetic search within a goal programming framework. *Proceedings of the 2000 Winter Simulation Conference*, 788-794.

20. Banks, J. and Carson, J.S., (1984). *Discrete-Event System Simulation*, Englewood Cliffs, NJ: Prentice-Hall, Inc.

21. Banks, J., Carson II, J.S., Nelson, B.L., and. Nicol, D.M., (2000). *Discrete-Event System Simulation*, 3rd ed., Upper Saddle River, New Jersey: Prentice-Hall.

22. Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., and Stewart, W.R., (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1, 9-32.

23. Barton, R.S. and Ivey, J.S. Jr., (1996). Nelder-Mead simplex modifications for simulation optimization, *Management Science,* 42(7), 954-973.

24. Bazaraa, M.S.; Jarvis, J.J.; and Sherali, H.D., (1990). *Linear Programming and Network Flows*, New York: John Wiley and Sons.

25. Benayoun, R., de Montgolfier, J., Tergny, J., and Laritchev, O, (1971) Linear programming with multiple objective functions: Step method (stem), *Journal Mathematical Programming*, 1(1), 366-375.

26. Boesel, J., (1999). *Search and Selection for Large-Scale Stochastic Optimization*, Ph.D. Dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

27. Boesel, J., Bowden, R.O., Jr., Glover, F., Kelly, J. and Westwig, E., (2001). Future of simulation optimization, *Proceedings of the 2001 Winter Simulation Conference,* 1466-1469.

28. Bowden, R.O. and Hall, J.D., (1998). Simulation optimization research and development, *Proceedings of the 1998 Winter Simulation Conference,* 1693-1698.

29. Boyle, C.R. and Shin, W.S., (1996). An interactive multiple-response simulation optimization method, *IIE Transactions*, 28, 453-462.

30. Carson, Y. and Maria, A., (1997). Simulation optimization: methods and applications, *Proceedings of the 1997 Winter Simulation Conference,* 118-126.

31. Cavin, L., Fischer, U., Glover, F., and Hungerbuhler, K. (2004). Multi-objective process design in multi-purpose batch plants using a Tabu Search optimization algorithm, *Computers and Chemical Engineering*, 28, 459-478.

32. Chambers, L. (1995). *Practical handbook of genetic algorithms*. Boca Raton, FL: CRC Press.

33. Cheng, R.C.H. and Currie, C.S.M., (2004). Optimization by simulation metamodelling methods, *Proceedings of the 2004 Winter Simulation Conference,* 485-490.

34. Clayton, E.R., Warren, E.W., and Taylor, B.W. (1982). A goal programming approach to the optimization of multiresponse simulation models, *IIE Transactions*, 14(4), 282-287.

35. Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2001). *Introduction to Algorithms*. 2nd ed. Cambridge, Massachusetts: The MIT Press.

36. Curry, G.L. and Hartfiel, D.J. (1983). A simulation-optimization method: its convergence and utility, *Naval Research Logistics Quarterly,* 30, 227-236.

37. Dengiz, B. and Alabas, C. (2000). Simulation optimization using tabu search, *Proceedings of the 2000 Winter Simulation Conference*, 805-810.

38. Eglese, R.W., (1990). Simulated annealing: a tool for operational research, *European Journal of Operational Research*, 46(3), 271-281.

39. Eskandari, H., Rabelo, L, and Mollaghasemi, M. (2005). Multiobjective simulation optimization using an enhanced genetic algorithm, *Proceedings of the 2005 Winter Simulation Conference*, 833-841.

40. Evans, G.W. and Anderson, N.P., (2006). Development of criterion models for use in simulation analysis, to appear in *Proceedings of the 2006 Winter Simulation Conference*.

41. Evans, G.W., Stuckman, B., and Mollaghesemi, M., (1991). Multicriteria optimization of simulation models, *Proceedings of the 1991 Winter Simulation Conference*, 894-900.

42. Fleischer, M., (1995). Simulated annealing: past, present, and future, *Proceedings of the 1995 Winter Simulation Conference*, 155-161.

43. Freund, J.E. (1992). *Mathematical Statistics*, 5th ed., Englewood Cliffs, NJ: Prentice Hall.

44. Fu, M.C., Andradottir, S., Carson, J.C., Glover, F., Harrell, C.R., Ho, Y.C., Kelly, J.P. and Robinson, S.M., (2000). Integrating optimization and simulation: research and practice, *Proceedings of the 2000 Winter Simulation Conference,* 610-616.

45. Fu, M.C., (2002). Optimization for simulation: theory vs. practice, *INFORMS Journal on Computing, 14, 3*, 192-215.

46. Fu, M.C. (2001). Simulation optimization. *Proceedings of the 2001 Winter Simulation Conference*, 53-61.

47. Fu, M.C., (1994a). Simulation optimization: a review, *Annals of Operations Research*, 53, 199-248.

48. Fu, M.C., (1994b). Sample path derivatives for (s,S) inventory systems, *Operations Research*, 42, 351-364.

49. Glover, F. (1990). Tabu Search – Part II. *ORSA Journal on Computing*. 2(1), 4-32.

50. Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing*. 1(3), 190-206.

51. Glover, F. (1986). Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, 13, 533-549.

52. Glover, F. and Laguna, M. (1997): *Tabu Search*, Norwell, MA: Kluwer.

53. Glynn, P.W., (2002). Additional perspectives on simulation for optimization, *INFORMS Journal on Computing,* 14(3) 220-222.

54. Guimaraes, R.C. and Kingsman, B.G., (1989). Simulation-optimisation: the method and its application in the analysis of grain terminal operations, *European Journal of Operational Research,* 41, 44-53.

55. Hanne, T. (2001). *Intelligent Strategies for Meta Multiple Criteria Decision Making*. Boston: Kluwer Academic Publishers.

56. Henderson, S.G. and Mason, A.J., (1998). Rostering by iterating integer programming and simulation, *Proceedings of the 1998 Winter Simulation Conference,* 677-683.

57. Hertz, A., Taillard, E., and de Werra, D., (1997). A tutorial on tabu search, in E.H.L. Aarts and J.K. Lenstra, *Local search in combinatorial optimization*. Wiley, 121-136.

58. Hong, L.J. and Nelson, B.L., (2006). Discrete optimization via simulation using COMPASS, *Operations Research*, 54(1), 115-129.

59. Hong, L.J. and Nelson, B.L. (2005). A framework of locally convergent random-search algorithms for discrete optimization via simulation. Working paper, Department of Industrial Engineering and Management Sciences, Northwestern University.

60. Hooker, J.N. (1995). Testing heuristics: we have it all wrong. *Journal of Heuristics*, 1, 33-42.

61. Hooker, J.N. (1994). Needed: an empirical science of algorithms. *Operations Research*, 42(2), 201-212.

62. Ignizio, J., (1976). *Goal Programming and Extensions*, Lexington, MA: Lexington Books.

63. Kelton, W.D., Sadowski, R.P., and Sturrock, D.T., (2007). *Simulation with Arena*, 4th ed., Boston: McGraw Hill.

64. Keys, A.C., Rees, L.P., and Greenwood, A.G., (2002). Performance measures for selection of metamodels to be used in simulation optimization. *Decision Sciences*, 33, 31-57.

65. Kim, S-H., and Nelson, B.L. (2005). Selecting the best system. In: Henderson, S.G. and Nelson, B.L. (editors), *Handbooks in Operations Research and Management Science: Simulation, ed*, Chapter 17. Oxford: Elsevier Science.

66. Konak, A. and Kulturel-Konak, S. (2005). Simulation optimization using tabu search: an empirical study, *Proceedings of the 2005 Winter Simulation Conference.* 686-2692.

67. Kuriger, G., (1998). *Intelligent Search Methods for Nonlinear Goal Programming Problems*. Master's Thesis, University of Oklahoma, Norman, OK.

68. Kuriger, G. and Ravindran, A., (2005). Intelligent search methods for nonlinear goal programming problems, *Information Systems and Operational Research,* 43(2), 79-92.

69. Law, A.M. and Kelton, W.D. (2000). *Simulation Modeling and Analysis*. 3rd ed., Boston: McGraw Hill.

70. Law, A.M. and McComas, M.G., (2002). Simulation-based optimization, *Proceedings of the 2002 Winter Simulation Conference,* 41-44.

71. Lee, S.M., and. Morris, R.L. (1977). Integer goal programming methods. *TIMS Studies in the Management Sciences*. 6, 273-289.

72. L'Ecuyer, P.L. (1996). Simulation of algorithms for performance analysis. *INFORMS Journal on Computing*, 8(1), 16-20.

73. Lin, B.W., and Rardin, R.L. (1980). Controlled experimental design for statistical comparison of integer programming algorithms. *Management Science*, 25(12), 1258-1271.

74. Mason, A.J., Ryan, D.M., and Panton, D.M., (1998). Integrated simulation, heuristic and optimisation approaches to staff scheduling, *Operations Research,* 46(2), 161-175.

75. McGeoch, C.C. (1996a). Toward an experimental method for algorithm simulation. *INFORMS Journal on Computing*, 8(1), 1-15.

76. McGeoch, C.C. (1996b). Challenges in algorithm simulation. *INFORMS Journal on Computing*, 8(1), 27-28.

77. Mollaghesemi, M. and Evans, G.W., (1994). Multicriteria design of manufacturing systems through simulation optimization. *Transactions on Systems, Man, and Cybernetics*, 24(9), 1407-1411.

78. Molnar, B., (2005). Multi-criteria scheduling of order picking processes with simulation optimization, *Periodica Polytechnica Ser. Transp. Eng.*, 33(1-2), 59-68.

79. Montgomery, D.C. (1991). *Design and Analysis of Experiments*, 3rd ed., New York, NY: John Wiley and Sons.

80. Nelder, J.A. and Mead, R., (1965). A simplex method for function minimization, *Computer Journal.,* 7, 308-313.

81. Nelson, B.L. and Staum, J., (2006). Control variates for screening, selection and estimation of the best, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(1), 52 – 75.

82. Nwana, V., Darby-Dowman, K., and Mitra, G. (2005). A co-operative parallel heuristic for mixed zero-one linear programming: combining simulated annealing with branch and bound, *European Journal of Operational Research,* 164, 12-23.

83. Olafsson, S., and Kim, J., (2002). Simulation optimization, *Proceedings of the 2002 Winter Simulation Conference,* 79-84.

84. Orlin, J.B. (1996). On experimental methods for algorithm simulation*. INFORMS Journal on Computing,* 8(1), 21-23.

85. Pichitlamken, J. (2002). *A Combined Procedure for Simulation Optimization*, Ph.D. Dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

86. Pichitlamken, J. and Nelson, B.L. (2003). A combined procedure for simulation optimization, *ACM Transactions on Modeling and Computer Simulation*, 13(2), 155-179.

87. Pichitlamken, J., Nelson, B.L., and Hong, L.J., (2004). A sequential procedure for neighborhood selection-of-the-best in optimization via simulation, to appear in *European Journal of Operational Research*.

88. Pritsker, A.A. and O'Reilly, J.J. (1999). *Simulation with Visual SLAM and AweSim*. 2nd ed., New York: John Wiley & Sons.

89. Pritsker, A.A., Sigal, C.E., and Hammesfahr, R.D. (1994). *Slam II: Network Models for Decision Support*. The Scientific Press.

90. Rardin, R.L. (1998). *Optimization in Operations Research*. Saddle River, NJ: Prentice Hall.

91. Rardin, R.L, and Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: a tutorial. *Journal of Heuristics*, 7, 261-304.

92. Ravindran, A., Phillips, D.T., and Solberg, J.J., (1987). *Operations Research Principles and Practice*, 2nd ed., New York, NY: John Wiley & Sons, Inc.

93. Rees, L.P., Clayton, E.R., and Taylor, B.W. (1985). Solving multiple response simulation models using modified response surface methodology within a lexicographic goal programming framework. *IIE Transactions*, 17(1), 47-57.

94. Reeves, C.R., (1996). Modern heuristic techniques. In: Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., and Smith, G.D. (editors), *Modern Heuristic Search Methods*, Chichester, England: John Wiley and Sons, 1-26.

95. Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M., (1983). *Engineering Optimization: Methods and Applications*, New York, NY: John Wiley & Sons, Inc.

96. Saber, H.M. and Ravindran, A., (1995). A partitioning gradient based (PGB) algorithm for solving nonlinear goal programming problems, *Computers and Operations Research,* 23, 141-152.

97. Safizadeh, M.H., (1990). Optimization in simulation: current issues and the future outlook, *Naval Research Logistics*, 37, 807-825.

98. Schniederjans, M.J., (1995). *Goal Programming: Methodology and Applications*. Boston: Kluwer Academic Publishers, 49.

99. Schniederjans, M.J., (1984). *Linear Goal Programming*. Princeton, New Jersey: Petrocelli Books, 98-102.

100. Shier, D.R. (1996). On algorithm analysis. *INFORMS Journal on Computing*, 8(1), 24-26.

101. Spendley, W., Hext, G.R., and Himsworth, F.R., (1962). Sequential application of simplex designs in optimization and evolutionary operation, *Technometrics,* 4, 441-461.

102. Swisher, J.R., Hyden, P.D., Jacobson, S.H. and Schruben, L.W., (2000). A survey of simulation optimization techniques and procedures, *Proceedings of the 2000 Winter Simulation Conference,* 119-128.

103. Szidarovszky, F., and Eskandari, A. (1999). A simulation aided solution to an MCDM problem, *Proceedings of the 1999 Winter Simulation Conference,* 573-577.

104. Tekin, E. and Sabuncuoglu, I. (2004). Simulation optimization: a comprehensive review on theory and applications. *IIE Transactions*. 36, 1067-1081.

105. Teleb, R. and Azadivar, F., (1994). A methodology for solving multi-objective simulation-optimization problems, *European Journal of Operational Research*, 72, 135-145.

106. Wall, M. (1996). *GAlib: A C++ Library of Genetic Algorithm Components*. Version 2.4, Documentation Revision B, Available at \url{http://lancet.mit.edu/ga/dist/galibdoc.pdf} (10-31-2006). Copyright Mathew Wall.

107. Winston, W.L., (2004). *Operations Research: Applications and Algorithms*, 4th ed., Belmont, CA: Brooks/Cole – Thomson Learning.

108. Wolsey, L.A., (1998). *Integer Programming*, New York, NY: John Wiley & Sons, Inc., 207-211.

109. Yang, T., Kuo, Y., and Chang, I., (2004). Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors—a case study, *International Journal of Production Research,* 42(19), 4015-4030.

APPENDIX A

COMPUTER CODES FOR EACH OPTIMIZATION METHOD

GA-SO Method

The GA-SO method was programmed using C++ and used the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology (Copyright (c) 1995-1996 Massachusetts Institute of Technology (MIT) and Copyright (c) 1996-2005 Matthew Wall (the Author). All rights reserved).The specific coding and the file used to call the simulation model for each domain are presented below.

Note that the GA-SO method was programmed as a maximization problem. Thus, values calculated and presented later should be multiplied by negative one.

Inventory Domain

```
/* ------------------------------------------------------------------------------------------------
DESCRIPTION: Genetic Algorithm Simulation Optimization (GA-SO) Method
 This problem examines 3 Variables, 3 Goals, using genetic algorithms
 to solve an inventory domain test bed problem.
---------------------------------------------------------------------------------------------------- */
#include <iostream>
#include <fstream>     // needed for direct file manipulation
#include <string>
#include <stdlib.h>
#include <time.h>
#include <cmath>

#include <stdio.h>
#include <ga/ga.h>
#include <ga/std_stream.h>

#define cout STD_COUT
using namespace std;

float objective(GAGenome &);
int calls = 0;
int count1 = 0;
double begin;

int
```

```
main(int argc, char **argv)
{

	time_t start;
	time(&start);
	begin = start;


	// we initially want to create a blank file called sampleout.dat, sampleout1.dat, and
	   sampleout3.dat
	// we do this in case the file already exists, so that the data is not carried over
	   between experiments
	ofstream outfil1("C:/galib/problems/invent/inv_out1.dat", ios::out);
	ofstream outfil2("C:/galib/problems/invent/inv_out2.dat", ios::out);
	ofstream outfil3("C:/galib/problems/invent/inv_out3.dat", ios::out);
	ofstream outfil4("C:/galib/problems/invent/inv_out4.dat", ios::out);
	ofstream outfil6("C:/galib/problems/invent/inv_out6.dat", ios::out);
	ofstream outfil7("C:/galib/problems/invent/inv_out7.dat", ios::out);
	ofstream outfil8("C:/galib/problems/invent/inv_out8.dat", ios::out);
	ofstream outfil9("C:/galib/problems/invent/inv_results.dat", ios:: out);

	// here we seed an initial value of 0's into sampleou2.dat, so our loop will have
	   data to start searching the first time through.
	outfil2 << 0 << '\t' << 0.0 << endl;
	outfil7 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "Q" << '\t' <<
	     "R" << '\t' << "TBREV" << '\t' << "LT" << '\t' << "TBD" << '\t' <<"Z" << '\t'
	     << "Goal1" << '\t' << "Goal2" << '\t'<< "Goal3" << '\t' << "TS" << '\t' <<
	     "TO" << '\t' << "TR" << '\t' << "AvgInv" << '\t'<< "Runtime" << '\t' << "LS"
	     << '\t' << "BO" << '\t' << "BWT" << '\t'<< "CWT" << '\n';
	outfil8 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "Q" << '\t' <<
	     "R" << '\t' << "TBREV" << '\t' << "LT" << '\t' << "TBD" << '\t' <<"Z" << '\t'
	     << "Goal1" << '\t' << "Goal2" << '\t'<< "Goal3" << '\t' << "TS" << '\t' <<
	     "TO" << '\t' << "TR" << '\t' << "AvgInv" << '\t'<< "Runtime" << '\t' << "LS"
	     << '\t' << "BO" << '\t' << "BWT" << '\t'<< "CWT" << '\n';

	outfil1.close();
	outfil2.close();
	outfil3.close();
	outfil4.close();
	outfil6.close();
	outfil7.close();
	outfil8.close();

	cout << "Test Problem\n\n";
	cout << "This program finds the maximum value in the function\n";
	cout << "  TP\n";
```

```
cout << "with the constraints\n";
cout << "     30 <= Q <= 100\n";
cout << "      0 <= R <= 30\n";
cout << "      1 <= TBREV <= 10\n";
cout << "\n\n"; cout.flush();

// See if we've been given a seed to use (for testing purposes).  When you
// specify a random seed, the evolution will be exactly the same each time
// you use that seed number.

unsigned int seed = 0;
for(int i=1; i<argc; i++) {
        if(strcmp(argv[i++],"seed") == 0) {
                seed = atoi(argv[i]);
        }
}

// Declare variables for the GA parameters and set them to some default values.

int popsize  = 30;
int ngen     = 50;
float pmut   = 0.01;
float pcross = 0.6;
float pconv = 0.99;     //Convergence threshold
float nconv = 100;      //Number of generations to examine

// Create a phenotype for with the appropriate range for each variable.

GABin2DecPhenotype map;
map.add(4, 30, 100);
map.add(4, 0, 30);
map.add(4, 1, 10);


// Create the template genome using the phenotype map we just made.

GABin2DecGenome genome(map, objective);

// Now create the GA using the genome and run it.  We'll use sigma truncation
// scaling so that we can handle negative objective scores.

GASimpleGA ga(genome);
GASigmaTruncationScaling scaling;
ga.populationSize(popsize);
ga.pConvergence(pconv);
ga.nConvergence(nconv);
```

```
ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);
ga.pMutation(pmut);
ga.pCrossover(pcross);
ga.scaling(scaling);
ga.scoreFilename("C:/galib/problems/invent/bog3.dat");
ga.scoreFrequency(1);
ga.flushFrequency(50);
ga.evolve(seed);

// Dump the results of the GA to the screen.
double Q, R, TBREV;
int LT, TBD, Q1, R1, TBREV1;
genome = ga.statistics().bestIndividual();
Q = genome.phenotype(0);
R = genome.phenotype(1);
TBREV = genome.phenotype(2);
LT = 0;
TBD = 0;

// The following code, converts the real number to only 2 decimal points
Q1 = (100 * Q) + 0.5*(Q/fabs(Q));
R1 = (100 * R) + 0.5*(R/fabs(R));
TBREV1 = (100 * TBREV) + 0.5*(TBREV/fabs(TBREV));
Q = Q1/100.0;
R = R1/100.0;
TBREV = TBREV1/100.0;

Q1 = (100 * Q) + 0.5*(Q/fabs(Q));
R1 = (100 * R) + 0.5*(R/fabs(R));
TBREV1 = (100 * TBREV) + 0.5*(TBREV/fabs(TBREV));
Q = Q1/100.0;
R = R1/100.0;
TBREV = TBREV1/100.0;

cout << "the ga found an optimum at the point (";
cout << Q << ", " << R << ", " << TBREV <<")\n\n";

cout << "best of generation data are in '" << ga.scoreFilename() << "'\n";
cout << "The GA found: \n" <<ga.statistics().bestIndividual() <<"\n";

time_t end;
time(&end);
double timedif = end - start;
clock_t clock(void);

cout << "time difference = " << timedif << " seconds" << endl;
```

```
	cout << "time difference = " << timedif/60 << " minutes" << endl;
	cout << "the number of calls to the simulation = " << calls << endl;
	cout << "then number of generations was = " << ga.generation() << endl;

	outfil9 <<	"time difference = " << timedif << " seconds" << endl << "time
		difference = " << timedif/60 << " minutes" << endl << "the number of calls to
		the simulation = " << calls << endl << "then number of generations was = "
		<< ga.generation() << endl;
	outfil9.close();
	return 0;
}

// This objective function tries to minimize the value of the function
//
//			Z = P1*d1+ + P2*d2+ + P3*d3+
//			Goal 1: max  TP = TS*(price per unit - cost per unit) - TO*(cost
//				per order) - TR*(cost per review) - AveInv*(holding cost
//				per unit)*(time period) - LS*(cost per lost sale) - BO*(cost
//				per backorder)
//				therefore, Goal 1: min -TP
//			Goal 2:min CWT
//			Goal 3: min BWT
//

float
objective(GAGenome & c)
{
	GABin2DecGenome & genome = (GABin2DecGenome &)c;

	ifstream infile1("C:/galib/problems/Core_GA_output.dat", ios::in);
			// associate Core_GA_output.dat with infile1
			// this file will hold the output from AweSim needed to guide the
			optimization program
			// specifically, the total sales (TS), the total orders (TO), the total
			reviews (TR), and the average inventory(AvgInv)
	ofstream outfile("C:/galib/problems/GA_data.dat", ios::out);
			// associate GA_data.dat with outfile
			// this file will hold the input data for Awesim
			// specifically, the number to order (Q), the re-order point (R), and the
			time between reviews (TBREV)
	ofstream outfile1("C:/galib/problems/invent/inv_pts_visited1.dat", ios::app);
			// associate ga1_out1.dat with outfile1, set it up to add to the file each
			time through
			// this file will store all visited points, and the corresponding Z and
			goal values
	ofstream outfile2("C:/galib/problems/invent/inv_out2.dat", ios::app);
```

```
            // associate ga1_out2.dat with outfile2, set it up to add to the file each
            time through
            // this file will store all visited points, want to use it to compare if
            point previously visited
ofstream outfile4("C:/galib/problems/invent/inv_out4.dat", ios::app);
            // associate ga1_out2.dat with outfile4, set it up to add to the file each
            time through
ofstream outfile5("C:/galib/problems/invent/inv_out5.dat", ios::out);
            // associate ga1_out5.dat with outfile5, set it up to add to the file each
            time through
            // trying to see if this will work
ofstream outfile6("C:/galib/problems/invent/inv_out6.dat", ios::app);
            // associate ga1_out6.dat with outfile6, set it up to add to the file each
            time through
            // trying to see if this will work
ofstream outfile7("C:/galib/problems/invent/inv_out7.dat", ios::app);
            // associate ga1_out7.dat with outfile7, set it up to add to the file each
            time through
            // want to store visited points, and simulation output data to check
            calculations
ofstream outfile8("C:/galib/problems/invent/inv_out8.dat", ios::app);
            // associate ga1_out8.dat with outfile8, set it up to add to the file each
            time through
            // want to store visited points, and simulation output data to check
            calculations

outfile1.setf(ios::fixed);
outfile6.setf(ios::fixed);
outfile7.setf(ios::fixed);
outfile8.setf(ios::fixed);


double Z, goal[3], targets[3], Runtime;
double Q, R, TBREV;
int LT, TBD, g, count = 0, i = 0;
double P;
double TS, TO, TR, LS, AvgInv, number, BWT, CWT, BO, epsilon;
string hold, test;
double timedif;
int Q1, R1, TBREV1;

// associate Q, R, and TBREV with their corresponding genome.phenotypes
Q = genome.phenotype(0);
R = genome.phenotype(1);
TBREV = genome.phenotype(2);
LT = 0;
```

```
TBD = 0;

// The following code, converts the real number to only 2 decimal points
Q1 = (100 * Q) + 0.5*(Q/fabs(Q));
R1 = (100 * R) + 0.5*(R/fabs(R));
TBREV1 = (100 * TBREV) + 0.5*(TBREV/fabs(TBREV));
Q = Q1/100.0;
R = R1/100.0;
TBREV = TBREV1/100.0;

// define the target values for each goal
targets[0] = 12000;
targets[1] = -0.05;
targets[2] = -0.225;

// write the input data for AweSim
outfile << Q << '\t' << R << '\t' << TBREV << '\t' << LT << '\t' << TBD << endl;

outfile5 << Q1 << R1 << TBREV1 << endl;
outfile.close();
outfile5.close();

ifstream infile2("C:/galib/problems/invent/inv_out2.dat", ios::in);
            // associate ga1_out2.dat with infile2
            // note that this is also the file associated with outfile2
            // we use this to see if we have previously visited a point or not
ifstream infile5("C:/galib/problems/invent/inv_out5.dat", ios::in);
            // associate ga1_out5.dat with infile5
            // note that this is also the file associated with outfile2
            // we use this to determine the current point

infile5 >> test;

infile2 >> hold >> number;
count1++;
cout << "count1 = " << count1 << endl;

while (!infile2.fail() && i < 10000)
{
        i++;
        count++;
        if (hold == test) {
                Z = number;
                i = 12000;
        }
        else {
```

```
			infile2 >> hold >> number;
		}
	}

	infile2.close();
	infile5.close();

	outfile4 << hold << endl;

	// only call AweSim if the point has not previously been visited
	if (i != 12000)
	{
		// run AweSim and generate the output
		cout << "(" << Q << ", " << R << ", " << TBREV << ")\n\n";  // Write out
		   current point being simulated
		system ("C:/galib/problems/invent/call_INV.bat");  // this calls a file that
		   calls the simulation model
		calls = calls + 1;
		cout << "calls = " << calls << endl;

		// collect the output from the AweSim runs
		infile1 >> TS >> TO >> TR >> AvgInv >> Runtime >> LS >> CWT >>
		   BWT >> BO;

		// For goal 0, we want to maximize the total profit (TP)
		// For goal 1, we want to minimize the customer wait time (CWT)
		// For goal 2, we want to minimize the bin wait time (BWT)

		goal[0] = ((TS * (65 - 40)) - (TO * 50) - (TR * 30) - (AvgInv * .004 *
				Runtime) - (LS * 20) - (BO *10));
		goal[1] = -1 * BWT;
		goal[2] = -1 * CWT;

		// loop to calculate Z
		Z = 0;
		epsilon = 0.00005;
		g = 2;				// where g is the number of goals - 1 (to
						   work with C++ notation)
		for (int j = 0; j<=g; j++)
		{
			P = pow(100,(g-j));
			if (goal[j] >= (targets[j] - epsilon))
			{
				goal[j] = targets[j];
			}
```

```
            // we want to calculate the normalized objective function
            // following the equation, Z = sum (Pi*di/Ti)
            Z = Z + ((goal[j] - targets[j])/fabs(targets[j]))*P;

        }
        Z = 1*Z;
        time_t temp;
        time(&temp);
        timedif = temp - begin;
        clock_t clock(void);

        outfile8 << i << '\t' << calls << '\t' << timedif << '\t' << Q << '\t' << R <<
            '\t' << TBREV << '\t' << LT << '\t' << TBD <<'\t' << Z << '\t' <<
            goal[0] << '\t' << goal[1] << '\t' << goal[2] << '\t' << TS << '\t' << TO
            << '\t' << TR << '\t' << AvgInv << '\t' << Runtime << '\t' << LS << '\t'
            << BO << '\t' << BWT << '\t' << CWT << '\n';
        outfile6 << Q1 << R1 << TBREV1 << '\t' << Q << '\t' << R << '\t' <<
            TBREV << '\t' << LT << '\t' << TBD << '\t' << Z << '\t' << "count="
            << count << '\t' << "i=" << i << '\n';

    }

    // write out the results
    outfile1 << Q << '\t' << R << '\t' << TBREV << '\t' << Z << '\t' << goal[0] << '\t'
        << goal[1] << '\t' << goal[2] << '\t' << "count=" << count << '\t' << "i=" << i
        << '\n';
    outfile2 << Q1 << R1 << TBREV1 << '\t' << Z << '\n';
    outfile7 << i << '\t' << calls << '\t' << timedif << '\t' << Q << '\t' << R << '\t' <<
        TBREV << '\t' << LT << '\t' << TBD <<'\t' << Z << '\t' << goal[0] << '\t' <<
        goal[1] << '\t' << goal[2] << '\t' << TS << '\t' << TO << '\t' << TR << '\t' <<
        AvgInv << '\t' << Runtime << '\t' << LS << '\t' << BO << '\t' << BWT << '\t'
        << CWT << '\n';


    infile1.close();
    outfile1.close();
    outfile2.close();
    outfile4.close();
    outfile6.close();
    outfile7.close();
    outfile8.close();

    return Z;

}
```

Here is the file "call_INV.bat";

```
cd\
cd projects\dissert
execute invent
cd\
cd galib\problems\invent
```

Logistics Domain

```
/* -------------------------------------------------------------------------
DESCRIPTION: Genetic Algorithm Simulation Optimization (GA-SO) Method
 This problem examines 5 Variables, 2 Goals, using genetic algorithms
 to solve a logistics domain test bed problem.
------------------------------------------------------------------------- */
#include <iostream>
#include <fstream>     // needed for direct file manipulation
#include <string>
#include <stdlib.h>
#include <time.h>
#include <cmath>

#include <stdio.h>
#include <ga/ga.h>
#include <ga/std_stream.h>

#define cout STD_COUT
using namespace std;

float objective(GAGenome &);
int calls = 0;
int count1 = 0;
double begin;

int
main(int argc, char **argv)
{

        time_t start;
        time(&start);
        begin = start;

        // we initially want to create a blank file called sampleout.dat, sampleout1.dat, and
           sampleout3.dat
        // we do this in case the file already exists, so that the data is not carried over
           between experiments
        ofstream outfil1("C:/galib/problems/logistic/logout1.dat", ios::out);
        ofstream outfil2("C:/galib/problems/logistic/logout2.dat", ios::out);
        ofstream outfil3("C:/galib/problems/logistic/logout3.dat", ios::out);
        ofstream outfil4("C:/galib/problems/logistic/logout4.dat", ios::out);
        ofstream outfil6("C:/galib/problems/logistic/logout6.dat", ios::out);
        ofstream outfil7("C:/galib/problems/logistic/logout7.dat", ios::out);
        ofstream outfil8("C:/galib/problems/logistic/logout8.dat", ios::out);
        ofstream outfil9("C:/galib/problems/logistic/log_results.dat", ios:: out);
```

```
// here we seed an initial value of 0's into sampleou2.dat, so our loop will have
   data to start searching the first time through.
outfil2 << 0.0 << '\t' << 0.0 << endl;
outfil7 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "FAIL" << '\t' <<
     "ARRIVE" << '\t' << "DISC1" << '\t' << "DISC2" << '\t' << "DISC3" << '\t'
     << "Z" << '\t' << "Goal1" << '\t' << "Goal2" << '\t'<< "ORD1_AVG" << '\t'
     << "ORD2_AVG" << '\t' << "ORD3_AVG" << '\t'<< "OUT1_AVG" << '\t'<<
     "OUT2_AVG" << '\t' << "OUT3_AVG" << '\t' << "OUT1_STD" << '\t'<<
     "OUT2_STD" << '\t' << "OUT3_STD" << '\t' << "UTIL_AVG" << '\t' <<
     "UTIL_STD" << '\n';
outfil8 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "FAIL" << '\t' <<
     "ARRIVE" << '\t' << "DISC1" << '\t' << "DISC2" << '\t' << "DISC3" << '\t'
     << "Z" << '\t' << "Goal1" << '\t' << "Goal2" << '\t'<< "ORD1_AVG" << '\t'
     << "ORD2_AVG" << '\t' << "ORD3_AVG" << '\t'<< "OUT1_AVG" << '\t'<<
     "OUT2_AVG" << '\t' << "OUT3_AVG" << '\t' << "OUT1_STD" << '\t'<<
     "OUT2_STD" << '\t' << "OUT3_STD" << '\t' << "UTIL_AVG" << '\t' <<
     "UTIL_STD" << '\n';

outfil1.close();
outfil2.close();
outfil3.close();
outfil4.close();
outfil6.close();
outfil7.close();
outfil8.close();

cout << "Test Problem\n\n";
cout << "This program finds the minimum cost and the minimum project
            completion time\n";
cout << "  TC\n";
cout << "with the constraints\n";
cout << "     5 <= LOW3 <= 20\n";
cout << "     1 <= LOW4 <= 8\n";
cout << "     1 <= LOW6 <= 8\n";
cout << "\n\n"; cout.flush();

// See if we've been given a seed to use (for testing purposes).  When you
// specify a random seed, the evolution will be exactly the same each time
// you use that seed number.

unsigned int seed = 0;
for(int i=1; i<argc; i++) {
        if(strcmp(argv[i++],"seed") == 0) {
                seed = atoi(argv[i]);
        }
```

```
}

// Declare variables for the GA parameters and set them to some default values.

int popsize  = 30;
int ngen     = 50;
float pmut   = 0.01;
float pcross = 0.6;
float pconv = 0.99;     //Convergence threshold
float nconv = 100;      //Number of generations to examine

// Create a phenotype for with the appropriate range for each variable.

GABin2DecPhenotype map;
map.add(4, 2, 15);
map.add(4, 1, 5);
map.add(4, 1, 10);
map.add(4, 5, 10);
map.add(4, 7, 12);


// Create the template genome using the phenotype map we just made.

GABin2DecGenome genome(map, objective);

// Now create the GA using the genome and run it.  We'll use sigma truncation
// scaling so that we can handle negative objective scores.

GASimpleGA ga(genome);
GASigmaTruncationScaling scaling;
ga.populationSize(popsize);
ga.pConvergence(pconv);
ga.nConvergence(nconv);
ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);
ga.pMutation(pmut);
ga.pCrossover(pcross);
ga.scaling(scaling);
ga.scoreFilename("C:/galib/problems/logistic/boglog.dat");
ga.scoreFrequency(1);
ga.flushFrequency(50);
ga.evolve(seed);

// Dump the results of the GA to the screen.
double ARRIVE, DISC1, DISC2, DISC3;
int FAIL, ARRIVEa, DISC1a, DISC2a, DISC3a;
genome = ga.statistics().bestIndividual();
```

```
        FAIL = genome.phenotype(0);
        ARRIVE = genome.phenotype(1);
        DISC1 = genome.phenotype(2);
        DISC2 = genome.phenotype(3);
        DISC3 = genome.phenotype(4);

        // The following code, converts the real number to only 2 decimal points
        ARRIVEa = (100 * ARRIVE) + 0.5*(ARRIVE/fabs(ARRIVE));
        DISC1a = (100 * DISC1) + 0.5*(DISC1/fabs(DISC1));
        DISC2a = (100 * DISC2) + 0.5*(DISC2/fabs(DISC2));
        DISC3a = (100 * DISC3) + 0.5*(DISC3/fabs(DISC3));
        ARRIVE = ARRIVEa/100.0;
        DISC1 = DISC1a/100.0;
        DISC2 = DISC2a/100.0;
        DISC3 = DISC3a/100.0;

        cout << "the ga found an optimum at the point (";
        cout << FAIL << ", " << ARRIVE << ", " << DISC1 << ", " << DISC2 << ", " <<
                DISC3 << ")\n\n";

        cout << "best of generation data are in '" << ga.scoreFilename() << "'\n";
        cout << "The GA found: \n" <<ga.statistics().bestIndividual() <<"\n";

        time_t end;
        time(&end);
        double timedif = end - start;
        clock_t clock(void);

        cout << "time difference = " << timedif << " seconds" << endl;
        cout << "time difference = " << timedif/60 << " minutes" << endl;
        cout << "the number of calls to the simulation = " << calls << endl;
        cout << "then number of generations was = " << ga.generation() << endl;

        outfil9 <<      "time difference = " << timedif << " seconds" << endl << "time
            difference = " << timedif/60 << " minutes" << endl << "the number of calls to
            the simulation = " << calls << endl << "then number of generations was = "
            << ga.generation() << endl;
        outfil9.close();
        return 0;
}



// This objective function tries to minimize the value of the function
//
//                          Z = P1*(d1+)/T1+ P2*(d2+)/T1
```

```
//                          Goal 1: Minimize Total Costs (TC).
//                          Goal 2: Maximize Average Part Utilization (UTIL_AVG)
//                                   or min -UTIL_AVG

float
objective(GAGenome & c)
{
        GABin2DecGenome & genome = (GABin2DecGenome &)c;

        ifstream infile1("C:/galib/problems/logistic/log_out.dat", ios::in);
                    // associate Core_GA_output.dat with infile1
                    // this file will hold the output from AweSim needed to guide the
                       optimization program
                    // specifically, the total sales (TS), the total orders (TO), the total
                       reviews (TR), and the average inventory(AvgInv)
        ofstream outfile("C:/galib/problems/logistic/log_in.dat", ios::out);
                    // associate GA_data.dat with outfile
                    // this file will hold the input data for Awesim
                    // specifically, the number to order (Q), the re-order point (R), and the
                       time between reviews (TBREV)
        ofstream outfile1("C:/galib/problems/logistic/logout1.dat", ios::app);
                    // associate ga1_out1.dat with outfile1, set it up to add to the file each
                       time through
                    // this file will store all visited points, and the corresponding Z and
                       goal values
        ofstream outfile2("C:/galib/problems/logistic/logout2.dat", ios::app);
                    // associate ga1_out2.dat with outfile2, set it up to add to the file each
                       time through
                    // this file will store all visited points, want to use it to compare if point
                       previously visited
        ofstream outfile4("C:/galib/problems/logistic/logout4.dat", ios::app);
                    // associate ga1_out2.dat with outfile4, set it up to add to the file each
                       time through
                    // trying to see if this will work
        ofstream outfile5("C:/galib/problems/logistic/logout5.dat", ios::out);
                    // associate ga1_out5.dat with outfile5, set it up to add to the file each
                       time through
                    // trying to see if this will work
        ofstream outfile6("C:/galib/problems/logistic/logout6.dat", ios::app);
                    // associate ga1_out6.dat with outfile6, set it up to add to the file each
                       time through
                    // trying to see if this will work
        ofstream outfile7("C:/galib/problems/logistic/logout7.dat", ios::app);
                    // associate ga1_out7.dat with outfile7, set it up to add to the file each
                       time through
```

```
        // want to store visited points, and simulation output data to check
           calculations
ofstream outfile8("C:/galib/problems/logistic/logout8.dat", ios::app);
        // associate ga1_out8.dat with outfile8, set it up to add to the file each
           time through
        // want to store visited points, and simulation output data to check
           calculations

outfile1.setf(ios::fixed);
outfile6.setf(ios::fixed);
outfile7.setf(ios::fixed);
outfile8.setf(ios::fixed);


double Z, goal[2], targets[2];
int g, count = 0, i = 0;
double ARRIVE, DISC1, DISC2, DISC3;
int FAIL, ARRIVEa, DISC1a, DISC2a, DISC3a;
double P;
double number,epsilon;
string test, hold;
double ORD1_AVG, ORD2_AVG, ORD3_AVG, OUT1_AVG, OUT1_STD,
   OUT2_AVG, OUT2_STD, OUT3_AVG, OUT3_STD, UTIL_AVG,
   UTIL_STD;
double timedif;


// associate Q, R, and TBREV with their corresponding genome.phenotypes
FAIL = genome.phenotype(0);
ARRIVE = genome.phenotype(1);
DISC1 = genome.phenotype(2);
DISC2 = genome.phenotype(3);
DISC3 = genome.phenotype(4);

// The following code, converts the real number to only 2 decimal points
ARRIVEa = (100 * ARRIVE) + 0.5*(ARRIVE/fabs(ARRIVE));
DISC1a = (100 * DISC1) + 0.5*(DISC1/fabs(DISC1));
DISC2a = (100 * DISC2) + 0.5*(DISC2/fabs(DISC2));
DISC3a = (100 * DISC3) + 0.5*(DISC3/fabs(DISC3));
ARRIVE = ARRIVEa/100.0;
DISC1 = DISC1a/100.0;
DISC2 = DISC2a/100.0;
DISC3 = DISC3a/100.0;

// define the target values for each goal
targets[0] = -409.8;
```

```
targets[1] = 0.85;

// write the input data for AweSim
outfile << FAIL << '\t' << ARRIVE << '\t' << DISC1 << '\t' << DISC2 << '\t' <<
        DISC3 << endl;

outfile5 << FAIL << ARRIVEa << DISC1a << DISC2a << DISC3a << endl;
outfile.close();
outfile5.close();

ifstream infile2("C:/galib/problems/logistic/logout2.dat", ios::in);
        // associate ga1_out2.dat with infile2
        // note that this is also the file associated with outfile2
        // we use this to see if we have previously visited a point or not
ifstream infile5("C:/galib/problems/logistic/logout5.dat", ios::in);
        // associate ga1_out5.dat with infile5
        // note that this is also the file associated with outfile2
        // we use this to determine the current point

infile5 >> test;

infile2 >> hold >> number;
count1++;
cout << "count1 = " << count1 << endl;

while (!infile2.fail() && i < 10000)
{
    i++;
    count++;
    if (hold == test) {
        Z = number;
        i = 12000;
    }
    else {
        infile2 >> hold >> number;
    }
}

infile2.close();
infile5.close();

outfile4 << hold << endl;

// only call AweSim if the point has not previously been visited
if (i != 12000)
{
```

```
// run AweSim and generate the output
cout << "(" << FAIL << ", " << ARRIVE << ", " << DISC1 << ", " <<
        DISC2 << ", " << DISC3 <<")\n\n";  // Write out current point being
        simulated
system ("callLOG.bat");  // this calls a file that calls the simulation model
calls = calls + 1;
cout << "calls = " << calls << endl;


// collect the output from the AweSim runs
infile1 >> ORD1_AVG >> ORD2_AVG >> ORD3_AVG >>
           OUT1_AVG >> OUT2_AVG >> OUT3_AVG >> OUT1_STD
           >> OUT2_STD >> OUT3_STD >> UTIL_AVG >> UTIL_STD;

// For goal 0, we want to minimize the total costs (TC)
// For goal 1, we want to maximize average part utilization (UTIL_AVG)

goal[0] = -1 * ((ORD1_AVG * 25) + (ORD2_AVG * 20) + (ORD3_AVG
            * 30) + (3 * OUT1_AVG) + (2 * OUT2_AVG) + (4 *
            OUT3_AVG));
goal[1] = (UTIL_AVG/100);

// loop to calculate Z
Z = 0;
epsilon = 0.00005;
g = 1;                          // where g is the number of goals - 1 (to
                                   work with C++ notation)
for (int j = 0; j<=g; j++)
{
        P = pow(100,(g-j));
        if (goal[j] >= (targets[j] - epsilon))
        {
                goal[j] = targets[j];
        }

        // we want to calculate the normalized objective function
        // following the equation, Z = sum (Pi*di/Ti)
        Z = Z + ((goal[j] - targets[j])/fabs(targets[j]))*P;
}
Z = 1*Z;
time_t temp;
time(&temp);
timedif = temp - begin;
clock_t clock(void);
outfile8 << i << '\t' << calls << '\t' << timedif << '\t' << FAIL << '\t' <<
    ARRIVE << '\t' << DISC1 << '\t' << DISC2 << '\t' << DISC3<< '\t' <<
```

```
            Z << '\t' << goal[0] << '\t' << goal[1] << '\t' << ORD1_AVG << '\t' <<
            ORD2_AVG << '\t' << ORD3_AVG << '\t' << '\t' << OUT1_AVG <<
            '\t' << OUT2_AVG << '\t' << OUT3_AVG << '\t' << OUT1_STD <<
            '\t' << OUT2_STD << '\t' << OUT3_STD << '\t' << UTIL_AVG << '\t'
            << UTIL_STD << '\n';

        outfile6 << FAIL << ARRIVEa << DISC1a << DISC2a << DISC3a << '\t'
            << FAIL << '\t' << ARRIVE << '\t' << DISC1 << '\t' << DISC2 << '\t'
            << DISC3<< '\t' << Z << '\t' << "count=" << count << '\t' << "i=" << i
            << '\n';
    }

    // write out the results
    outfile1 << FAIL << '\t' << ARRIVE << '\t' << DISC1 << '\t' << DISC2 << '\t' <<
        DISC3<< '\t' << Z << '\t' << goal[0] << '\t' << goal[1] << '\t' << '\t' <<
        "count=" << count << '\t' << "i=" << i << '\n';
    outfile2 << FAIL << ARRIVEa << DISC1a << DISC2a << DISC3a << '\t' << Z
        << '\n';
    outfile7 << i << '\t' << calls << '\t' << timedif << '\t' << FAIL << '\t' << ARRIVE
        << '\t' << DISC1 << '\t' << DISC2 << '\t' << DISC3 << '\t' << Z << '\t' <<
        goal[0] << '\t' << goal[1] << '\t' << ORD1_AVG << '\t' << ORD2_AVG << '\t'
        << ORD3_AVG << '\t' << '\t' << OUT1_AVG << '\t' << OUT2_AVG << '\t'
        << OUT3_AVG << '\t' << OUT1_STD << '\t' << OUT2_STD << '\t' <<
        OUT3_STD << '\t' << UTIL_AVG << '\t' << UTIL_STD << '\n';

    infile1.close();
    outfile1.close();
    outfile2.close();
    outfile4.close();
    outfile6.close();
    outfile7.close();
    outfile8.close();

    return Z;

}
```

Here is the file "callLOG.bat";

```
cd\
cd projects\dissert
execute logist2
cd\
cd galib\problems\logistic
```

PERT Domain

```
/* ---------------------------------------------------------------------------
DESCRIPTION: Genetic Algorithm Simulation Optimization (GA-SO) Method
 This problem examines 3 Variables, 2 Goals, using genetic algorithms
 to solve a PERT domain test bed problem.
--------------------------------------------------------------------------- */
#include <iostream>
#include <fstream>     // needed for direct file manipulation
#include <string>
#include <stdlib.h>
#include <time.h>
#include <cmath>

#include <stdio.h>
#include <ga/ga.h>
#include <ga/std_stream.h>

#define cout STD_COUT
using namespace std;

float objective(GAGenome &);
int calls = 0;
int count1 = 0;
double begin;

int
main(int argc, char **argv)
{

	time_t start;
	time(&start);
	begin = start;

	// we initially want to create a blank file called sampleout.dat, sampleout1.dat, and
	   sampleout3.dat
	// we do this in case the file already exists, so that the data is not carried over
	   between experiments
	ofstream outfil1("C:/galib/problems/pert/pertout1.dat", ios::out);
	ofstream outfil2("C:/galib/problems/pert/pertout2.dat", ios::out);
	ofstream outfil3("C:/galib/problems/pert/pertout3.dat", ios::out);
	ofstream outfil4("C:/galib/problems/pert/pertout4.dat", ios::out);
	ofstream outfil6("C:/galib/problems/pert/pertout6.dat", ios::out);
	ofstream outfil7("C:/galib/problems/pert/pertout7.dat", ios::out);
	ofstream outfil8("C:/galib/problems/pert/pertout8.dat", ios::out);
	ofstream outfil9("C:/galib/problems/pert/pert_results.dat", ios:: out);
```

```
// here we seed an initial value of 0's into sampleou2.dat, so our loop will have
   data to start searching the first time through.
outfil2 << 0.0 << '\t' << 0.0 << endl;
outfil7 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "LOW3" << '\t'
     << "LOW4" << '\t' << "LOW6" << '\t' << "Z" << '\t' << "Goal1" << '\t' <<
     "Goal2" << '\t'<< "CRIT1" << '\t' << "CRIT2" << '\t' << "CRIT3" << '\t'<<
     "CRIT4" << '\t' << "CRIT5" << '\t' << "CRIT6" << '\t'<< "CRIT7" << '\t' <<
     "CRIT8" << '\t' << "CRIT9" << '\t' << "PT_AVG" << '\t' << "PT_STD" <<
     '\n';
outfil8 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "LOW3" << '\t'
     << "LOW4" << '\t' << "LOW6" << '\t' << "Z" << '\t' << "Goal1" << '\t' <<
     "Goal2" << '\t'<< "CRIT1" << '\t' << "CRIT2" << '\t' << "CRIT3" << '\t'<<
     "CRIT4" << '\t' << "CRIT5" << '\t' << "CRIT6" << '\t'<< "CRIT7" << '\t' <<
     "CRIT8" << '\t' << "CRIT9" << '\t' << "PT_AVG" << '\t' << "PT_STD" <<
     '\n';

outfil1.close();
outfil2.close();
outfil3.close();
outfil4.close();
outfil6.close();
outfil7.close();
outfil8.close();

cout << "Test Problem\n\n";
cout << "This program finds the minimum cost and the minimum project
        completion time\n";
cout << "  TC\n";
cout << "with the constraints\n";
cout << "     5 <= LOW3 <= 12\n";
cout << "     1 <= LOW4 <= 8\n";
cout << "     1 <= LOW6 <= 8\n";
cout << "\n\n"; cout.flush();

// See if we've been given a seed to use (for testing purposes).  When you
// specify a random seed, the evolution will be exactly the same each time
// you use that seed number.

unsigned int seed = 0;
for(int i=1; i<argc; i++) {
        if(strcmp(argv[i++],"seed") == 0) {
                seed = atoi(argv[i]);
        }
}
```

```
// Declare variables for the GA parameters and set them to some default values.

int popsize  = 30;
int ngen     = 100;
float pmut   = 0.01;
float pcross = 0.6;
float pconv = 0.99;     //Convergence threshold
float nconv = 100;      //Number of generations to examine

// Create a phenotype for with the appropriate range for each variable.

GABin2DecPhenotype map;
map.add(4, 5, 12);
map.add(4, 1, 8);
map.add(4, 1, 8);


// Create the template genome using the phenotype map we just made.

GABin2DecGenome genome(map, objective);

// Now create the GA using the genome and run it.  We'll use sigma truncation
// scaling so that we can handle negative objective scores.

GASimpleGA ga(genome);
GASigmaTruncationScaling scaling;
ga.populationSize(popsize);
ga.pConvergence(pconv);
ga.nConvergence(nconv);
ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);
ga.pMutation(pmut);
ga.pCrossover(pcross);
ga.scaling(scaling);
ga.scoreFilename("C:/galib/problems/pert/bogpert.dat");
ga.scoreFrequency(1);
ga.flushFrequency(50);
ga.evolve(seed);

// Dump the results of the GA to the screen.
double LOW3, LOW4, LOW6;
int LOW3a, LOW4a, LOW6a;
genome = ga.statistics().bestIndividual();
LOW3 = genome.phenotype(0);
LOW4 = genome.phenotype(1);
LOW6 = genome.phenotype(2);
```

```
        // The following code, converts the real number to only 2 decimal points
        LOW3a = (100 * LOW3) + 0.5*(LOW3/fabs(LOW3));
        LOW4a = (100 * LOW4) + 0.5*(LOW4/fabs(LOW4));
        LOW6a = (100 * LOW6) + 0.5*(LOW6/fabs(LOW6));
        LOW3 = LOW3a/100.0;
        LOW4 = LOW4a/100.0;
        LOW6 = LOW6a/100.0;

        cout << "the ga found an optimum at the point (";
        cout << LOW3 << ", " << LOW4 << ", " << LOW6 << ")\n\n";

        cout << "best of generation data are in '" << ga.scoreFilename() << "'\n";
        cout << "The GA found: \n" <<ga.statistics().bestIndividual() <<"\n";

        time_t end;
        time(&end);
        double timedif = end - start;
        clock_t clock(void);

        cout << "time difference = " << timedif << " seconds" << endl;
        cout << "time difference = " << timedif/60 << " minutes" << endl;
        cout << "the number of calls to the simulation = " << calls << endl;
        cout << "then number of generations was = " << ga.generation() << endl;

        outfil9 <<      "time difference = " << timedif << " seconds" << endl << "time
            difference = " << timedif/60 << " minutes" << endl << "the number of calls to
            the simulation = " << calls << endl << "then number of generations was = "
            << ga.generation() << endl;
        outfil9.close();
        return 0;
}



// This objective function tries to minimize the value of the function
//
//                      Z = P1*(d1+)/T1 + P2*(d2+)/T2
//                      Goal 1: Minimize Total Cost (TC)
//                      Goal 2: Minimize Project Completion Time (PT_AVG)
//

float
objective(GAGenome & c)
{
        GABin2DecGenome & genome = (GABin2DecGenome &)c;
```

```
ifstream infile("C:/galib/problems/pert/pertcrit.dat", ios::in);
ifstream infile1("C:/galib/problems/pert/pertout.dat", ios::in);
        // associate Core_GA_output.dat with infile1
        // this file will hold the output from AweSim needed to guide the
           optimization program
        // specifically, the total sales (TS), the total orders (TO), the total
           reviews (TR), and the average inventory(AvgInv)
ofstream outfile("C:/galib/problems/pert/pertin.dat", ios::out);
        // associate GA_data.dat with outfile
        // this file will hold the input data for Awesim
        // specifically, the number to order (Q), the re-order point (R), and the
           time between reviews (TBREV)
ofstream outfile1("C:/galib/problems/pert/pertout1.dat", ios::app);
        // associate ga1_out1.dat with outfile1, set it up to add to the file each
           time through
        // this file will store all visited points, and the corresponding Z and
           goal values
ofstream outfile2("C:/galib/problems/pert/pertout2.dat", ios::app);
        // associate ga1_out2.dat with outfile2, set it up to add to the file each
           time through
        // this file will store all visited points, want to use it to compare if point
           previously visited
ofstream outfile4("C:/galib/problems/pert/pertout4.dat", ios::app);
        // associate ga1_out2.dat with outfile4, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile5("C:/galib/problems/pert/pertout5.dat", ios::out);
        // associate ga1_out5.dat with outfile5, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile6("C:/galib/problems/pert/pertout6.dat", ios::app);
        // associate ga1_out6.dat with outfile6, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile7("C:/galib/problems/pert/pertout7.dat", ios::app);
        // associate ga1_out7.dat with outfile7, set it up to add to the file each
           time through
        // want to store visited points, and simulation output data to check
           calculations
ofstream outfile8("C:/galib/problems/pert/pertout8.dat", ios::app);
        // associate ga1_out8.dat with outfile8, set it up to add to the file each
           time through
        // want to store visited points, and simulation output data to check
           calculations

outfile1.setf(ios::fixed);
```

```
outfile6.setf(ios::fixed);
outfile7.setf(ios::fixed);
outfile8.setf(ios::fixed);

double Z, goal[2], targets[2];
int g, count = 0, i = 0;
double LOW3, LOW4, LOW6;
int LOW3a, LOW4a, LOW6a;
double P;
double number, epsilon;
string hold, test;
double CRIT1, CRIT2, CRIT3, CRIT4, CRIT5, CRIT6, CRIT7, CRIT8, CRIT9,
    PT_AVG, PT_STD;
double NODE2AVG, NODE2STD, NODE3AVG, NODE3STD, NODE4AVG,
    NODE4STD, NODE5AVG, NODE5STD;

double timedif;

// associate Q, R, and TBREV with their corresponding genome.phenotypes
LOW3 = genome.phenotype(0);
LOW4 = genome.phenotype(1);
LOW6 = genome.phenotype(2);

// The following code, converts the real number to only 2 decimal points
LOW3a = (100 * LOW3) + 0.5*(LOW3/fabs(LOW3));
LOW4a = (100 * LOW4) + 0.5*(LOW4/fabs(LOW4));
LOW6a = (100 * LOW6) + 0.5*(LOW6/fabs(LOW6));
LOW3 = LOW3a/100.0;
LOW4 = LOW4a/100.0;
LOW6 = LOW6a/100.0;

// define the target values for each goal
targets[0] = -21.76;
targets[1] = -18.04;

// write the input data for AweSim
outfile << LOW3 << '\t' << LOW4 << '\t' << LOW6 << endl;

outfile5 << LOW3a << LOW4a << LOW6a << endl;
outfile.close();
outfile5.close();

ifstream infile2("C:/galib/problems/pert/pertout2.dat", ios::in);
// associate ga1_out2.dat with infile2
// note that this is also the file associated with outfile2
// we use this to see if we have previously visited a point or not
```

```
ifstream infile5("C:/galib/problems/pert/pertout5.dat", ios::in);
// associate ga1_out5.dat with infile5
// note that this is also the file associated with outfile2
// we use this to determine the current point

infile5 >> test;

infile2 >> hold >> number;
count1++;
cout << "count1 = " << count1 << endl;

while (!infile2.fail() && i < 10000)
{
        i++;
        count++;
        if (hold == test) {
                Z = number;
                i = 12000;
        }
        else {
                infile2 >> hold >> number;
        }
}

infile2.close();
infile5.close();

outfile4 << hold << endl;

// only call AweSim if the point has not previously been visited
if (i != 12000)
{
        // run AweSim and generate the output
        cout << "(" << LOW3 << ", " << LOW4 << ", " << LOW6 <<")\n\n";
                    // Write out current point being simulated
        system ("callPERT.bat");  // this calls a file that calls the simulation model
        calls = calls + 1;
        cout << "calls = " << calls << endl;

        // collect the output from the AweSim runs
        infile1   >>   NODE2AVG   >>   NODE2STD   >>   NODE3AVG   >>
                 NODE3STD >> NODE4AVG >> NODE4STD >> NODE5AVG
                 >> NODE5STD >> PT_AVG >> PT_STD;
        infile >> CRIT1 >> CRIT2 >> CRIT3 >> CRIT4 >> CRIT5 >> CRIT6 >>
                 CRIT7 >> CRIT8 >> CRIT9;
```

```
        // For goal 0, we want to minimize total cost (TC)
        // For goal 1, we want to minimize the project completion time (PT_AVG)

        goal[0] = -1 * ((CRIT1 * 10) + (CRIT2 * 6) + (CRIT3 * 2) + (CRIT4 * 7)
                  + (CRIT5 * 8) + (CRIT6 * 3) + (CRIT7 * 4) + (CRIT8 * 5) +
                  (CRIT9 * 8));
        goal[1] = -1 * PT_AVG;

        // loop to calculate Z
        Z = 0;
        epsilon = 0.00005;
        g = 1;                          // where g is the number of goals - 1 (to
                                           work with C++ notation)
        for (int j = 0; j<=g; j++)
        {
                P = pow(100,(g-j));
                if (goal[j] >= (targets[j] - epsilon))
                {
                        goal[j] = targets[j];
                }

                // we want to calculate the normalized objective function
                // following the equation, Z = sum (Pi*di/Ti)
                Z = Z + ((goal[j] - targets[j])/fabs(targets[j]))*P;
        }
        time_t temp;
        time(&temp);
        timedif = temp - begin;
        clock_t clock(void);


        outfile8 << i << '\t' << calls << '\t' << timedif << '\t' << LOW3 << '\t' <<
            LOW4 << '\t' << LOW6 << '\t' << Z << '\t' << goal[0] << '\t' <<
            goal[1] << '\t' << CRIT1 << '\t' << CRIT2 << '\t' << CRIT3 << '\t' <<
            CRIT4 << '\t' << CRIT5 << '\t' << CRIT6 << '\t' << CRIT7 << '\t' <<
            CRIT8 << '\t' << CRIT9 << '\t' << PT_AVG << '\t' << PT_STD <<
            '\n';

        outfile6 << LOW3 << LOW4 << LOW6 << '\t' << LOW3 << '\t' <<
            LOW4 << '\t' << LOW6 << '\t' << Z << '\t' << "count=" << count <<
            '\t' << "i=" << i << '\n';
}

// write out the results
```

```
        outfile1 << LOW3 << '\t' << LOW4 << '\t' << LOW6 << '\t' << Z << '\t' <<
            goal[0] << '\t' << goal[1] << '\t' << "count=" << count << '\t' << "i=" << i <<
            '\n';
        outfile2 << LOW3a << LOW4a << LOW6a << '\t' << Z << '\n';
        outfile7 << i << '\t' << calls << '\t' << timedif << '\t' << LOW3 << '\t' << LOW4
            << '\t' << LOW6 << '\t' << Z << '\t' << goal[0] << '\t' << goal[1] << '\t' <<
            CRIT1 << '\t' << CRIT2 << '\t' << CRIT3 << '\t' << CRIT4 << '\t' << CRIT5
            << '\t' << CRIT6 << '\t' << CRIT7 << '\t' << CRIT8 << '\t' << CRIT9 << '\t' <<
            PT_AVG << '\t' << PT_STD << '\n';


        infile1.close();
        outfile1.close();
        outfile2.close();
        outfile4.close();
        outfile6.close();
        outfile7.close();
        outfile8.close();

        return Z;

}
```

Here is the file "callPERT.bat";

```
cd\
cd projects\dissert
execute pert
cd\
cd galib\problems\pert
```

Production Domain

```
/* --------------------------------------------------------------------------
DESCRIPTION: Genetic Algorithm Simulation Optimization (GA-SO) Method
 This problem examines 6 Variables, 2 Goals, using genetic algorithms
 to solve a production domain test bed problem.
-------------------------------------------------------------------------- */
#include <iostream>
#include <fstream>     // needed for direct file manipulation
#include <string>
#include <stdlib.h>
#include <time.h>
#include <cmath>

#include <stdio.h>
#include <ga/ga.h>
#include <ga/std_stream.h>

#define cout STD_COUT
using namespace std;

float objective(GAGenome &);
int calls = 0;
int count1 = 0;
double begin;

int
main(int argc, char **argv)
{

        time_t start;
        time(&start);
        begin = start;

        // we initially want to create a blank file called sampleout.dat, sampleout1.dat, and
           sampleout3.dat
        // we do this in case the file already exists, so that the data is not carried over
           between experiments
        ofstream outfil1("C:/galib/problems/product/prodout1.dat", ios::out);
        ofstream outfil2("C:/galib/problems/product/prodout2.dat", ios::out);
        ofstream outfil3("C:/galib/problems/product/prodout3.dat", ios::out);
        ofstream outfil4("C:/galib/problems/product/prodout4.dat", ios::out);
        ofstream outfil6("C:/galib/problems/product/prodout6.dat", ios::out);
        ofstream outfil7("C:/galib/problems/product/prodout7.dat", ios::out);
        ofstream outfil8("C:/galib/problems/product/prodout8.dat", ios::out);
        ofstream outfil9("C:/galib/problems/product/prod_results.dat", ios:: out);
```

```
// here we seed an initial value of 0's into sampleou2.dat, so our loop will have
   data to start searching the first time through.
outfil2 << 0.0 << '\t' << 0.0 << endl;
outfil7 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "LOAD" << '\t'
    << "INSPECT" << '\t' << "UNLOAD" << '\t' << "TRANSPORT1" << '\t' <<
    "TRANSPORT2" << '\t' << "TRANSPORT3" << '\t' << "Z" << '\t' <<
    "Goal1" << '\t' << "Goal2" << '\t'<< "TPUT" << '\t' << "TIS_AVG" << '\t'<<
    "TIS_STD" << '\t' << "PU_AVG" << '\t' << "PU_STD" << '\t'<< "CU_AVG"
    << '\t' << "CU_STD" << '\n';
outfil8 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "LOAD" << '\t'
    << "INSPECT" << '\t' << "UNLOAD" << '\t' << "TRANSPORT1" << '\t' <<
    "TRANSPORT2" << '\t' << "TRANSPORT3" << '\t' << "Z" << '\t' <<
    "Goal1" << '\t' << "Goal2" << '\t'<< "TPUT" << '\t' << "TIS_AVG" << '\t'<<
    "TIS_STD" << '\t' << "PU_AVG" << '\t' << "PU_STD" << '\t'<< "CU_AVG"
    << '\t' << "CU_STD" << '\n';

outfil1.close();
outfil2.close();
outfil3.close();
outfil4.close();
outfil6.close();
outfil7.close();
outfil8.close();

cout << "Test Problem\n\n";
cout << "This program finds the minimum cost and the minimum project
        completion time\n";
cout << "  TC\n";
cout << "with the constraints\n";
cout << "     5 <= LOW3 <= 20\n";
cout << "     1 <= LOW4 <= 8\n";
cout << "     1 <= LOW6 <= 8\n";
cout << "\n\n"; cout.flush();

// See if we've been given a seed to use (for testing purposes).  When you
// specify a random seed, the evolution will be exactly the same each time
// you use that seed number.

unsigned int seed = 0;
for(int i=1; i<argc; i++) {
        if(strcmp(argv[i++],"seed") == 0) {
                seed = atoi(argv[i]);
        }
}
```

```
// Declare variables for the GA parameters and set them to some default values.

int popsize  = 30;
int ngen     = 50;
float pmut   = 0.01;
float pcross = 0.6;
float pconv = 0.99;     //Convergence threshold
float nconv = 100;      //Number of generations to examine

// Create a phenotype for with the appropriate range for each variable.

GABin2DecPhenotype map;
map.add(4, 5, 10);
map.add(4, 10, 20);
map.add(4, 5, 15);
map.add(4, 3, 7);
map.add(4, 1, 4);
map.add(4, 2, 6);

// Create the template genome using the phenotype map we just made.

GABin2DecGenome genome(map, objective);

// Now create the GA using the genome and run it.  We'll use sigma truncation
// scaling so that we can handle negative objective scores.

GASimpleGA ga(genome);
GASigmaTruncationScaling scaling;
ga.populationSize(popsize);
ga.pConvergence(pconv);
ga.nConvergence(nconv);
ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);
ga.pMutation(pmut);
ga.pCrossover(pcross);
ga.scaling(scaling);
ga.scoreFilename("C:/galib/problems/product/bogprod.dat");
ga.scoreFrequency(1);
ga.flushFrequency(50);
ga.evolve(seed);

// Dump the results of the GA to the screen.
double LOAD, INSPECT, UNLOAD, TRANSPORT1, TRANSPORT2,
    TRANSPORT3;
int LOADa, INSPECTa, UNLOADa, TRANSPORT1a, TRANSPORT2a,
    TRANSPORT3a;
genome = ga.statistics().bestIndividual();
```

```
LOAD = genome.phenotype(0);
INSPECT = genome.phenotype(1);
UNLOAD = genome.phenotype(2);
TRANSPORT1 = genome.phenotype(3);
TRANSPORT2 = genome.phenotype(4);
TRANSPORT3 = genome.phenotype(5);

// The following code, converts the real number to only 2 decimal points
LOADa = (100 * LOAD) + 0.5*(LOAD/fabs(LOAD));
INSPECTa = (100 * INSPECT) + 0.5*(INSPECT/fabs(INSPECT));
UNLOADa = (100 * UNLOAD) + 0.5*(UNLOAD/fabs(UNLOAD));
TRANSPORT1a = (100 * TRANSPORT1) +
                    0.5*(TRANSPORT1/fabs(TRANSPORT1));
TRANSPORT2a = (100 * TRANSPORT2) +
                    0.5*(TRANSPORT2/fabs(TRANSPORT2));
TRANSPORT3a = (100 * TRANSPORT3) +
                    0.5*(TRANSPORT3/fabs(TRANSPORT3));
LOAD = LOADa/100.0;
INSPECT = INSPECTa/100.0;
UNLOAD = UNLOADa/100.0;
TRANSPORT1 = TRANSPORT1a/100.0;
TRANSPORT2 = TRANSPORT2a/100.0;
TRANSPORT3 = TRANSPORT3a/100.0;

cout << "the ga found an optimum at the point (";
cout << LOAD << ", " << INSPECT << ", " << UNLOAD << ", " <<
      TRANSPORT1 << ", " << TRANSPORT2 << ", " << TRANSPORT3 <<
      ")\n\n";

cout << "best of generation data are in '" << ga.scoreFilename() << "'\n";
cout << "The GA found: \n" <<ga.statistics().bestIndividual() <<"\n";

time_t end;
time(&end);
double timedif = end - start;
clock_t clock(void);

cout << "time difference = " << timedif << " seconds" << endl;
cout << "time difference = " << timedif/60 << " minutes" << endl;
cout << "the number of calls to the simulation = " << calls << endl;
cout << "then number of generations was = " << ga.generation() << endl;

outfil9 <<    "time difference = " << timedif << " seconds" << endl << "time
    difference = " << timedif/60 << " minutes" << endl << "the number of calls to
    the simulation = " << calls << endl << "then number of generations was = "
    << ga.generation() << endl;
```

```
        outfil9.close();
        return 0;
}



// This objective function tries to minimize the value of the function
//
//                      Z = P1*(d1+)/T1 + P2*(d2+)/T2
//                      Goal 1: Maximize Total Profit (TP)
//                              or min -TP
//                      Goal 2: Maximize Pallet and Cart Utilization
//                              or min -utilization

float
objective(GAGenome & c)
{
        GABin2DecGenome & genome = (GABin2DecGenome &)c;

        ifstream infile1("C:/galib/problems/product/prod_out.dat", ios::in);
                // associate Core_GA_output.dat with infile1
                // this file will hold the output from AweSim needed to guide the
                   optimization program
                // specifically, the total sales (TS), the total orders (TO), the total
                   reviews (TR), and the average inventory(AvgInv)
        ofstream outfile("C:/galib/problems/product/prod_in.dat", ios::out);
                // associate GA_data.dat with outfile
                // this file will hold the input data for Awesim
                // specifically, the number to order (Q), the re-order point (R), and the
                   time between reviews (TBREV)
        ofstream outfile1("C:/galib/problems/product/prodout1.dat", ios::app);
                // associate ga1_out1.dat with outfile1, set it up to add to the file each
                   time through
                // this file will store all visited points, and the corresponding Z and
                   goal values
        ofstream outfile2("C:/galib/problems/product/prodout2.dat", ios::app);
                // associate ga1_out2.dat with outfile2, set it up to add to the file each
                   time through
                // this file will store all visited points, want to use it to compare if point
                   previously visited
        ofstream outfile4("C:/galib/problems/product/prodout4.dat", ios::app);
                // associate ga1_out2.dat with outfile4, set it up to add to the file each
                   time through
                // trying to see if this will work
        ofstream outfile5("C:/galib/problems/product/prodout5.dat", ios::out);
```

```
                // associate ga1_out5.dat with outfile5, set it up to add to the file each
                   time through
                // trying to see if this will work
ofstream outfile6("C:/galib/problems/product/prodout6.dat", ios::app);
                // associate ga1_out6.dat with outfile6, set it up to add to the file each
                   time through
                // trying to see if this will work
ofstream outfile7("C:/galib/problems/product/prodout7.dat", ios::app);
                // associate ga1_out7.dat with outfile7, set it up to add to the file each
                   time through
                // want to store visited points, and simulation output data to check
                   calculations
ofstream outfile8("C:/galib/problems/product/prodout8.dat", ios::app);
                // associate ga1_out8.dat with outfile8, set it up to add to the file each
                   time through
                // want to store visited points, and simulation output data to check
                   calculations

outfile1.setf(ios::fixed);
outfile6.setf(ios::fixed);
outfile7.setf(ios::fixed);
outfile8.setf(ios::fixed);


double Z, goal[2], targets[2];
int g, count = 0, i = 0;
double LOAD, INSPECT, UNLOAD, TRANSPORT1, TRANSPORT2,
    TRANSPORT3;
int LOADa, INSPECTa, UNLOADa, TRANSPORT1a, TRANSPORT2a,
    TRANSPORT3a;
double P;
double number, epsilon;
string hold, test;
double TPUT, TIS_AVG, TIS_STD, PU_AVG, PU_STD, CU_AVG, CU_STD;
double timedif;

// associate Q, R, and TBREV with their corresponding genome.phenotypes
LOAD = genome.phenotype(0);
INSPECT = genome.phenotype(1);
UNLOAD = genome.phenotype(2);
TRANSPORT1 = genome.phenotype(3);
TRANSPORT2 = genome.phenotype(4);
TRANSPORT3 = genome.phenotype(5);

// The following code, converts the real number to only 2 decimal points
LOADa = (100 * LOAD) + 0.5*(LOAD/fabs(LOAD));
```

```
INSPECTa = (100 * INSPECT) + 0.5*(INSPECT/fabs(INSPECT));
UNLOADa = (100 * UNLOAD) + 0.5*(UNLOAD/fabs(UNLOAD));
TRANSPORT1a = (100 * TRANSPORT1) +
                    0.5*(TRANSPORT1/fabs(TRANSPORT1));
TRANSPORT2a = (100 * TRANSPORT2) +
                    0.5*(TRANSPORT2/fabs(TRANSPORT2));
TRANSPORT3a = (100 * TRANSPORT3) +
                    0.5*(TRANSPORT3/fabs(TRANSPORT3));
LOAD = LOADa/100.0;
INSPECT = INSPECTa/100.0;
UNLOAD = UNLOADa/100.0;
TRANSPORT1 = TRANSPORT1a/100.0;
TRANSPORT2 = TRANSPORT2a/100.0;
TRANSPORT3 = TRANSPORT3a/100.0;

// define the target values for each goal
targets[0] = 4600.00;
targets[1] = 1.90;

// write the input data for AweSim
outfile << LOAD << '\t' << INSPECT << '\t' << UNLOAD << '\t' <<
    TRANSPORT1 << '\t' << TRANSPORT2 << '\t' << TRANSPORT3 << endl;
outfile5 << LOADa << INSPECTa << UNLOADa << TRANSPORT1a <<
    TRANSPORT2a << TRANSPORT3a << endl;
outfile.close();
outfile5.close();

ifstream infile2("C:/galib/problems/product/prodout2.dat", ios::in);
// associate ga1_out2.dat with infile2
// note that this is also the file associated with outfile2
// we use this to see if we have previously visited a point or not
ifstream infile5("C:/galib/problems/product/prodout5.dat", ios::in);
// associate ga1_out5.dat with infile5
// note that this is also the file associated with outfile2
// we use this to determine the current point

infile5 >> test;

infile2 >> hold >> number;
count1++;
cout << "count1 = " << count1 << endl;

while (!infile2.fail() && i < 10000)
{
        i++;
        count++;
```

```
		if (hold == test) {
			Z = number;
			i = 12000;
		}
		else {
			infile2 >> hold >> number;
		}
	}

	infile2.close();
	infile5.close();

	outfile4 << hold << endl;

	// only call AweSim if the point has not previously been visited
	if (i != 12000)
	{
		// run AweSim and generate the output
		cout << "(" << LOAD << ", " << INSPECT << ", " << UNLOAD << ", "
			<< TRANSPORT1 << ", " << TRANSPORT2 << ", " <<
			TRANSPORT3 <<")\n\n";   // Write out current point being
			simulated
		system ("callPROD.bat");   // this calls a file that calls the simulation
			model
		calls = calls + 1;
		cout << "calls = " << calls << endl;


		// collect the output from the AweSim runs
		infile1 >> TPUT >> TIS_AVG >> TIS_STD >> PU_AVG >> PU_STD
			>> CU_AVG >> CU_STD;

		// For goal 0, we want to maximize the total profit (TP)
		// For goal 1, we want to maximize pallet and cart utilization

		goal[0] = 100 * TPUT - (1000 * (1/TIS_AVG));
		goal[1] = (PU_AVG/6 + CU_AVG/2);

		// loop to calculate Z
		Z = 0;
		epsilon = 0.00005;
		g = 1;				// where g is the number of goals - 1 (to
						work with C++ notation)
		for (int j = 0; j<=g; j++)
		{
			P = pow(100,(g-j));
```

```
            if (goal[j] >= (targets[j] - epsilon))
            {
                goal[j] = targets[j];
            }

            // we want to calculate the normalized objective function
            // following the equation, Z = sum (Pi*di/Ti)
            Z = Z + ((goal[j] - targets[j])/fabs(targets[j]))*P;

        }
        Z = 1*Z;
        time_t temp;
        time(&temp);
        timedif = temp - begin;
        clock_t clock(void);


        outfile8 << i << '\t' << calls << '\t' << timedif << '\t' << LOAD << '\t' <<
            INSPECT << '\t' << UNLOAD << '\t' << TRANSPORT1 << '\t' <<
            TRANSPORT2 << '\t' << TRANSPORT3 << '\t' << Z << '\t' <<
            goal[0] << '\t' << goal[1] << '\t' << TPUT << '\t' << TIS_AVG << '\t'
            << TIS_STD << '\t' << PU_AVG << '\t' << PU_STD << '\t' <<
            CU_AVG << '\t' << CU_STD << '\n';

        outfile6 << LOADa << INSPECTa << UNLOADa << TRANSPORT1a
            << TRANSPORT2a << TRANSPORT3a << '\t' << LOAD << '\t' <<
            INSPECT << '\t' << UNLOAD << '\t' << TRANSPORT1 << '\t' <<
            TRANSPORT2 << '\t' << TRANSPORT3 << '\t' << Z << '\t' <<
            "count=" << count << '\t' << "i=" << i << '\n';
}

// write out the results
outfile1 << LOAD << '\t' << INSPECT << '\t' << UNLOAD << '\t' <<
    TRANSPORT1 << '\t' << TRANSPORT2 << '\t' << TRANSPORT3 << '\t' <<
    Z << '\t' << goal[0] << '\t' << goal[1] << '\t' << goal[2] << '\t' << "count=" <<
    count << '\t' << "i=" << i << '\n';
outfile2 << LOADa << INSPECTa << UNLOADa << TRANSPORT1a <<
    TRANSPORT2a << TRANSPORT3a << '\t' << Z << '\n';
outfile7 << i << '\t' << calls << '\t' << timedif << '\t' << LOAD << '\t' <<
    INSPECT << '\t' << UNLOAD << '\t' << TRANSPORT1 << '\t' <<
    TRANSPORT2 << '\t' << TRANSPORT3 << '\t' << Z << '\t' << goal[0] << '\t'
    << goal[1] << '\t' << TPUT << '\t' << TIS_AVG << '\t' << TIS_STD << '\t' <<
    PU_AVG << '\t' << PU_STD << '\t' << CU_AVG << '\t' << CU_STD << '\n';

infile1.close();
outfile1.close();
```

```
        outfile2.close();
        outfile4.close();
        outfile6.close();
        outfile7.close();
        outfile8.close();

        return Z;
}
```

Here is the file "callPROD.bat";

```
cd\
cd projects\dissert
execute product
cd\
cd galib\problems\product
```

Reliability Domain

```
/* -------------------------------------------------------------------------
DESCRIPTION: Genetic Algorithm Simulation Optimization (GA-SO) Method
 This problem examines 4 Variables, 3 Goals, using genetic algorithms
 to solve a reliability domain test bed problem.
------------------------------------------------------------------------- */
#include <iostream>
#include <fstream>     // needed for direct file manipulation
#include <string>
#include <stdlib.h>
#include <time.h>
#include <cmath>

#include <stdio.h>
#include <ga/ga.h>
#include <ga/std_stream.h>

#define cout STD_COUT
using namespace std;

float objective(GAGenome &);
int calls = 0;
int count1 = 0;
double begin;

int
main(int argc, char **argv)
{
	time_t start;
	time(&start);
	begin = start;

	// we initially want to create a blank file called sampleout.dat, sampleout1.dat, and
	   sampleout3.dat
	// we do this in case the file already exists, so that the data is not carried over
	   between experiments
	ofstream outfil1("C:/galib/problems/reliable/relioout1.dat", ios::out);
	ofstream outfil2("C:/galib/problems/reliable/reliout2.dat", ios::out);
	ofstream outfil3("C:/galib/problems/reliable/reliout3.dat", ios::out);
	ofstream outfil4("C:/galib/problems/reliable/reliout4.dat", ios::out);
	ofstream outfil6("C:/galib/problems/reliable/reliout6.dat", ios::out);
	ofstream outfil7("C:/galib/problems/reliable/reliout7.dat", ios::out);
	ofstream outfil8("C:/galib/problems/reliable/reliout8.dat", ios::out);
	ofstream outfil9("C:/galib/problems/reliable/reli_results.dat", ios:: out);
```

```
// here we seed an initial value of 0's into sampleou2.dat, so our loop will have
   data to start searching the first time through.
outfil2 << 0.0 << '\t' << 0.0 << endl;
outfil7 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "Fail1" << '\t' <<
    "Fail2" << '\t' << "Repair1" << '\t' << "Repair2" << '\t' << "Z" << '\t' <<
    "Goal1" << '\t' << "Goal2" << '\t'<< "Goal3" << '\t' << "BD1_AVG" << '\t' <<
    "BD2_AVG" << '\t'<< "SYSBD_A" << '\t' << "BD1_STD" << '\t' <<
    "BD2_STD" << '\t'<< "SYSBD_S" << '\t' << "DT_AVG" << '\t' <<
    "DT_STD" << '\n';
outfil8 << "i" << '\t' << "Calls" << '\t' << "CPU Time" << '\t' << "Fail1" << '\t' <<
    "Fail2" << '\t' << "Repair1" << '\t' << "Repair2" << '\t' << "Z" << '\t' <<
    "Goal1" << '\t' << "Goal2" << '\t'<< "Goal3" << '\t' << "BD1_AVG" << '\t' <<
    "BD2_AVG" << '\t'<< "SYSBD_A" << '\t' << "BD1_STD" << '\t' <<
    "BD2_STD" << '\t'<< "SYSBD_S" << '\t' << "DT_AVG" << '\t' <<
    "DT_STD" << '\n';

outfil1.close();
outfil2.close();
outfil3.close();
outfil4.close();
outfil6.close();
outfil7.close();
outfil8.close();

cout << "Test Problem\n\n";
cout << "This program finds the minimum cost and the minimum project
        completion time\n";
cout << "  TC\n";
cout << "with the constraints\n";
cout << "     5 <= LOW3 <= 12\n";
cout << "     1 <= LOW4 <= 8\n";
cout << "     1 <= LOW6 <= 8\n";
cout << "\n\n"; cout.flush();

// See if we've been given a seed to use (for testing purposes).  When you
// specify a random seed, the evolution will be exactly the same each time
// you use that seed number.

unsigned int seed = 0;
for(int i=1; i<argc; i++) {
        if(strcmp(argv[i++],"seed") == 0) {
                seed = atoi(argv[i]);
        }
}

// Declare variables for the GA parameters and set them to some default values.
```

```
int popsize  = 30;
int ngen     = 50;
float pmut   = 0.01;
float pcross = 0.6;
float pconv = 0.99;     //Convergence threshold
float nconv = 100;      //Number of generations to examine

// Create a phenotype for with the appropriate range for each variable.

GABin2DecPhenotype map;
map.add(4, 2, 15);
map.add(4, 1, 10);
map.add(4, 1, 10);
map.add(4, 2, 15);

// Create the template genome using the phenotype map we just made.

GABin2DecGenome genome(map, objective);

// Now create the GA using the genome and run it.  We'll use sigma truncation
// scaling so that we can handle negative objective scores.

GASimpleGA ga(genome);
GASigmaTruncationScaling scaling;
ga.populationSize(popsize);
ga.pConvergence(pconv);
ga.nConvergence(nconv);
ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);
ga.pMutation(pmut);
ga.pCrossover(pcross);
ga.scaling(scaling);
ga.scoreFilename("C:/galib/problems/reliable/bogreli.dat");
ga.scoreFrequency(1);
ga.flushFrequency(50);
ga.evolve(seed);

// Dump the results of the GA to the screen.
double Fail1, Fail2, Repair1, Repair2;
int Fail1a, Fail2a, Repair1a, Repair2a;
genome = ga.statistics().bestIndividual();
Fail1 = genome.phenotype(0);
Fail2 = genome.phenotype(1);
Repair1 = genome.phenotype(2);
Repair2 = genome.phenotype(3);

// The following code, converts the real number to only 2 decimal points
```

```
        Fail1a = (100 * Fail1) + 0.5*(Fail1/fabs(Fail1));
        Fail2a = (100 * Fail2) + 0.5*(Fail2/fabs(Fail2));
        Repair1a = (100 * Repair1) + 0.5*(Repair1/fabs(Repair1));
        Repair2a = (100 * Repair2) + 0.5*(Repair2/fabs(Repair2));
        Fail1 = Fail1a/100.0;
        Fail2 = Fail2a/100.0;
        Repair1 = Repair1a/100.0;
        Repair2 = Repair2a/100.0;

        cout << "the ga found an optimum at the point (";
        cout << Fail1 << ", " << Fail2<< ", " << Repair1 << ", " << Repair2 << ")\n\n";

        cout << "best of generation data are in '" << ga.scoreFilename() << "'\n";
        cout << "The GA found: \n" <<ga.statistics().bestIndividual() <<"\n";

        time_t end;
        time(&end);
        double timedif = end - start;
        clock_t clock(void);

        cout << "time difference = " << timedif << " seconds" << endl;
        cout << "time difference = " << timedif/60 << " minutes" << endl;
        cout << "the number of calls to the simulation = " << calls << endl;
        cout << "then number of generations was = " << ga.generation() << endl;
        cout << "count1 = " << count1 << endl;

        outfil9 <<      "time difference = " << timedif << " seconds" << endl << "time
            difference = " << timedif/60 << " minutes" << endl << "the number of calls to
            the simulation = " << calls << endl << "then number of generations was = "
            << ga.generation() << endl;
        outfil9.close();

        return 0;
}


// This objective function tries to minimize the value of the function
//
//                      Z = P1*(d1+)/T1 + P2*(d2+)/T2 + P3*(d3+)/T3
//                      Goal 1:  Minimize total costs (per thousand dollars)
//                      Goal 2:  Minimize Total Downtime
//                      Goal 3:  Minimize Total Failures (TF)

float
objective(GAGenome & c)
{
```

```
GABin2DecGenome & genome = (GABin2DecGenome &)c;

ifstream infile1("C:/galib/problems/reliable/reli_out.dat", ios::in);
        // associate Core_GA_output.dat with infile1
        // this file will hold the output from AweSim needed to guide the
           optimization program
        // specifically, the total sales (TS), the total orders (TO), the total
           reviews (TR), and the average inventory(AvgInv)
ofstream outfile("C:/galib/problems/reliable/reli_in.dat", ios::out);
        // associate GA_data.dat with outfile
        // this file will hold the input data for Awesim
        // specifically, the number to order (Q), the re-order point (R), and the
           time between reviews (TBREV)
ofstream outfile1("C:/galib/problems/reliable/reliout1.dat", ios::app);
        // associate ga1_out1.dat with outfile1, set it up to add to the file each
           time through
        // this file will store all visited points, and the corresponding Z and
           goal values
ofstream outfile2("C:/galib/problems/reliable/reliout2.dat", ios::app);
        // associate ga1_out2.dat with outfile2, set it up to add to the file each
           time through
        // this file will store all visited points, want to use it to compare if point
           previously visited
ofstream outfile4("C:/galib/problems/reliable/reliout4.dat", ios::app);
        // associate ga1_out2.dat with outfile4, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile5("C:/galib/problems/reliable/reliout5.dat", ios::out);
        // associate ga1_out5.dat with outfile5, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile6("C:/galib/problems/reliable/reliout6.dat", ios::app);
        // associate ga1_out6.dat with outfile6, set it up to add to the file each
           time through
        // trying to see if this will work
ofstream outfile7("C:/galib/problems/reliable/reliout7.dat", ios::app);
        // associate ga1_out7.dat with outfile7, set it up to add to the file each
           time through
        // want to store visited points, and simulation output data to check
           calculations
ofstream outfile8("C:/galib/problems/reliable/reliout8.dat", ios::app);
        // associate ga1_out8.dat with outfile8, set it up to add to the file each
           time through
        // want to store visited points, and simulation output data to check
           calculations
```

```
outfile1.setf(ios::fixed);
outfile6.setf(ios::fixed);
outfile7.setf(ios::fixed);
outfile8.setf(ios::fixed);

double Z, goal[3], targets[3];
int g, count = 0, i = 0;
double Fail1, Fail2, Repair1, Repair2;
int Fail1a, Fail2a, Repair1a, Repair2a;
double P;
double number, epsilon;
string hold, test;
double BD1_NUM, BD2_NUM, SYSBD_N, BD1_AVG, BD2_AVG, SYSBD_A,
      BD1_STD, BD2_STD, SYSBD_S;
double DT_AVG, DT_STD;
double timedif;


// associate Q, R, and TBREV with their corresponding genome.phenotypes
Fail1 = genome.phenotype(0);
Fail2 = genome.phenotype(1);
Repair1 = genome.phenotype(2);
Repair2 = genome.phenotype(3);

// The following code, converts the real number to only 2 decimal points
Fail1a = (100 * Fail1) + 0.5*(Fail1/fabs(Fail1));
Fail2a = (100 * Fail2) + 0.5*(Fail2/fabs(Fail2));
Repair1a = (100 * Repair1) + 0.5*(Repair1/fabs(Repair1));
Repair2a = (100 * Repair2) + 0.5*(Repair2/fabs(Repair2));
Fail1 = Fail1a/100.0;
Fail2 = Fail2a/100.0;
Repair1 = Repair1a/100.0;
Repair2 = Repair2a/100.0;

// define the target values for each goal
targets[0] = -126.16;
targets[1] = -0.10;
targets[2] = -6.62;

// write the input data for AweSim
outfile << Fail1 << '\t' << Fail2 << '\t' << Repair1 << '\t' << Repair2 << endl;

outfile5 << Fail1a << Fail2a << Repair1a << Repair2a << endl;
outfile.close();
outfile5.close();
```

```
ifstream infile2("C:/galib/problems/reliable/reliout2.dat", ios::in);
// associate ga1_out2.dat with infile2
// note that this is also the file associated with outfile2
// we use this to see if we have previously visited a point or not
ifstream infile5("C:/galib/problems/reliable/reliout5.dat", ios::in);
// associate ga1_out5.dat with infile5
// note that this is also the file associated with outfile2
// we use this to determine the current point

infile5 >> test;

infile2 >> hold >> number;
count1++;
cout << "count1 = " << count1 << endl;
while (!infile2.fail() && i < 10000)
{
        i++;
        count++;
        if (hold == test) {
                Z = number;
                i = 12000;
        }
        else {
                infile2 >> hold >> number;
        }
}

infile2.close();
infile5.close();

outfile4 << hold << endl;

// only call AweSim if the point has not previously been visited
if (i != 12000)
{
        // run AweSim and generate the output
        cout << "(" << Fail1 << ", " << Fail2 << ", " << Repair1 << ", " <<
            Repair2 <<")\n\n";  // Write out current point being simulated
        system ("callRELI.bat"); // this calls a file that calls the simulation model
        calls = calls + 1;
        cout << "calls = " << calls << endl;

        // collect the output from the AweSim runs
        infile1 >> BD1_NUM >> BD2_NUM >> SYSBD_N >> BD1_AVG >>
          BD2_AVG >> SYSBD_A >> BD1_STD >> BD2_STD >> SYSBD_S
          >> DT_AVG >> DT_STD;
```

```
        // For goal 0, we want to minimize total costs (per thousand dollars)
        // For goal 1, we want to minimize total downtime
        // For goal 2, we want to minimize total failures (TF)

        goal[0] = -1 * ((BD1_AVG * (Fail1 * 1)) + (BD2_AVG * (Fail2 * 2)) +
                ((SYSBD_A * 10) * (((1/Repair1) * 5) + ((1/Repair2) * 6))));
        goal[1] = -1 * DT_AVG;
        goal[2] = -1 * (BD1_AVG + BD2_AVG + SYSBD_A);

        // loop to calculate Z
        Z = 0;
        epsilon = 0.00005;
        g = 2;                          // where g is the number of goals - 1 (to
                                        work with C++ notation)
        for (int j = 0; j<=g; j++)
        {
                P = pow(100,(g-j));
                if (goal[j] >= (targets[j] - epsilon))
                {
                        goal[j] = targets[j];
                }

                // we want to calculate the normalized objective function
                // following the equation, Z = sum (Pi*di/Ti)
                Z = Z + ((goal[j] - targets[j])/fabs(targets[j]))*P;
        }
        Z = 1*Z;
        time_t temp;
        time(&temp);
        timedif = temp - begin;
        clock_t clock(void);

        outfile8 << i << '\t' << calls << '\t' << timedif << '\t' << Fail1 << '\t' <<
            Fail2 << '\t' << Repair1 << '\t' << Repair2 << '\t' << Z << '\t' <<
            goal[0] << '\t' << goal[1] << '\t' << goal[2] << '\t' << BD1_AVG << '\t'
            << BD2_AVG << '\t' << SYSBD_A << '\t' << BD1_STD << '\t' <<
            BD2_STD << '\t' << SYSBD_S << '\t' << DT_AVG << '\t' <<
            DT_STD << '\n';

        outfile6 << Fail1 << Fail2 << Repair1 << Repair2 << '\t' << Fail1 << '\t'
            << Fail2 << '\t' << Repair1 << '\t' << Repair2 << '\t' << Z << '\t' <<
            "count=" << count << '\t' << "i=" << i << '\n';
}

// write out the results
```

```
        outfile1 << Fail1 << '\t' << Fail2 << '\t' << Repair1 << '\t' << Repair2 << '\t' << Z
            << '\t' << goal[0] << '\t' << goal[1] << '\t' << goal[2] << '\t' << "count=" <<
            count << '\t' << "i=" << i << '\n';
        outfile2 << Fail1a << Fail2a << Repair1a << Repair2a << '\t' << Z << '\n';
        outfile7 << i << '\t' << calls << '\t' << timedif << '\t' << Fail1 << '\t' << Fail2 <<
            '\t' << Repair1 << '\t' << Repair2 << '\t' << Z << '\t' << goal[0] << '\t' <<
            goal[1] << '\t' << goal[2] << '\t' << BD1_AVG << '\t' << BD2_AVG << '\t' <<
            SYSBD_A << '\t' << BD1_STD << '\t' << BD2_STD << '\t' << SYSBD_S <<
            '\t' << DT_AVG << '\t' << DT_STD << '\n';

        infile1.close();
        outfile1.close();
        outfile2.close();
        outfile4.close();
        outfile6.close();
        outfile7.close();
        outfile8.close();

        return Z;
}
```

Here is the file "callRELI.bat";

```
cd\
cd projects\dissert
execute reliable
cd\
cd galib\problems\reliable
```

LNM-SO Method

The LNM-SO method was coded using MATLAB. The specific coding for each test bed problem consisted of a main program and several subroutines. For the inventory domain, the main program "LNM_INV" is presented below along with the following subroutines: "Xsimplex2", "simulate_det_inv", "ftn_eval_inv1", "matrix_compare", "range2", "Z_objectiveFunction". Additionally, a file that was used to call the simulation model "call_inv.bat" is presented.

For the remaining domains, the coding is identical for the "matric_compare", "Z_objectiveFunction", and very similar for "range2" subroutines (the only difference for "range2" was that it was modified to be able to handle the integer variable for the logistics domain. This subroutine was called "range1"). Additionally, the coding is very similar for the main program, and the "Xsimplex2_inv" and "simulate_det_inv" subroutines. The main difference in these programs is that they reference the appropriate domain as well as the specific files written to and the information included in these files. Thus, these programs are not included. However, the corresponding ftn_eval_inv1 file for each domain is included below. The subroutine sets up the specific goal equations. Additionally, the file used to call the appropriate simulation model is included for each domain.

Inventory Domain

Main Program: LNM_INV

function [optZ,Xoptimal] = LNM_INV

%This is the Lexicographic Nelder-Mead Simulation Optimization (LNM-SO) method that was used to solve the inventory domain test bed problem.

```
% This program creates five files:
% 1) "LNMsummary_data" is used to summarize the results of each of run in 1 file
% 2) "soln_hist" is used to keep track of the the best solution found thus far and is
        recorded after each iteration
% 3) "pts_simulated" keeps track of all the solutions that have been simulated
% 4) "pts_all" keeps track of all the solutions that have been evaluated (regardless of
        whether they have previously been simulated or not)
% 5) "results" is used to write out the final results from the LNM-SO run

fid=fopen('C:/TEMP/methods/work/inv/LNMsummary_data.dat', 'w');
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'Run #', 'starting point',
      'starting Z', 'i', 'ROCCA', 'CPU_CA', 'Count', 'Optimal Point', 'OptZ', '% Goals Sat');
fclose(fid);

rand('state', sum(100*clock));
for run = 1:20
   runnum = num2str(run);

   t_begin = clock;
   Xrange = [30 100; 0 30; 1 10];
   X = start(Xrange);
   startingpoint = X
   Phi = [-12000 0.05 0.225]; %The target values
   targets = Phi;
   n=length(X);      %The number of variables
   g=length(targets);    %The number of goals
   calls = 0;
   count = 0;

   alpha = 1;
   beta = 0.5;
   gamma = 2;
   delta = 0.9;
   End = 0;
   optimal = 0;
   epsilon = 0.1;
   P=n+1;

   soln_hist = ['C:/TEMP/methods/work/inv/LNMrun',runnum,'/soln_hist.dat'];
   pts_simulated = ['C:/TEMP/methods/work/inv/LNMrun',runnum,'/pts_simulated.dat'];
   pts_all = ['C:/TEMP/methods/work/inv/LNMrun',runnum,'/pts_all.dat'];
   results = ['C:/TEMP/methods/work/inv/LNMrun',runnum,'/results.dat'];

   % Clear the files, so that we ensure they only contain data from this particular run
   fid=fopen(soln_hist, 'w');
```

```
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'i', 'Q', 'R', 'TBREV', 'Z', 'G1',
   'G2', 'G3');
fclose(fid);

fid=fopen(pts_simulated, 'w');
fclose(fid);

fid=fopen(pts_all, 'w');
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'Pt', 'i', 'calls',
   'clocktime', 'Q', 'R', 'TBREV', 'Z', 'G1', 'G2', 'G3');
fclose(fid);

loop = 5;
X = [X; 86 24 8.2; 44 6 2.8; 86 24 2.8]
%Determine the goal and objective function value of the starting solution
[G_start, calls, count] = simulate_det_inv(startingpoint, Phi, g, End, calls, t_begin,
   Xrange, pts_simulated, pts_all, count);
Z_start = Z_ObjectiveFunction(G_start, Phi, g);
Goptimal = [10000 10000 10000];

%Start the first iteration:
while loop == 5
   End = End + 1;
   [X_low, X_high, X_sechi, G_low, G_high, G_sechi, calls, count] =
         Xsimplex2_inv(X, Phi, n, P, g, End, calls, t_begin, Xrange, pts_simulated,
         pts_all, count);
   for u = 1:n+1
      for v = 1:n
         Xtemp(v) = X(u,v);
      end
      if Xtemp == X_low
         I_low = u;
      elseif Xtemp == X_sechi
         I_sechi = u;
      elseif Xtemp == X_high
         I_high = u;
      end
   end

   %The range2 subroutine ensures feasibility and rounds each variable to the nearest
   %hundredth.
   X_low = range2(X_low, Xrange, n);
   X_sechi = range2(X_sechi, Xrange, n);
   X_high = range2(X_high, Xrange, n);

   Xsum = sum(X);
```

```
X_cent = (1/n) * (Xsum - X_high);
X_cent1 = X_cent;    %This is only used for calculations and thus is not restricted to
        2 decimal places.
X_reflect = (1 + alpha) * X_cent - (alpha * X_high);
X_cent = range2(X_cent, Xrange, n);
X_reflect = range2(X_reflect, Xrange, n);

[G_reflect, calls, count] = simulate_det_inv(X_reflect, Phi, g, End, calls, t_begin,
        Xrange, pts_simulated, pts_all, count);
[G_cent, calls, count] = simulate_det_inv(X_cent, Phi, g, End, calls, t_begin,
        Xrange, pts_simulated, pts_all, count);

Gcompare_low_opt = matrix_compare(G_low, Goptimal, g);
Gcompare_ref_opt = matrix_compare(G_reflect, Goptimal, g);
Gcompare_cent_opt = matrix_compare(G_cent, Goptimal, g);

if G_low <= Phi
    Xoptimal = X_low;
    loop = 0;
    optimal = 1;
elseif Gcompare_low_opt < 0
    Xoptimal = X_low;
    Goptimal = G_low;
end

if optimal ~= 1
    if G_reflect <= Phi
        Xoptimal = X_reflect;
        loop = 0;
        optimal = 1;
    elseif G_cent <= Phi
        Xoptimal = X_cent;
        loop = 0;
        optimal = 1;
    end
end

if Gcompare_ref_opt < 0
    Xoptimal = X_reflect;
    Goptimal = G_reflect;
end
if Gcompare_cent_opt < 0
    Xoptimal = X_cent;
    Goptimal = G_cent;
end
```

```
Gcompare_low_ref = matrix_compare(G_low, G_reflect, g);
Gcompare_ref_sechi = matrix_compare(G_reflect, G_sechi, g);
Gcompare_ref_high = matrix_compare(G_reflect, G_high, g);

%Is G_low <= G(Xreflect) <= G_sechi?  Yes:  Xreflect replaces Xhigh
if (Gcompare_low_ref <= 0) && (Gcompare_ref_sechi <= 0)
   for v = 1:n
      X(I_high,v) = X_reflect(v);
   end
   %Is G(Xreflect) < G_low?  Yes:  Attempt expansion.
elseif Gcompare_low_ref > 0
   X_expansion = (gamma*X_reflect) + (1-gamma)*X_cent1;
   X_expansion = range2(X_expansion, Xrange, n);
   [G_expansion, calls, count] = simulate_det_inv(X_expansion, Phi, g, End, calls,
         t_begin, Xrange, pts_simulated, pts_all, count);

   Gcompare_exp_low = matrix_compare(G_expansion, G_low, g);
   Gcompare_exp_ref = matrix_compare(G_expansion, G_reflect, g);
   Gcompare_exp_opt = matrix_compare(G_expansion, Goptimal, g);

   if Gcompare_exp_opt < 0
      Xoptimal = X_expansion;
      Goptimal = G_expansion;
   end

   if Gcompare_exp_low <= 0
      if Gcompare_exp_ref <= 0
         for v = 1:n
            X(I_high,v) = X_expansion(v);
         end
      else
         for v = 1:n
            X(I_high,v) = X_reflect(v);
         end
      end
   else
      for v = 1:n
         X(I_high,v) = X_reflect(v);
      end
   end

   %Is G(Xreflect) > G_sechi?  Yes:  Then attempt contraction.
   %If G(Xreflect) <= G_high, then Xrefl replaces Xhigh before attempting
         contraction.
elseif Gcompare_ref_sechi > 0;
   if Gcompare_ref_high <= 0;
```

```
      X_high = X_reflect;
      G_high = G_reflect;
   end
   X_contraction = (beta*X_high) + (1-beta)*X_cent1;
   X_contraction = range2(X_contraction, Xrange, n);
   [G_contraction, calls, count] = simulate_det_inv(X_contraction, Phi, g, End, calls,
        t_begin, Xrange, pts_simulated, pts_all, count);
   Gcompare_cont_high = matrix_compare(G_contraction, G_high, g);
   Gcompare_cont_opt = matrix_compare(G_contraction, Goptimal, g);
   if Gcompare_cont_opt < 0
      Xoptimal = X_contraction;
      Goptimal = G_contraction;
   end
   if Gcompare_cont_high <= 0
      for v = 1:n
         X(I_high,v) = X_contraction(v);
      end
   elseif Gcompare_cont_high > 0
      %We now need to Shrink the entire simplex.
      for u = 1:n+1
         for v = 1:n
            Xhold(v) = X(u,v);
         end
         if Xhold ~= X_low
            Xshrink = (delta*Xhold) + (1-delta)*X_low;
            Xshrink = range2(Xshrink, Xrange, n);
            [Gshrink, calls, count] = simulate_det_inv(Xshrink, Phi, g, End, calls,
                  t_begin, Xrange, pts_simulated, pts_all, count);
            Gcompare_shrink_opt = matrix_compare(Gshrink, Goptimal, g);
            if Gcompare_shrink_opt < 1
               Xoptimal = Xshrink;
               Goptimal = Gshrink;
            end
            for v = 1:n
               X(u,v) = Xshrink(v);
            end
         end
      end
   end
end

%Step 9

if optimal == 1
   loop = 0;
end
```

```
    Zoptimal = Z_ObjectiveFunction(Goptimal, Phi, g);
    solution_history(End,:)=[End,Xoptimal,Zoptimal,Goptimal];
    solution_store(End)=Zoptimal;

    if End >= 100
       stop = terminate(solution_store,End);     % Terminate if convergence is better than
            0.99
       if stop >= 0.99
          loop = 0;
       end
    end
    if End >= 250  %Set the maximum number of iterations
       loop = 0;
    end

    %Print Solution history to file
    fid=fopen(soln_hist, 'a');
    fprintf(fid, '%d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', solution_history(End,:));
    fclose(fid);
end

Phi
optX = Xoptimal
[Goptimal, calls, count] = simulate_det_inv(Xoptimal, Phi, g, End, calls, t_begin,
        Xrange, pts_simulated, pts_all, count)
optZ = Z_ObjectiveFunction(Goptimal, Phi, g)

% Calculate and print the results to the screen
GOAL = satisfied_goals(Goptimal, targets, g);
t_end = clock;
clocktime = etime(t_end, t_begin)
run
'Number of Goals Satisfied = ', disp(GOAL)
'Number of Goals Not Satisfied = ', disp(g - GOAL)
'Goals Satisfied = ', disp(goals_satisfied(Goptimal, targets, g))
'Percentage of Goals Satisfied = ', disp((GOAL/g)*100)
GS = GOAL;
GNS = g - GOAL;
GSS = goals_satisfied(Goptimal, targets, g);
GSP = (GOAL/g)*100;
cputime1 = ['Run Time = ' num2str(clocktime) ' seconds'];
cputime2 = ['Run Time = ' num2str(clocktime/60) ' minutes'];
clocktime2 = clocktime/60;
disp(cputime1)
disp(cputime2)
```

```
   fid=fopen(results, 'w');
   fprintf(fid, '%s%f %s%f %s%f%s\n %s%d\n %s%d\n %s%d\n %s%f\n %s%f\n
        %s%f\n %s%f\n %s%f\n %s%f\n %s%d\n %s%d\n %s%f\n', 'starting point = [',
        startingpoint(1), ' ', startingpoint(2), ' ', startingpoint(3), ']', '# of iterations = ', End,
        'Count = ', count, 'ROCCA = ', calls, 'Run Time in seconds = ', clocktime, 'Run
        Time in minutes = ', clocktime2, 'OptQ = ', optX(1), 'OptR = ', optX(2),
        'OptTBREV = ', optX(3), 'OptZ = ', optZ, 'Number of Goals Satisfied = ', GS,
        'Number of Goals Not Satisfied = ', GNS, 'Percentage of Goals Satisfied = ', GSP);
   fclose(fid);

   fid=fopen(pts_simulated, 'a');
   fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t
        %s\t %s\t %s\t %s\t %s\n', 'Pt', 'i', 'calls', 'clocktime', 'Q', 'R', 'TBREV', 'Z', 'G1',
        'G2', 'G3', 'TS', 'TO', 'TR', 'AvgInv', 'Runtime', 'BWT', 'CWT', 'LS', 'BO');
   fclose(fid);

   fid=fopen('C:/TEMP/methods/work/inv/LNMsummary_data.dat', 'a');
   fprintf(fid, '%d\t %s%4f%s%4f%s%4f%s\t %f\t %d\t %d\t %f\t %d\t
        %s%4f%s%4f%s%4f%s\t %f\t %f\n', run, '[', startingpoint(1), ', ', startingpoint(2),
        ', ', startingpoint(3), ']', Z_start, End, calls, clocktime, count, '[', optX(1), ', ',
        optX(2), ', ', optX(3), ']', optZ, GSP);
   fclose(fid);

   clear all %This is done to clear all the variables before we begin the next run.
end
```

<u>Xsimplex2_inv subroutine</u>

```
function [X_low, X_high, X_sechi, G_low, G_high, G_sechi, calls1, count] =
      Xsimplex2_inv(X, Phi, n, P, g, i, calls1, t_begin, Xrange, pts_simulated, pts_all,
      count)
%X_low calculates the vertices with the lowest (best) function value.
%X_high calculates the vertices with the highest (worst) function value.
%X_sechi calculates the vertices with the second hightest function value.

for q = 1:2
   for u = 1:P
      if q == 1
         for v = 1:n
            Xtemp(u,v) = X(u,v);
         end
         [Gtemp(u,:), calls1, count] = simulate_det_inv(Xtemp(u,:), Phi, g, i, calls1,
               t_begin, Xrange, pts_simulated, pts_all, count);
      end
```

```
        if q==2
            if Xtemp(u,:) == X_hightest
                Gtemp(u,:) = -1 * (abs(100 * Gtemp(u,:)));
            end
        end

        if u == 1
            Xhold_hi = Xtemp(u,:);
            Ghold_hi = Gtemp(u,:);
            Xhold_low = Xtemp(u,:);
            Ghold_low = Gtemp(u,:);
        else
            Gcompare_hi = matrix_compare(Gtemp(u,:), Ghold_hi, g);
            Gcompare_low = matrix_compare(Gtemp(u,:), Ghold_low, g);

            if Gcompare_hi > 0
                Xhold_hi = Xtemp(u,:);
                Ghold_hi = Gtemp(u,:);
            elseif Gcompare_low < 0 && q==1;
                Xhold_low = Xtemp(u,:);
                Ghold_low = Gtemp(u,:);
            end
        end
    end
    X_hightest = Xhold_hi;
    G_hightest = Ghold_hi;

    if q == 1
        X_high = Xhold_hi;
        G_high = Ghold_hi;
        X_low = Xhold_low;
        G_low = Ghold_low;
    elseif q ==2
        X_sechi = X_hightest;
        G_sechi = G_hightest;
    end
end
```

<u>simulate_det_inv subroutine</u>

```
function [goal, calls1, count] = simulate_det_inv(Xnow, targets, g, i, calls1, t_begin,
        Xrange, pts_simulated, pts_all, count)

%This program determines if we have already simulated the point or not
XXnow = Xnow;
```

```
hold1 = 0;
testa = 'test';
goal = [250000000, 250000000, 250000000];

% First we check to see if the point is feasible, if so we call the
% simulation, otherwise we do not.
infeasible = feasibility(Xnow,Xrange);
if infeasible > 0
   objval = 100000000000000000000;
else
   % If the point is feasible, we check to see if we have previously simulated the point
   % If so, we get the data from the previously visited point
   testa = [num2str(XXnow(1)) num2str(XXnow(2)) num2str(XXnow(3))];
   [hold, misc1, misc2, misc3, misca, miscb, miscc, num, tempgoal1, tempgoal2,
        tempgoal3, misc4, misc5, misc6, misc7, misc8, misc9, misc10, misc11, misc12] =
        textread(pts_simulated, '%s %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
        %f %f %f %f');
   calls2 = calls1;

   for jj = 1:calls1
      hold1(jj) = strcmp(testa, hold(jj));
   end

   [compare,index] = max(hold1);
   if (compare == 1)
      objval = num(index);
      goal = [tempgoal1(index), tempgoal2(index), tempgoal3(index)];
   else
      %if the point has not been previously visited, we call the
      %simulation and get the results
      calls1 = calls1 + 1;
      goal = ftn_eval_inv1(Xnow, targets, g, i, calls1, t_begin, pts_simulated);
      objval = Z_ObjectiveFunction(goal, targets, g);
   end
end

%Print all points evaluated to file: "pts_all.dat"
fid=fopen(pts_all, 'a');
count = count + 1;
t_now = clock;
clocktime = etime(t_now, t_begin);
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', testa, i, calls1,
      clocktime, Xnow(1), Xnow(2), Xnow(3), objval, goal(1), goal(2), goal(3));
fclose(fid);
```

ftn_eval_inv1 subroutine

```
function goal = ftn_eval_inv1(X, targets, g, i, calls, t_begin, pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [Q, R, TBREV]
% These values are sent to AweSim via the file "tabudata.dat"

% The output is then written in AweSim to the file "Core_tabu_output.dat", that is read
% here to get the needed output to calculate the objective function value.

LT = 1;     %This variable is fixed.
TBD = 1;    %This variable is fixed.
XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);

fid=fopen('C:/TEMP/work/tabudata.dat', 'w');
fprintf(fid, '%3f\t %3f\t %3f\t %3ld\t %3ld', X, LT, TBD);
fclose(fid);

% Call the file call_INV.bat to call the simulation model
system('C:/TEMP/methods/work/call_INV.bat');

[TS, TO, TR, AvgInv, Runtime, LS, CWT, BWT, BO] =
      textread('C:/TEMP/work/Core_tabu_output.dat', '%f %f %f %f %f %f %f %f %f');
goal(1) = -1 * ((TS * (65-40)) - (TO * 50) - (TR * 30) - (AvgInv * .004 * Runtime) - (LS
      * 20) - (BO * 10));
goal(2) = CWT;
goal(3) = BWT;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
```

```
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = Z;
sample1(8) = goal(1);
sample1(9) = goal(2);
sample1(10) = goal(3);
sample1(11) = TS;
sample1(12) = TO;
sample1(13) = TR;
sample1(14) = AvgInv;
sample1(15) = Runtime;
sample1(16) = BWT;
sample1(17) = CWT;
sample1(18) = LS;
sample1(19) = BO;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
    %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

<u>matrix_compare subroutine</u>

```
function x = matrix_compare(Gone, Gzero, g)
%Matrix_Compare allows comparison of two goals (i.e., < > =).
%If Gone = Gzero, Gcompare = 0.
%If Gone < Gzero, Gcompare = -1.
%If Gone > Gzero, Gcompare = 1.

i = 1;
while i <= g
  if Gone == Gzero
    Gcheck = 0;
    i = g + 1;
  elseif Gone(i) == Gzero(i)
    i = i + 1;
  elseif Gone(i) < Gzero(i)
    Gcheck = -1;
    i = g + 1;
```

```
    elseif Gone(i) > Gzero(i)
        Gcheck = 1;
        i = g + 1;
    end
end

x = Gcheck;
```

---

range2 subroutine

```
function test = range2(Xstore, Xrange, n)
%Range checks to see if the new point is within the range.

for v = 1:n
    if Xrange(v,1) > Xstore(v)
        w(v) = -1000;
    elseif Xstore(v) > Xrange(v,2)
        w(v) = -100;
    else
        w(v) = 1;
    end
end
sum = 0;
for v = 1:n
    sum = sum + w(v);
end

if sum < 0
    for v = 1:n
        if w(v) == -1000
            Xstore(v) = Xrange(v,1);
        elseif w(v) == -100
            Xstore(v) = Xrange(v,2);
        end
    end
end

for v=1:n
    XX(v) = round((Xstore(v) * 100));
    Xstore(v) = XX(v)/100.0;
end
test = Xstore;
```

Z_ObjectiveFunction subroutine

```
function Z = Z_ObjectiveFunction(objective, targets, g)
% Objective calculates the normalized Objective Function, Z, for the problem.
% We are trying to minimize the following equation: sum(Pi*(di+)/Ti).

Z = 0;
for i = 1:g
   P = 100^(g-i);
   Z = Z + ((objective(i) - targets(i))/abs(targets(i)) * P);
end
```

call_inv.bat

```
cd\
cd projects\dissert2
execute inv
cd\
cd TEMP\methods\work
```

Logistics Domain

ftn_eval_log1 subroutine

```
function goal = ftn_eval_log1(X, targets, g, i, calls, t_begin, pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [FAIL, ARRIVE, DISC1, DISC2, DISC3]
% These values are sent to AweSim via the file "log_in.dat"

% The output is then written in AweSim to the file "log_out.dat", that is read here to get
% the needed output to calculate the objective function value.

XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);
XX(4) = X(4);
XX(5) = X(5);

fid=fopen('C:/TEMP/work/logistic/log_in.dat', 'w');
fprintf(fid, '%3ld\t %f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callLOG.bat to call the simulation model
system('C:/TEMP/work/logistic/callLOG.bat');

[ORD1_AVG, ORD2_AVG, ORD3_AVG, OUT1_AVG, OUT2_AVG, OUT3_AVG,
      OUT1_STD, OUT2_STD, OUT3_STD, UTIL_AVG, UTIL_STD] =
      textread('C:/TEMP/work/logistic/log_out.dat', '%f %f %f %f %f %f %f %f %f %f
      %f');
goal(1) = (ORD1_AVG * 25) + (ORD2_AVG * 20) + (ORD3_AVG * 30) +
      (OUT1_AVG * 3) + (OUT2_AVG * 2) + (OUT3_AVG *4);
goal(2) = -1 * (UTIL_AVG/100);

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);
```

```
% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(X(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))
num2str(XX(5))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = X(5);
sample1(9) = Z;
sample1(10) = goal(1);
sample1(11) = goal(2);
sample1(12) = ORD1_AVG;
sample1(13) = ORD2_AVG;
sample1(14) = ORD3_AVG;
sample1(15) = OUT1_AVG;
sample1(16) = OUT2_AVG;
sample1(17) = OUT3_AVG;
sample1(18) = OUT1_STD;
sample1(19) = OUT2_STD;
sample1(20) = OUT3_STD;
sample1(21) = UTIL_AVG;
sample1(22) = UTIL_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
    %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callLOG.bat

```
cd\
cd projects\dissert2
execute logist2
cd\
cd TEMP\work
```

PERT Domain

ftn_eval_pert1 subroutine

```
function goal = ftn_eval_pert1(X, targets, g, i, calls, t_begin, pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [LOW3, LOW4, LOW6]
% These values are sent to AweSim via the file "pertin.dat"

% The output is then written in AweSim to the files "pertout.dat" and "pertcrit.dat", that
% is read here to get the needed output to calculate the objective function value.

XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);

fid=fopen('C:/TEMP/work/pert/pertin.dat', 'w');
fprintf(fid, '%f\t %f\t %f', X);
fclose(fid);

% Call the file callPERT.bat to call the simulation model
system('C:/TEMP/work/pert/callPERT.bat');

[NODE2AVG, NODE2STD, NODE3AVG, NODE3STD, NODE4AVG, NODE4STD,
      NODE5AVG, NODE5STD, PT_AVG, PT_STD] =
      textread('C:/TEMP/work/pert/pertout.dat', '%f %f %f %f %f %f %f %f %f %f');
[CRIT1, CRIT2, CRIT3, CRIT4, CRIT5, CRIT6, CRIT7, CRIT8, CRIT9 ]=
      textread('C:/TEMP/work/pert/pertcrit.dat', '%f %f %f %f %f %f %f %f %f');
goal(1) = (CRIT1 * 5 * 2) + (CRIT2 * 3 * 2) + (CRIT3 * 2 * 1) + (CRIT4 * 7 * 1) +
      (CRIT5 * 4 * 2) + (CRIT6 * 3 * 1) + (CRIT7 * 2 * 2) + (CRIT8 * 5 * 1) + (CRIT9 *
      4 * 2);
goal(2) = PT_AVG;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
```

```
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = Z;
sample1(8) = goal(1);
sample1(9) = goal(2);
sample1(10) = CRIT1;
sample1(11) = CRIT2;
sample1(12) = CRIT3;
sample1(13) = CRIT4;
sample1(14) = CRIT5;
sample1(15) = CRIT6;
sample1(16) = CRIT7;
sample1(17) = CRIT8;
sample1(18) = CRIT9;
sample1(19) = PT_AVG;
sample1(20) = PT_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
        %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callPERT.bat

```
cd\
cd projects\dissert2
execute pert
cd\
cd TEMP\work
```

Production Domain

ftn_eval_prod1 subroutine

```
function goal = ftn_eval_prod1(X, targets, g, i, calls, t_begin, pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [LOAD, INSPECT, UNLOAD, TRANSPORT1, TRANSPORT2,
      TRANSPORT3]
% These values are sent to AweSim via the file "prod_in.dat"

% The output is then written in AweSim to the file "prod_out.dat", that is read here to get
% the needed output to calculate the objective function value.

XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);
XX(4) = X(4);
XX(5) = X(5);
XX(6) = X(6);

fid=fopen('C:/TEMP/work/product/prod_in.dat', 'w');
fprintf(fid, '%f\t %f\t %f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callPROD.bat to call the simulation model
system('C:/TEMP/work/product/callPROD.bat');

[TPUT, TIS_AVG, TIS_STD, PU_AVG, PU_STD, CU_AVG, CU_STD] =
      textread('C:/TEMP/work/product/prod_out.dat', '%f %f %f %f %f %f %f');
goal(1) = -1 * (100 * TPUT - (1000 * (1/TIS_AVG)));
goal(2) = -1 * (PU_AVG/6 + CU_AVG/2);

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);
```

```
% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))
num2str(XX(5)) num2str(XX(6))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = X(5);
sample1(9) = X(6);
sample1(10) = Z;
sample1(11) = goal(1);
sample1(12) = goal(2);
sample1(13) = TPUT;
sample1(14) = TIS_AVG;
sample1(15) = TIS_STD;
sample1(16) = PU_AVG;
sample1(17) = PU_STD;
sample1(18) = CU_AVG;
sample1(19) = CU_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
    %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callPROD.bat

```
cd\
cd projects\dissert2
execute product
cd\
cd TEMP\work
```

Reliability Domain

ftn_eval_reli1 subroutine

```
function goal = ftn_eval_reli1(X, targets, g, i, calls, t_begin, pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [Fail1, Fail2, Repair1, Repair2]
% These values are sent to AweSim via the file "reli_in.dat"

% The output is then written in AweSim to the file "reli_out.dat", that is read here to get
% the needed output to calculate the objective function value.

XX(1) = X(1);
XX(2) = X(2);
XX(3) = X(3);
XX(4) = X(4);

fid=fopen('C:/TEMP/work/reliable/reli_in.dat', 'w');
fprintf(fid, '%f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callRELI.bat to call the simulation model
system('C:/TEMP/work/reliable/callRELI.bat');

[BD1_NUM, BD2_NUM, SYSBD_N, BD1_AVG, BD2_AVG, SYSBD_A, BD1_STD,
      BD2_STD, SYSBD_S, DT_AVG, DT_STD] =
      textread('C:/TEMP/work/reliable/reli_out.dat', '%f %f %f %f %f %f %f %f %f %f
      %f');
goal(1) = (BD1_AVG * X(1) * 1) + (BD2_AVG * X(2) * 2) + (SYSBD_A * 10 *
      ((1/X(3) * 5) + (1/X(4) * 6)));
goal(2) = DT_AVG;
goal(3) = BD1_AVG + BD2_AVG + SYSBD_A;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);
```

```
% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = Z;
sample1(9) = goal(1);
sample1(10) = goal(2);
sample1(11) = goal(3);
sample1(12) = BD1_NUM;
sample1(13) = BD2_NUM;
sample1(14) = SYSBD_N;
sample1(15) = BD1_AVG;
sample1(16) = BD2_AVG;
sample1(17) = SYSBD_A;
sample1(18) = BD1_STD;
sample1(19) = BD2_STD;
sample1(20) = SYSBD_S;
sample1(21) = DT_AVG;
sample1(22) = DT_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
    %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callRELI.bat

```
cd\
cd projects\dissert2
execute reliable
cd\
cd TEMP\work
```

TS-SO Method

The TS-SO method was coded using MATLAB. The basic tabu search code used to drive the main program was based on “tabuks.m”, a MATLAB file used to solve a knapsack problem using tabu search (Aurdal, 2006). This code was altered extensively to allow the code to work with real numbers and to be able to solve the TS-SO method. The specific coding for each test bed problem consisted of a main program and several subroutines. For the inventory domain, the main program “INV” is presented below along with the following subroutines: “Evaluate_Simulation_TabuINV”, “neighbor_real”, and “Z_ObjectiveFunction”. Additionally, a file that was used to call the simulation model “call_inv.bat” is presented.

For the remaining domains, the coding is identical for “Z_ObjectiveFunction”, and very similar for “neighbor_real”subroutines (the only difference is that a different neighbor routine, called “neighbor_mixed”, was used for the logistics domain to address the integer variable). Additionally, the coding is very similar for the main program. The main difference between the main programs is that they reference the appropriate domain as well as the specific files written to and the information included in these files. Thus, these programs are not included. However, the corresponding “Evaluate_Simulation_TabuINV” file for each domain is included below. This subroutine sets up the specific goal equations. Additionally, the file used to call the appropriate simulation model is included for each domain.

Inventory Domain

Main Program: INV

```
function [optZ,optX]=INV

% This is the Tabu Search Simulation Optimization (TS-SO) method that
% was used to solve the inventory domain test bed problem.

% This program creates five files:
% 1) "summary_data" is used to summarize the results of each of run in one file
% 2) "soln_hist" is used to record the best solution found during the current iteration
% 3) "pts_simulated" keeps track of all the solutions that have been simulated
% 4) "pts_all" keeps track of all the solutions that have been evaluated (regardless of
        whether they have previously been simulated or not
% 5) "results" is used to write out the final results from the LNM-SO run

fid=fopen('C:/TEMP/work/invent/summary_data.dat', 'w');
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'start_Q', 'start_R',
      'start_TBREV', 'i', 'ROCCA', 'CPU_CA', 'OptQ', 'OptR', 'OptTBREV', 'OptZ', '%
      Goals Sat');
fclose(fid);

rand('state', sum(100*clock));
for run = 1:20
   runnum = num2str(run);

   % Start by initializing the number of iterations to zero
   t_begin = clock;
   i=1;   %Keeps track of the number of iterations
   calls = 0;   %Keeps track of the number of calls to the simulation
   stop = 0;
   Y = [-1 0 0; 1 0 0; 0 -1 0; 0 1 0; 0 0 -1; 0 0 1];   %left-hand side coefficients of upper
      and lower bound constraints
   q = [30; 100; 0; 30; 1; 10];   %right-hand side of constraints
   i_max = 100;
   ttinc = 2;   %Lifespan on tabu list if X(i) is increased
   ttdec = 2;   %Lifespan on tabu list if X(i) is decreased
   hist = 10;   %Lifespan on the tabu solution history list
   targets = [-12000 0.05 0.225];
   Xrange = [30 100; 0 30; 1 10];
   W=Y;
   m=q;
   X = start(Xrange);
   startingpoint = X;
   goal = [100000 100000];
```

```
complete = 1;

soln_hist = ['C:/TEMP/work/invent/run',runnum,'/soln_hist.dat'];
pts_simulated = ['C:/TEMP/work/invent/run',runnum,'/pts_simulated.dat'];
pts_all = ['C:/TEMP/work/invent/run',runnum,'/pts_all.dat'];
results = ['C:/TEMP/work/invent/run',runnum,'/results.dat'];

% Clear the files, so that we ensure they only contain data from this particular run
fid=fopen(soln_hist, 'w');
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'i', 'Q', 'R', 'TBREV', 'Z', 'G1',
   'G2', 'G3');
fclose(fid);

fid=fopen(pts_simulated, 'w');
fclose(fid);

fid=fopen(pts_all, 'w');
fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'Pt', 'i', 'calls',
   'clocktime', 'Q', 'R', 'TBREV', 'Z', 'G1', 'G2', 'G3');
fclose(fid);

while complete < 3
   n=length(X);         %The number of variables
   v=length(m);         %The number of constraints
   g=length(targets);   %The number of goals
   currP=zeros(1,4*n)'; %The weights of the objective function

   % Initialize the tabu lists
   tabu_add=zeros(1,4*n);
   tabu_decr=zeros(1,4*n);
   tabu_hist=zeros(hist,n);

   % Initially the optimal X value equals the X value
   if complete == 1
      optX=X;
   end

   % Initialize the optimal objective function value
   % First we check to see if the point is infeasible (i.e., infeasible > 0)
   % If so, we do not call the simulation model, otherwise we do.
   infeasible = feasibility(X,Xrange);
   if infeasible > 0
      optZ =  10000000000000000000;
   else
      calls = calls + 1;
```

```
   [optZ, optGoal] = Evaluate_Simulation_TabuINV(optX, targets, g, i, calls,
         t_begin, pts_simulated);
end

% Loop as long as the number of iterations is not exceeded
while(stop<0.99)
   % Decrement all non-zero values on the tabu lists
   tabu_add=tabu_add-(tabu_add>0);
   tabu_decr=tabu_decr-(tabu_decr>0);
   for h=hist:-1:2
      tabu_hist(h,:)=tabu_hist(h-1,:);
   end
   tabu_hist(1,:)=X;
   tabu_hist;

   % Here we generate the neighborhood (N) of the current solution X.
   N=neighbor_real(X,complete);

   % Based on the solutions contained in N, calculate a vector of corresponding
   % current weight values.
   currW=N*W';

   for j=1:4*n
      Xnow=N(j,:);

      % First we check to see if the point is feasible, if so we call the
      % simulation, otherwise we do not.
      infeasible = feasibility(Xnow,Xrange);
      if infeasible > 0
         objval = 1000000000000000000;
      else
         % If the point is feasible, we check to see if we have previously simulated
               the point
         % If so, we get the data from the previously visited point
         % Convert to only two decimal places
         XXnow(1) = round((Xnow(1) * 100));
         XXnow(2) = round((Xnow(2) * 100));
         XXnow(3) = round((Xnow(3) * 100));
         Xnow(1) = XXnow(1)/100.0;
         Xnow(2) = XXnow(2)/100.0;
         Xnow(3) = XXnow(3)/100.0;

         test = [num2str(XXnow(1)) num2str(XXnow(2)) num2str(XXnow(3))];
         [hold, misc1, misc2, misc3, misca, miscb, miscc, num, tempgoal1,
               tempgoal2, tempgoal3, misc4, misc5, misc6, misc7, misc8, misc9,
```

```
            misc10, misc11, misc12] = textread(pts_simulated, '%s %f %f %f %f
            %f %f %f %f %f %f %f %f %f %f %f %f %f %f');
        calls;
        for jj = 1:calls
          hold1(jj) = strcmp(test, hold(jj));
        end
        [compare,index] = max(hold1);
        if (compare == 1)
          objval = num(index);
          goal = [tempgoal1(index), tempgoal2(index), tempgoal3(index)];
        else
          %if the point has not been previously visited, we call the
          %simulation and get the results
          calls = calls + 1;
          [objval, goal] = Evaluate_Simulation_TabuINV(Xnow, targets, g, i, calls,
                t_begin, pts_simulated);
        end
    end

    currP(j) = objval;     % Objective Function Value from Simulation runs for
          Xnow
    currGoal(j,:) = goal;

    %Print all points evaluated to file: "pts_all.dat"
    fid=fopen(pts_all, 'a');
    t_now = clock;
    clocktime = etime(t_now, t_begin);
    fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', test, i,
        calls, clocktime, Xnow(1), Xnow(2), Xnow(3), objval, goal(1), goal(2),
        goal(3));
    fclose(fid);
end

% Based on the solutions contained in N, calculate a vector of corresponding
% current objective function values. Keep the minimum current objective
      function value corresponding
% to a feasible solution
currW_less_m=currW(:,1)<=m(1);
for c=1:v
    currW_less_m=(currW(:,c)<=m(c)).*currW_less_m;
end

currW_less_m;
currW_less_m = ones(4*n,1)-currW_less_m;
currW_less_m = 10000*currW_less_m;
currP=currP+currW_less_m;
```

```
innerloop=5;
End=0;
while innerloop == 5
   [currp,index]=min(currP);
   current_goal = currGoal(index,:);

   % Keep best X from neighborhood. Update tabu list to make a transition
   % back to the old X impossible for a period
   HIST=0;
   for k=1:hist
      if tabu_hist(k,:) == N(index,:)
         HIST=HIST+1;
      end
   end

   if HIST>0
      currP(index)=inf;
   elseif(currp<optZ)
      optX=N(index,:);
      t=0;
      ii=1;
      for ii=1:n
         t=t+sum(optX(ii)-X(ii));
      end
      if t>0
         tabu_add(index)=ttinc;
      elseif t<0
         tabu_decr(index)=ttdec;
      end

      % Update if this new objective function value is better than the
      % existing optimal objective function value.
      X=optX;
      optZ=currp;
      optGoal=current_goal;
      if optZ == 0
         i=i_max+1;
      end
      innerloop=0;
   else
      Xnew=N(index,:);
      t=0;
      ii=1;
      for ii=1:n
         t=t+sum(Xnew(ii)-X(ii));
```

```
      end
      if t>0
         if tabu_add(index)>0
            ADD=tabu_add(index);
            currP(index)=inf;
         else
            tabu_add(index)=ttinc;
            X=Xnew;
            innerloop=3;
         end
      elseif t<0
         if tabu_decr(index)>0
            DECR=tabu_decr(index);
            currP(index)=inf;
         else
            tabu_decr(index)=ttdec;
            X=Xnew;
            innerloop=2;
         end
      end

      if(currp<optZ)
         optZ=currp;
         optX=X;
         optGoal=current_goal;
      end
      End=End+1;
      if End>=5
         innerloop=10;
      end
   end
end

% Increment iteration counter
i=i+1;
solution_history(i,:)=[i,X,currp,current_goal];
solution_store(i)=currp;
if i>= 100
   stop = terminate(solution_store,i);     % Terminate if convergence is better than
          0.99
end
if i==i_max
   stop = 1;
end

% Print Solution history to file
```

```
        fid=fopen(soln_hist, 'a');
        fprintf(fid, '%d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', solution_history(i,:));
        fclose(fid);
    end

    if i>=50;
        if i<75
            complete = 2;
            X = optX;
        else
            complete = 3;
            X = optX;
        end
    end
end

% Record pertinent information and data
ttinc
ttdec
solution_history;
startingpoint

% Record the optimal values
optX
optGoal
optZ
Number_of_iterations = i

% Calculate and print the results
GOAL = satisfied_goals(optGoal, targets, g);
t_end = clock;
clocktime = etime(t_end, t_begin)
run
'Number of Goals Satisfied = ', disp(GOAL)
'Number of Goals Not Satisfied = ', disp(g - GOAL)
'Goals Satisfied = ', disp(goals_satisfied(optGoal, targets, g))
'Percentage of Goals Satisfied = ', disp((GOAL/g)*100)
GS = GOAL;
GNS = g - GOAL;
GSS = goals_satisfied(optGoal, targets, g);
GSP = (GOAL/g)*100;
cputime1 = ['Run Time = ' num2str(clocktime) ' seconds'];
cputime2 = ['Run Time = ' num2str(clocktime/60) ' minutes'];
clocktime2 = clocktime/60;
calls
disp(cputime1)
```

```
    disp(cputime2)

    fid=fopen(results, 'w');
    fprintf(fid, '%s%f %s%f %s%f%s\n %s%d\n %s%d\n %s%f\n %s%f\n %s%f\n
        %s%f\n %s%f\n %s%f\n %s%d\n %s%d\n %s%f\n', 'starting point = [',
        startingpoint(1), ' ', startingpoint(2), ' ', startingpoint(3), ']', '# of iterations = ', i,
        'ROCCA = ', calls, 'Run Time in seconds = ', clocktime, 'Run Time in minutes = ',
        clocktime2, 'OptQ = ', optX(1), 'OptR = ', optX(2), 'OptTBREV = ', optX(3), 'OptZ
        = ', optZ, 'Number of Goals Satisfied = ', GS, 'Number of Goals Not Satisfied = ',
        GNS, 'Percentage of Goals Satisfied = ', GSP);
    fclose(fid);

    fid=fopen(pts_simulated, 'a');
    fprintf(fid, '%s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t %s\t
        %s\t %s\t %s\t %s\t %s\n', 'Pt', 'i', 'calls', 'clocktime', 'Q', 'R', 'TBREV', 'Z', 'G1',
        'G2', 'G3', 'TS', 'TO', 'TR', 'AvgInv', 'Runtime', 'BWT', 'CWT', 'LS', 'BO');
    fclose(fid);

    fid=fopen('C:/TEMP/work/invent/summary_data1.dat', 'a');
    fprintf(fid, '%f\t %f\t %f\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\n', startingpoint(1),
        startingpoint(2), startingpoint(3), i, calls, clocktime, optX(1), optX(2), optX(3),
        optZ, GSP);
    fclose(fid);

    clear all;      %clear variables and statistics
end
```

Evaluate_Simulation_TabuINV

```
function [Z, goal] = Evaluate_Simulation_TabuINV(X, targets, g, i, calls, t_begin,
      pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [Q, R, TBREV]
% These values are sent to AweSim via the file "tabudata.dat"

% The output is then written in AweSim to the file "Core_tabu_output.dat", that is read
% here to get the needed output to calculate the objective function value.

LT = 1;     %This variable is fixed.
TBD = 1;    %This variable is fixed.
```

```
% Converts to only two decimal places
XX(1) = round((X(1) * 100));
XX(2) = round((X(2) * 100));
XX(3) = round((X(3) * 100));
X(1) = XX(1)/100.0;
X(2) = XX(2)/100.0;
X(3) = XX(3)/100.0;

fid=fopen('C:/TEMP/work/tabudata.dat', 'w');
fprintf(fid, '%3f\t %3f\t %3f\t %3ld\t %3ld', X, LT, TBD);
fclose(fid);

% Call the file call_INV.bat to call the simulation model
system('C:/TEMP/work/invent/call_INV.bat');

[TS, TO, TR, AvgInv, Runtime, LS, CWT, BWT, BO] =
      textread('C:/TEMP/work/Core_tabu_output.dat', '%f %f %f %f %f %f %f %f %f');
goal(1) = -1 * ((TS * (65-40)) - (TO * 50) - (TR * 30) - (AvgInv * .004 * Runtime) - (LS
      * 20) - (BO * 10));
goal(2) = CWT;
goal(3) = BWT;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
   end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = Z;
sample1(8) = goal(1);
```

```
sample1(9) = goal(2);
sample1(10) = goal(3);
sample1(11) = TS;
sample1(12) = TO;
sample1(13) = TR;
sample1(14) = AvgInv;
sample1(15) = Runtime;
sample1(16) = BWT;
sample1(17) = CWT;
sample1(18) = LS;
sample1(19) = BO;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
    %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

<u>neighbor_real</u>

```
function N=neighbor_real(X,complete)
n=length(X);
k=0;
if complete == 1
   p1 = 2;
   p2 = 1;
elseif complete == 2
   p1 = 1;
   p2 = 0.5;
else
   p1 = 0.25;
   p2 = 0.1;
end

for j=1:n
   incr=X;
   decr=X;;
   k=k+1;
   incr(j)=incr(j)+p1;
   N(k,:)=incr;
   k=k+1;
   decr(j)=decr(j)-p1;
   N(k,:)=decr;
end
for j=1:n
   incr=X;
   decr=X;
   k=k+1;
```

```
   incr(j)=incr(j)+p2;
   N(k,:)=incr;
   k=k+1;
   decr(j)=decr(j)-p2;
   N(k,:)=decr;
end
```

Z_ObjectiveFunction

```
function Z =  Z_ObjectiveFunction (objective, targets, g)
% Objective calculates the normalized Objective Function, Z, for the problem.
% We are trying to minimize the following equation: sum(Pi*(di+)/Ti).

Z = 0;
for i = 1:g
   P = 100^(g-i);
   Z = Z + ((objective(i) - targets(i))/abs(targets(i)) * P);
end
```

---

call_inv.bat

```
cd\
cd projects\dissert2
execute inv
cd\
cd TEMP\work
```

Logistics Domain

Evaluate_Simulation_TabuLOG

```
function [Z, goal] = Evaluate_Simulation_TabuLOG(X, targets, g, i, calls, t_begin,
      pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [FAIL, ARRIVE, DISC1, DISC2, DISC3]
% These values are sent to AweSim via the file "log_in.dat"

% The output is then written in AweSim to the file "log_out.dat", that is read here to get
% the needed output to calculate the objective function value.

% Converts to only two decimal places, except that X(1) is an integer
XX(1) = X(1);
XX(2) = round((X(2) * 100));
XX(3) = round((X(3) * 100));
XX(4) = round((X(4) * 100));
XX(5) = round((X(5) * 100));
X(2) = XX(2)/100.0;
X(3) = XX(3)/100.0;
X(4) = XX(4)/100.0;
X(5) = XX(5)/100.0;

fid=fopen('C:/TEMP/work/logistic/log_in.dat', 'w');
fprintf(fid, '%3ld\t %f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callLOG.bat to call the simulation model
system('C:/TEMP/work/logistic/callLOG.bat');

[ORD1_AVG, ORD2_AVG, ORD3_AVG, OUT1_AVG, OUT2_AVG, OUT3_AVG,
      OUT1_STD, OUT2_STD, OUT3_STD, UTIL_AVG, UTIL_STD] = textread
      ('C:/TEMP/work/logistic/log_out.dat', '%f %f %f %f %f %f %f %f %f %f %f');
goal(1) = (ORD1_AVG * 25) + (ORD2_AVG * 20) + (ORD3_AVG * 30) +
      (OUT1_AVG * 3) + (OUT2_AVG * 2) + (OUT3_AVG *4);
goal(2) = -1 * (UTIL_AVG/100);

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
```

```
  end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(X(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))
num2str(XX(5))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = X(5);
sample1(9) = Z;
sample1(10) = goal(1);
sample1(11) = goal(2);
sample1(12) = ORD1_AVG;
sample1(13) = ORD2_AVG;
sample1(14) = ORD3_AVG;
sample1(15) = OUT1_AVG;
sample1(16) = OUT2_AVG;
sample1(17) = OUT3_AVG;
sample1(18) = OUT1_STD;
sample1(19) = OUT2_STD;
sample1(20) = OUT3_STD;
sample1(21) = UTIL_AVG;
sample1(22) = UTIL_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
        %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callLOG.bat

```
cd\
cd projects\dissert2
execute logist2
cd\
cd TEMP\work
```

PERT Domain

Evaluate_Simulation_TabuPERT

```
function [Z, goal] = Evaluate_Simulation_TabuPERT(X, targets, g, i, calls, t_begin,
      pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [LOW3, LOW4, LOW6]
% These values are sent to AweSim via the file "pertin.dat"

% The output is then written in AweSim to the files "pertout.dat" and "pertcrit.dat", that
% is read here to get the needed output to calculate the objective function value.

% Converts to only two decimal places
XX(1) = round((X(1) * 100));
XX(2) = round((X(2) * 100));
XX(3) = round((X(3) * 100));
X(1) = XX(1)/100.0;
X(2) = XX(2)/100.0;
X(3) = XX(3)/100.0;

fid=fopen('C:/TEMP/work/pert/pertin.dat', 'w');
fprintf(fid, '%f\t %f\t %f', X);
fclose(fid);

% Call the file callPERT.bat to call the simulation model
system('C:/TEMP/work/pert/callPERT.bat');

[NODE2AVG, NODE2STD, NODE3AVG, NODE3STD, NODE4AVG, NODE4STD,
      NODE5AVG, NODE5STD, PT_AVG, PT_STD] =
      textread('C:/TEMP/work/pert/pertout.dat', '%f %f %f %f %f %f %f %f %f %f');
[CRIT1, CRIT2, CRIT3, CRIT4, CRIT5, CRIT6, CRIT7, CRIT8, CRIT9 ]=
      textread('C:/TEMP/work/pert/pertcrit.dat', '%f %f %f %f %f %f %f %f %f');
goal(1) = (CRIT1 * 5 * 2) + (CRIT2 * 3 * 2) + (CRIT3 * 2 * 1) + (CRIT4 * 7 * 1) +
      (CRIT5 * 4 * 2) + (CRIT6 * 3 * 1) + (CRIT7 * 2 * 2) + (CRIT8 * 5 * 1) + (CRIT9 *
      4 * 2);
goal(2) = PT_AVG;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
      goal(k) = targets(k);
```

```
    end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = Z;
sample1(8) = goal(1);
sample1(9) = goal(2);
sample1(10) = CRIT1;
sample1(11) = CRIT2;
sample1(12) = CRIT3;
sample1(13) = CRIT4;
sample1(14) = CRIT5;
sample1(15) = CRIT6;
sample1(16) = CRIT7;
sample1(17) = CRIT8;
sample1(18) = CRIT9;
sample1(19) = PT_AVG;
sample1(20) = PT_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
        %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callPERT.bat

```
cd\
cd projects\dissert2
execute pert
cd\
cd TEMP\work
```

Production Domain

Evaluate_Simulation_TabuPROD

```
function [Z, goal] = Evaluate_Simulation_TabuPROD(X, targets, g, i, calls, t_begin,
    pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [LOAD, INSPECT, UNLOAD, TRANSPORT1, TRANSPORT2,
    TRANSPORT3]
% These values are sent to AweSim via the file "prod_in.dat"

% The output is then written in AweSim to the file "prod_out.dat", that is read here to get
% the needed output to calculate the objective function value.

% Converts to only two decimal places
XX(1) = round((X(1) * 100));
XX(2) = round((X(2) * 100));
XX(3) = round((X(3) * 100));
XX(4) = round((X(4) * 100));
XX(5) = round((X(5) * 100));
XX(6) = round((X(6) * 100));
X(1) = XX(1)/100.0;
X(2) = XX(2)/100.0;
X(3) = XX(3)/100.0;
X(4) = XX(4)/100.0;
X(5) = XX(5)/100.0;
X(6) = XX(6)/100.0;

fid=fopen('C:/TEMP/work/product/prod_in.dat', 'w');
fprintf(fid, '%f\t %f\t %f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callPROD.bat to call the simulation model
system('C:/TEMP/work/product/callPROD.bat');

[TPUT, TIS_AVG, TIS_STD, PU_AVG, PU_STD, CU_AVG, CU_STD] =
    textread('C:/TEMP/work/product/prod_out.dat', '%f %f %f %f %f %f %f');
goal(1) = -1 * (100 * TPUT - (1000 * (1/TIS_AVG)));
goal(2) = -1 * (PU_AVG/6 + CU_AVG/2);

epsilon = 0.00005;
for k = 1:g
```

```
    if goal(k) <= targets(k) + epsilon
        goal(k) = targets(k);
    end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))
num2str(XX(5)) num2str(XX(6))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = X(5);
sample1(9) = X(6);
sample1(10) = Z;
sample1(11) = goal(1);
sample1(12) = goal(2);
sample1(13) = TPUT;
sample1(14) = TIS_AVG;
sample1(15) = TIS_STD;
sample1(16) = PU_AVG;
sample1(17) = PU_STD;
sample1(18) = CU_AVG;
sample1(19) = CU_STD;
fprintf(fid, '%s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t
        %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callPROD.bat

```
cd\
cd projects\dissert2
execute product
cd\
cd TEMP\work
```

Reliability Domain

Evaluate_Simulation_TabuRELI

```
function [Z, goal] = Evaluate_Simulation_TabuRELI(X, targets, g, i, calls, t_begin,
      pts_simulated)

% This file calls AweSim given the starting condition, to run the simulation
% and generate the required output.

% The input to the simulation is the current X value, where
% X = [Fail1, Fail2, Repair1, Repair2]
% These values are sent to AweSim via the file "reli_in.dat"

% The output is then written in AweSim to the file "reli_out.dat", that is read here to get
% the needed output to calculate the objective function value.

% Converts to only two decimal places
XX(1) = round((X(1) * 100));
XX(2) = round((X(2) * 100));
XX(3) = round((X(3) * 100));
XX(4) = round((X(4) * 100));
X(1) = XX(1)/100.0;
X(2) = XX(2)/100.0;
X(3) = XX(3)/100.0;
X(4) = XX(4)/100.0;

fid=fopen('C:/TEMP/work/reliable/reli_in.dat', 'w');
fprintf(fid, '%f\t %f\t %f\t %f', X);
fclose(fid);

% Call the file callRELI.bat to call the simulation model
system('C:/TEMP/work/reliable/callRELI.bat');

[BD1_NUM, BD2_NUM, SYSBD_N, BD1_AVG, BD2_AVG, SYSBD_A, BD1_STD,
      BD2_STD, SYSBD_S, DT_AVG, DT_STD] =
      textread('C:/TEMP/work/reliable/reli_out.dat',
      '%f %f %f %f %f %f %f %f %f %f %f');
goal(1) = (BD1_AVG * X(1) * 1) + (BD2_AVG * X(2) * 2) + (SYSBD_A * 10 *
      ((1/X(3) * 5) + (1/X(4) * 6)));
goal(2) = DT_AVG;
goal(3) = BD1_AVG + BD2_AVG + SYSBD_A;

epsilon = 0.00005;
for k = 1:g
   if goal(k) <= targets(k) + epsilon
```

```
        goal(k) = targets(k);
    end
end

goal;
Z = Z_ObjectiveFunction(goal, targets, g);

% Write the results of each call to the simulation to the file
% pts_simulated, this will allow us to check to see if a point has
% previously been simulated and thus prevent it from being re-simulated
fid=fopen(pts_simulated, 'a');
sample = [num2str(XX(1)) num2str(XX(2)) num2str(XX(3)) num2str(XX(4))];
sample1(1) = i;
sample1(2) = calls;
t_now = clock;
sample1(3) = etime(t_now, t_begin);
sample1(4) = X(1);
sample1(5) = X(2);
sample1(6) = X(3);
sample1(7) = X(4);
sample1(8) = Z;
sample1(9) = goal(1);
sample1(10) = goal(2);
sample1(11) = goal(3);
sample1(12) = BD1_NUM;
sample1(13) = BD2_NUM;
sample1(14) = SYSBD_N;
sample1(15) = BD1_AVG;
sample1(16) = BD2_AVG;
sample1(17) = SYSBD_A;
sample1(18) = BD1_STD;
sample1(19) = BD2_STD;
sample1(20) = SYSBD_S;
sample1(21) = DT_AVG;
sample1(22) = DT_STD;
fprintf(fid, %s\t %d\t %d\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\
        t %f\t %f\t %f\t %f\t %f\t %f\t %f\n', sample, sample1);
fclose(fid);
```

---

callRELI.bat

```
cd\
cd projects\dissert2
execute reliable
cd\
cd TEMP\work
```

APPENDIX B

AWESIM CODES FOR TEST BED PROBLEMS

AweSim Components

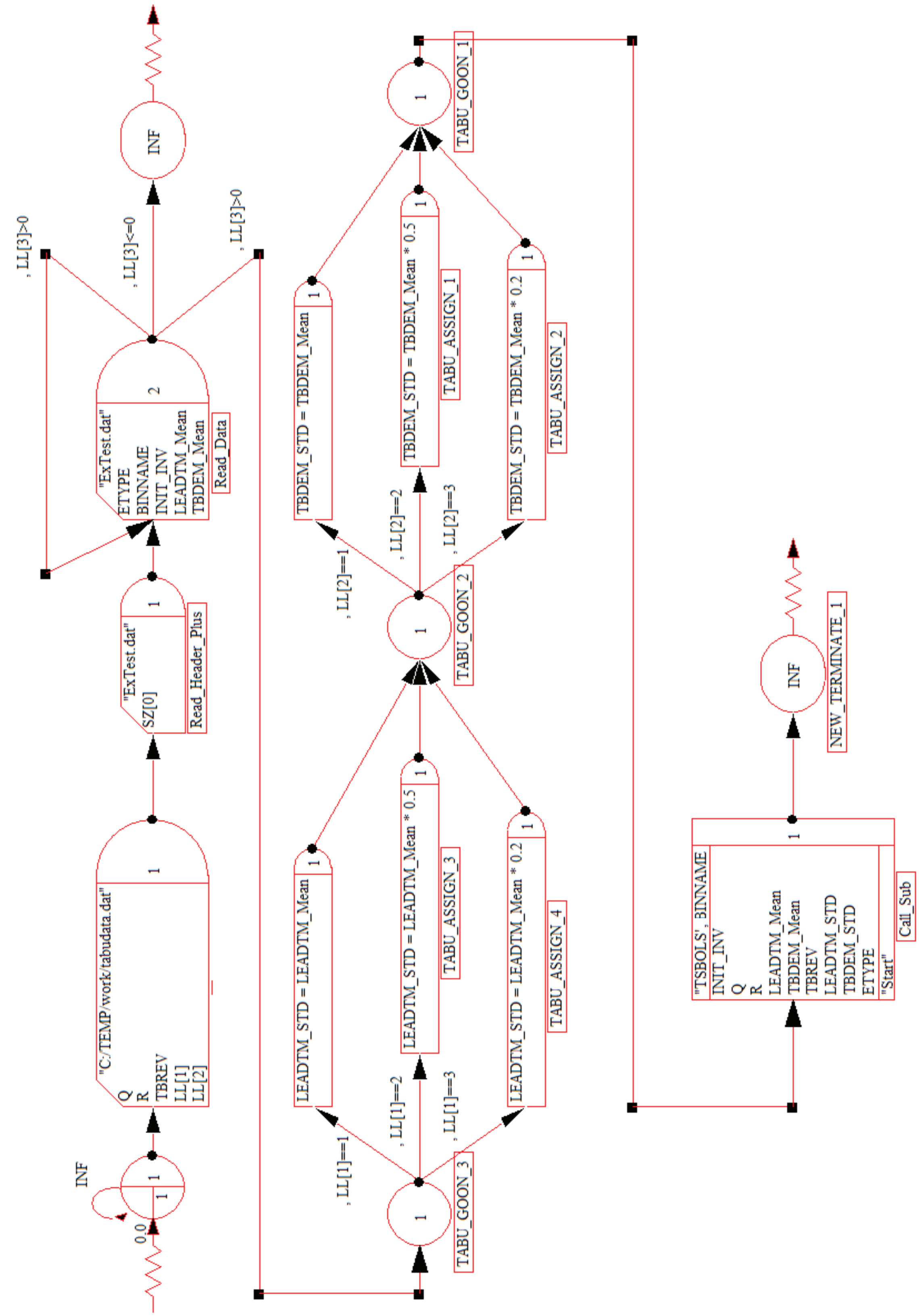
The AweSim network models, control statements, network statements, and all other components and files codes used to simulate the test bed problems for the LNM-SO and the TS-SO methods are presented below. The network and subnetwork models were divided into sections or components and modified to make the models more viewable. The only difference between the AweSim information presented here and the information used to simulate the GA-SO method was the specific file names used. Otherwise, they were the same.

Inventory Domain

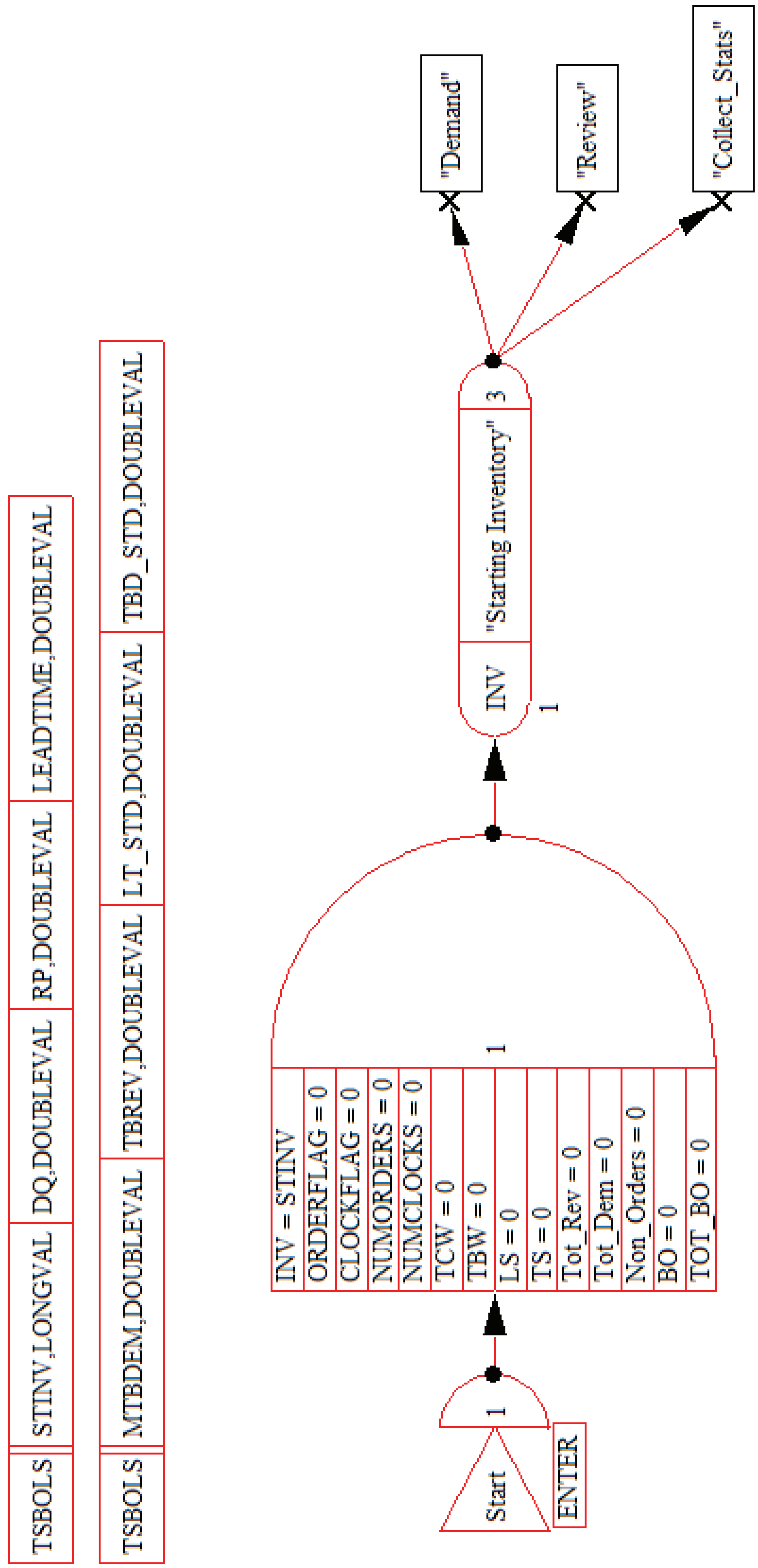
Input File: ”Extest.dat”

```
BIN_NUMBER---BIN_NAME---INITIAL_LEVEL---LEAD_TIME---DEMAN_FREQ
     1          Bin_1          72             3            .2
```

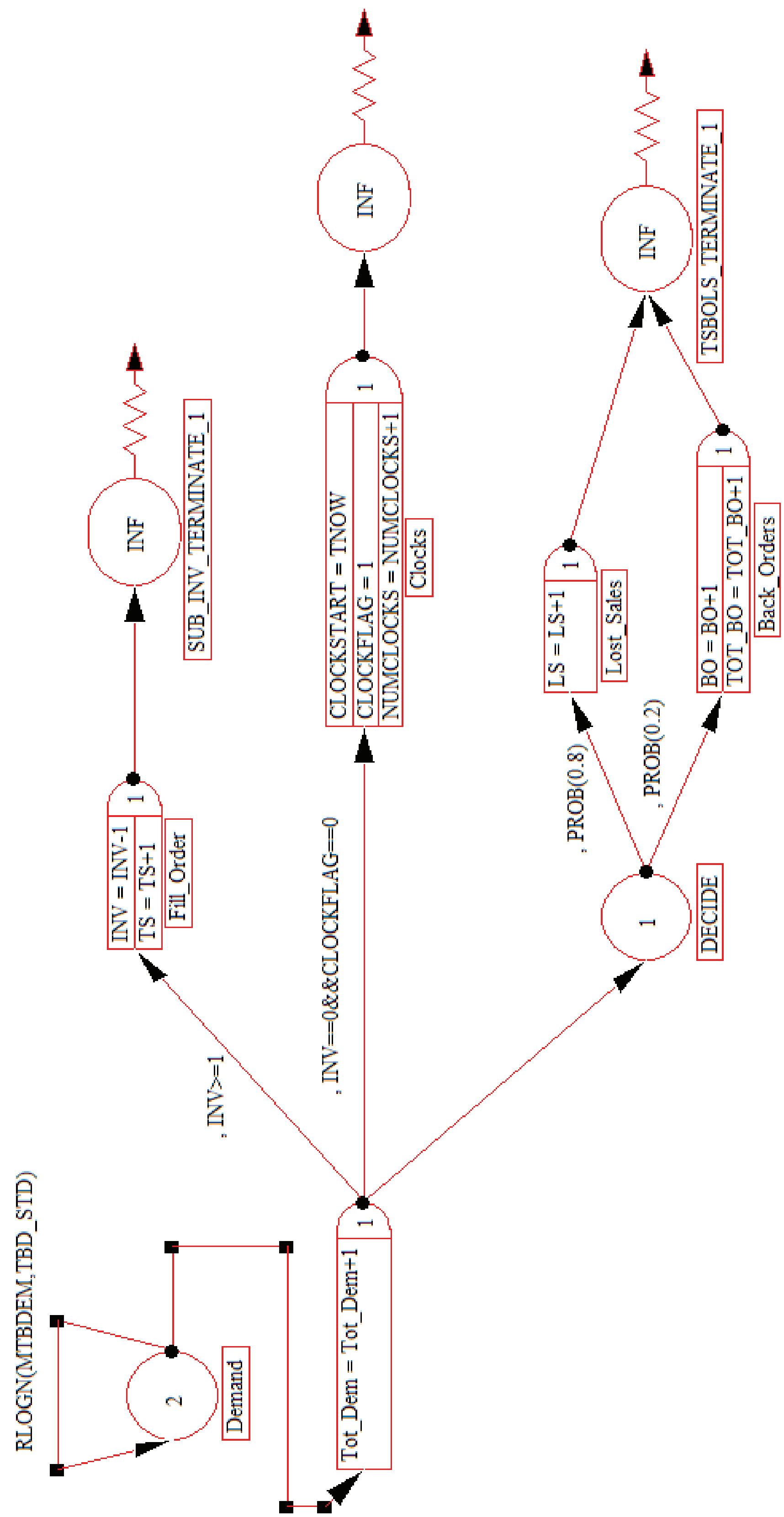
Network Model



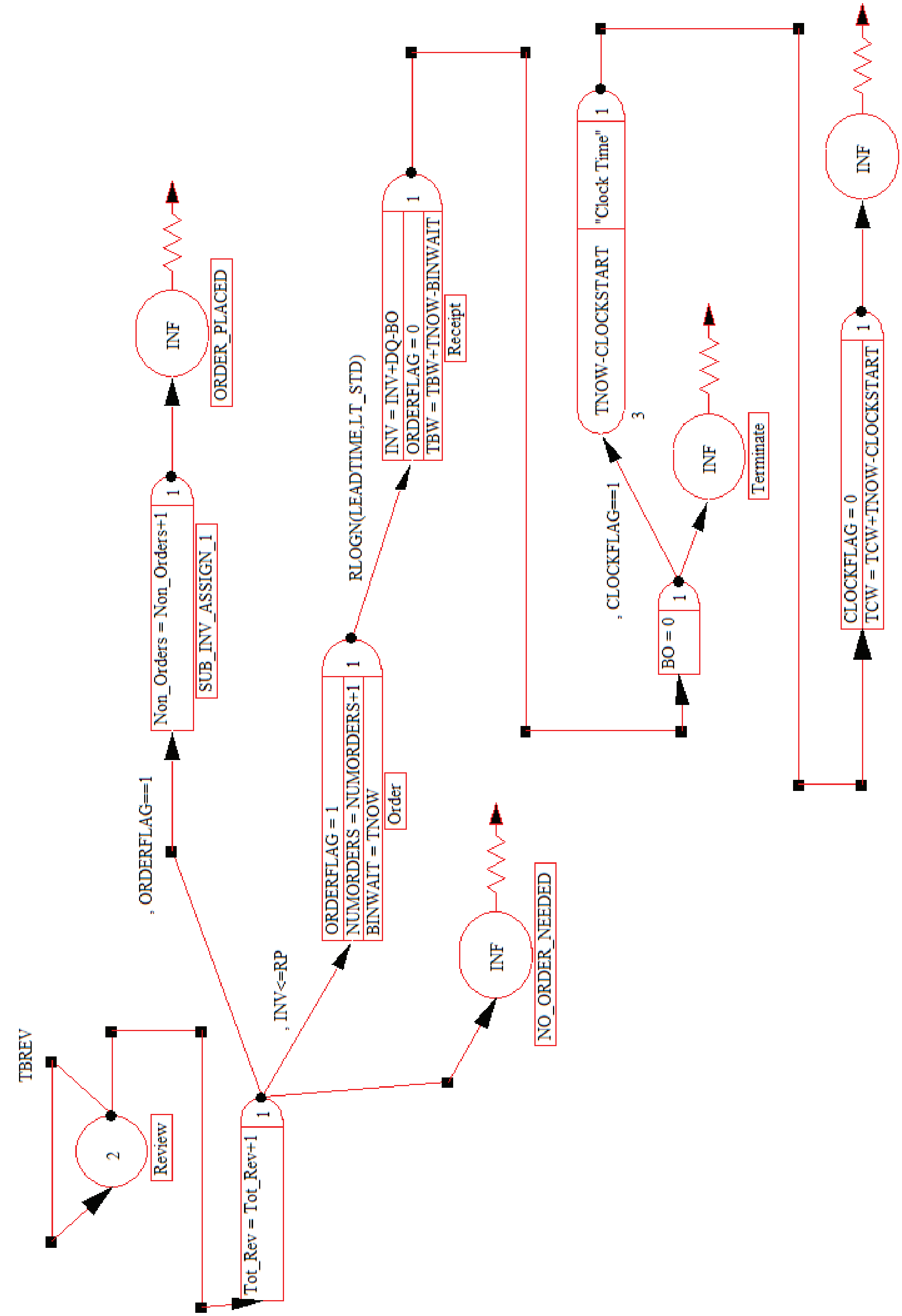
Subnetwork Model: Initial Setup



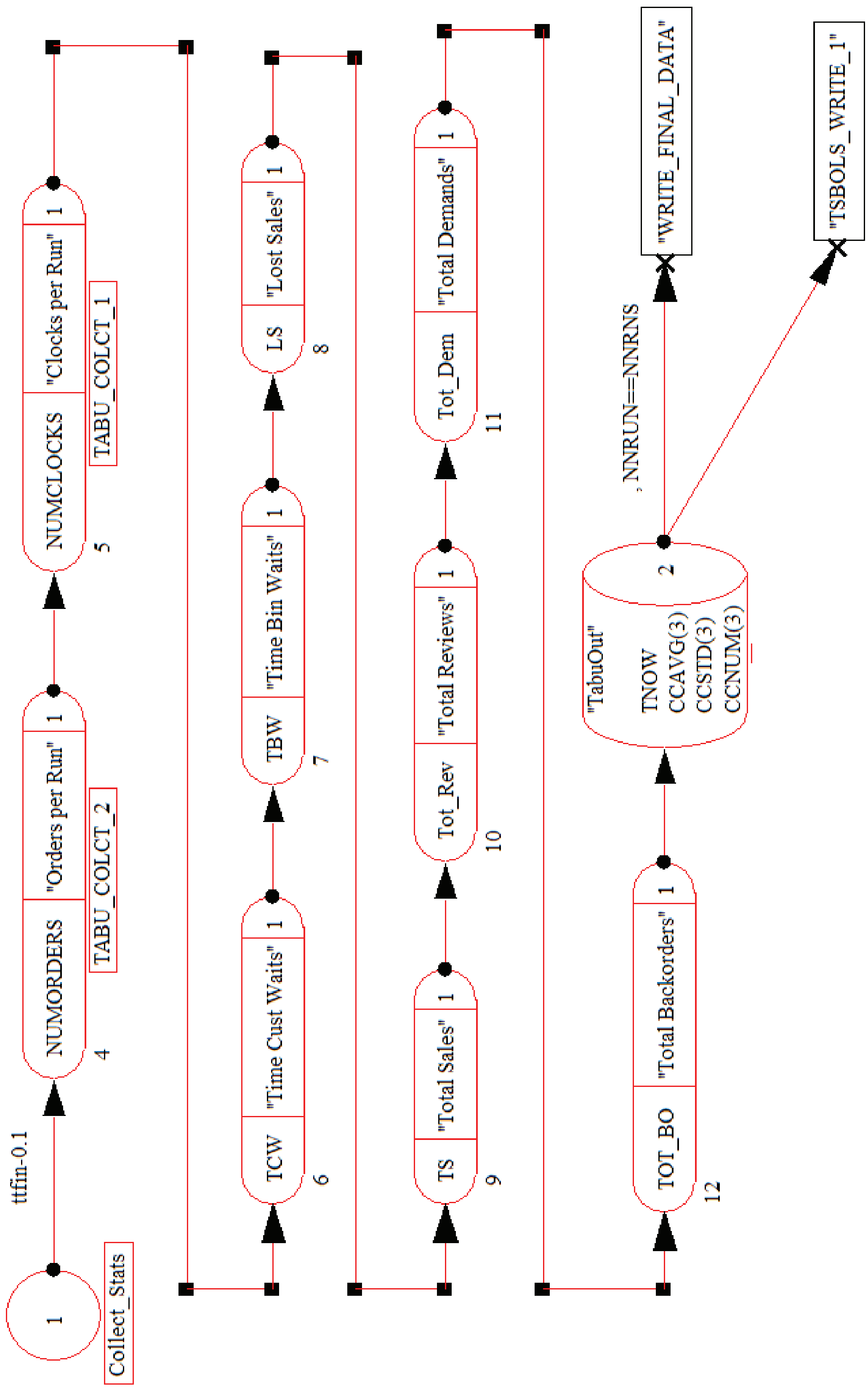
Subnetwork Model: Demand Component



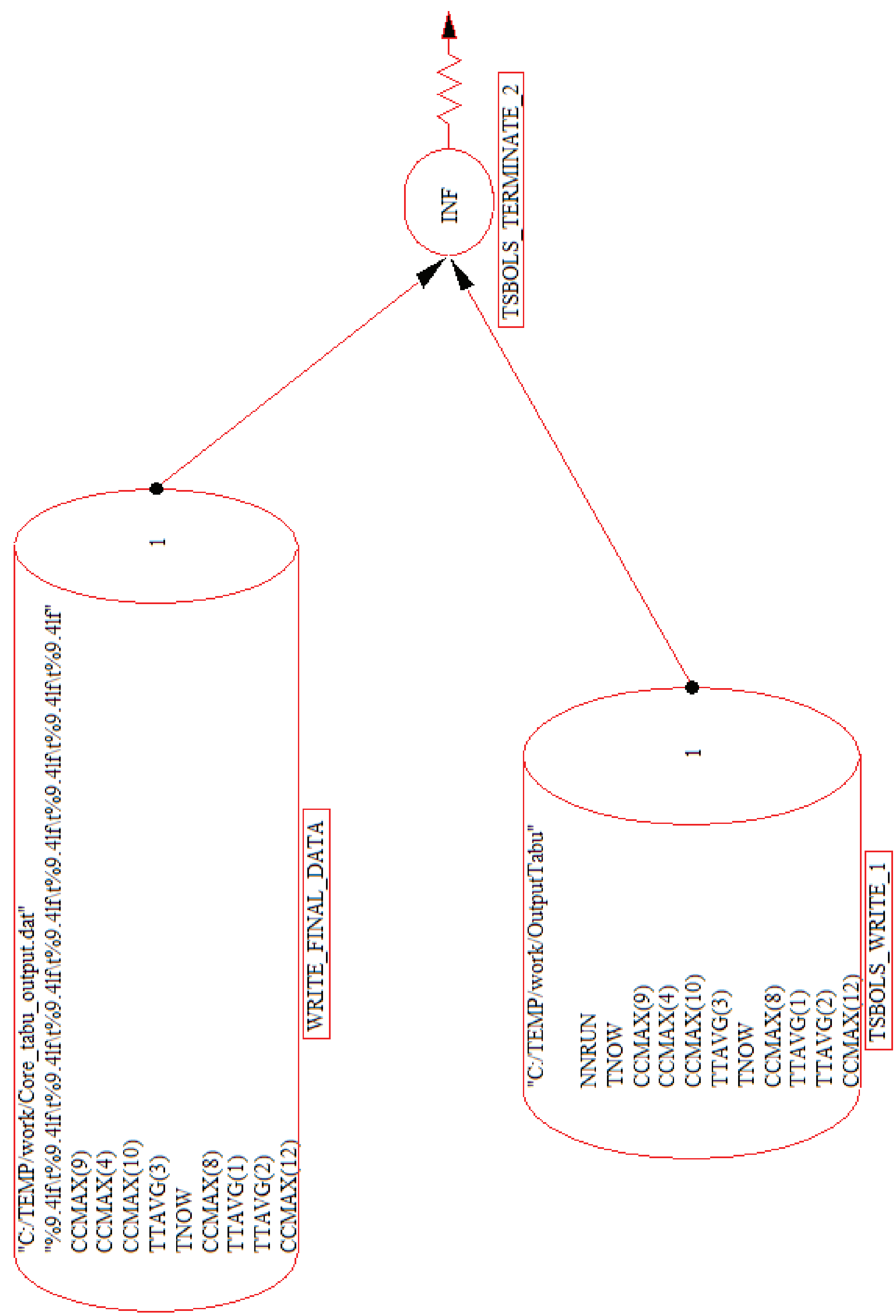
Subnetwork Model: Review Component



Subnetwork Model: Collection of Statistics (1 of 2)



Subnetwork Model: Collection of Statistics (2 of 2)



Control Statements:

```
GEN,"Glenn Kuriger","INVENTORY with SUBNETWORK",,400,YES,YES;
EQUIVALENCE,{{BINNAME,STRIB[0]},{INIT_INV,LTRIB[0]},{Q,ATRIB[5]},
        {R,ATRIB[6]},{LEADTM_Mean,ATRIB[0]},{TBDEM_Mean,ATRIB[1]},
        {TBREV,ATRIB[2]},{SIM_TIME,XX[1]},{LEADTM_STD,ATRIB[3]},
        {TBDEM_STD,ATRIB[4]}};
INITIALIZE,0.0,108,NO,,NO;
LIMITS,6,6,1,6,6,1,500;
MONTR,CLEAR,4;
REPORT,90,YES,YES,EVERY(400);
NET;
FIN;
```

Network Statements:

```
;DBF file created with Version 4
    CREATE,INF,0.0,,1,1,,,,,,{50,60};
    ACTIVITY,,,,,,,,,{1,3,,,};
    READ,"C:/TEMP/work/tabudata.dat",YES,,,{Q,R,TBREV,LL[1],LL[2]},1,,,,,
      {80,60};
    ACTIVITY,,,,,,,,,{3,5,,,};
Read_Header_Plus: READ,"ExTest.dat",YES,LL[3],,{SZ[0]},1,,,,,{220,60};
      ACTIVITY,,,,,,,,,{5,7,,,};
Read_Data: READ,"ExTest.dat",YES,LL[3],,{ETYPE,BINNAME,INIT_INV,
      LEADTM_Mean,TBDEM_Mean},2,,,,,{300,60};
    ACTIVITY,,,LL[3]>0,"Read_Data",,,,,,{7,7,,,400,20,280,20};
    ACTIVITY,,,LL[3]<=0,,,,,,,{7,11,,,};
    ACTIVITY,,,LL[3]>0,"TABU_GOON_3",,,,,,{7,12,,,400,100,10,100,10,160};
    TERMINATE,INF,,,,,,,,,{440,60};
TABU_GOON_3: GOON,1,,,,,,,,,{40,160};
    ACTIVITY,,,LL[1]==1,,,,,,,{12,16,,,};
    ACTIVITY,,,LL[1]==2,"TABU_ASSIGN_3",,,,,,{12,33,,,};
    ACTIVITY,,,LL[1]==3,"TABU_ASSIGN_4",,,,,,{12,35,,,};
    ASSIGN,{{LEADTM_STD,LEADTM_Mean}},1,,,,,,,,{80,120};
    ACTIVITY,,,,,,,,,{16,18,,,};
TABU_GOON_2: GOON,1,,,,,,,,,{260,160};
    ACTIVITY,,,LL[2]==1,,,,,,,{18,22,,,};
    ACTIVITY,,,LL[2]==2,"TABU_ASSIGN_1",,,,,,{18,29,,,};
    ACTIVITY,,,LL[2]==3,"TABU_ASSIGN_2",,,,,,{18,31,,,};
    ASSIGN,{{TBDEM_STD,TBDEM_Mean}},1,,,,,,,,{300,120};
    ACTIVITY,,,,,,,,,{22,24,,,};
TABU_GOON_1: GOON,1,,,,,,,,,{460,160};
    ACTIVITY,,,,,,,,,{24,26,,,480,160,480,240,80,240,80,300};
```

```
Call_Sub: CALLVSN,"TSBOLS",BINNAME,{INIT_INV,Q,R,LEADTM_Mean,
        TBDEM_Mean,TBREV,LEADTM_STD,TBDEM_STD,ETYPE},1,"Start",,,,,,
        {120,300};
     ACTIVITY,,,,,,,,,{26,28,,,};
NEW_TERMINATE_1: TERMINATE,INF,,,,,,,,,{240,300};
TABU_ASSIGN_1: ASSIGN,{{TBDEM_STD,TBDEM_Mean *
        0.5}},1,,,,,,,,{320,160};
     ACTIVITY,,,,"TABU_GOON_1",,,,,,{29,24,,,};
TABU_ASSIGN_2: ASSIGN,{{TBDEM_STD,TBDEM_Mean *
        0.2}},1,,,,,,,,{300,200};
     ACTIVITY,,,,"TABU_GOON_1",,,,,,{31,24,,,};
TABU_ASSIGN_3: ASSIGN,{{LEADTM_STD,LEADTM_Mean *
        0.5}},1,,,,,,,,{100,160};
     ACTIVITY,,,,"TABU_GOON_2",,,,,,{33,18,,,};
TABU_ASSIGN_4: ASSIGN,{{LEADTM_STD,LEADTM_Mean *
        0.2}},1,,,,,,,,{80,200};
     ACTIVITY,,,,"TABU_GOON_2",,,,,,{35,18,,,};
```

Subnetwork Statements:

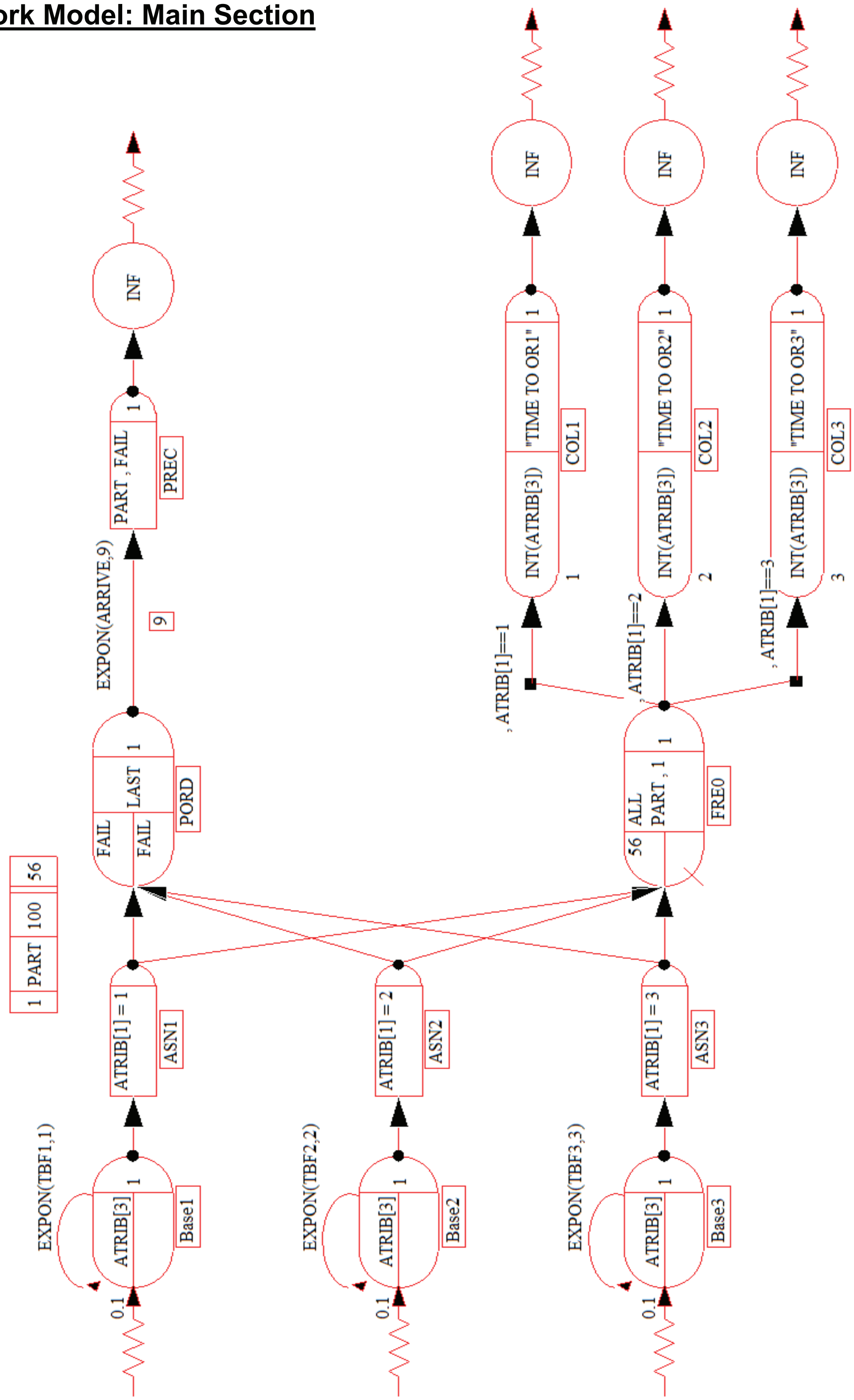
```
;DBF file created with Version 4
     VSN,TSBOLS,{{STINV,LONGVAL,Starting InventoryLevel},
        {DQ,DOUBLEVAL,Demand Quantity},{RP,DOUBLEVAL,Re-Order
        Point},{LEADTIME,DOUBLEVAL,Mean Lead time}},,,,,,,,{60,20};
     VSN,TSBOLS,{{MTBDEM,DOUBLEVAL,Mean time between
        demand},{TBREV,DOUBLEVAL,Time between
        reviews},{LT_STD,DOUBLEVAL,Standard Deviation of Lead
        Time},{TBD_STD,DOUBLEVAL,Standard Deviation of time between
        demands}},,,,,,,,{60,40};
     EQUIVALENCE,{{INV,XXINST[1]},{TCW,XXINST[2]},{TBW,XXINST[3]},
        {ORDERFLAG,LLINST[1]},{CLOCKFLAG,LLINST[2]},{NUMORDERS,
        LLINST[3]},{NUMCLOCKS,LLINST[4]},{LS,LLINST[5]},{TS,LLINST[6]},
        {Tot_Rev,LLINST[7]},{Tot_Dem,LLINST[8]},{Non_Orders,LLINST[9]},
        {CLOCKSTART,XXINST[4]},{BINWAIT,XXINST[5]},{BO,LLINST[10]},
        {TOT_BO,LLINST[11]}},,,,,,,,,{0,40};
     LIMITSVSN,5,11,-1,-1,-1,-1,,,,,{0,40};
     TIMST,1,CLOCKFLAG,"Customer Wait Time",0,0.0,1.0,,,,,{0,40};
     TIMST,2,ORDERFLAG,"Bin Wait Time",0,0.0,1.0,,,,,{0,40};
     TIMST,3,INV,"Average Inventory",0,0.0,1.0,,,,,{0,40};
ENTER: ENTERVSN,Start,1,,,,,,,,{90,120};
     ACTIVITY,,,,,,,,,{8,10,,,};
     ASSIGN,{{INV,STINV},{ORDERFLAG,0},{CLOCKFLAG,0},{NUMORDERS,
        0},{NUMCLOCKS,0},{TCW,0},{TBW,0},{LS,0},{TS,0},{Tot_Rev,0},
        {Tot_Dem,0},{Non_Orders,0},{BO,0},{TOT_BO,0}},1,,,,,,,,{120,120};
     ACTIVITY,,,,,,,,,{10,12,,,};
```

```
    COLCT,1,INV,"Starting Inventory",,,,3,,,,{250,120};
    ACTIVITY,,,,"Demand",,,,,,{12,-1,,,360,110};
    ACTIVITY,,,,"Review",,,,,,{12,-1,,,360,140};
    ACTIVITY,,,,"Collect_Stats",,,,,,{12,-1,,,360,170};
Collect_Stats: GOON,1,,,,,,,,,{101,796};
    ACTIVITY,,ttfin-0.1,,"TABU_COLCT_2",,,,,,{16,18,,,};
TABU_COLCT_2: COLCT,4,NUMORDERS,"Orders per Run",,,,1,,,,{164,796};
    ACTIVITY,,,,"TABU_COLCT_1",,,,,,{18,20,,,};
TABU_COLCT_1: COLCT,5,NUMCLOCKS,"Clocks per Run",,,,1,,,,{297,796};
    ACTIVITY,,,,,,,,,{20,22,,,427,796,427,830,101,830,101,852};
    COLCT,6,TCW,"Time Cust Waits",,,,1,,,,{131,852};
    ACTIVITY,,,,,,,,,{22,24,,,};
    COLCT,7,TBW,"Time Bin Waits",,,,1,,,,{241,852};
    ACTIVITY,,,,,,,,,{24,26,,,};
    COLCT,8,LS,"Lost Sales",,,,1,,,,{349,852};
    ACTIVITY,,,,,,,,,{26,28,,,424,852,424,873,101,873,101,896};
    COLCT,9,TS,"Total Sales",,,,1,,,,{131,896};
    ACTIVITY,,,,,,,,,{28,30,,,};
    COLCT,10,Tot_Rev,"Total Reviews",,,,1,,,,{221,896};
    ACTIVITY,,,,,,,,,{30,32,,,};
    COLCT,11,Tot_Dem,"Total Demands",,,,1,,,,{331,896};
    ACTIVITY,,,,,,,,,{32,34,,,431,896,431,921,101,921,101,952};
    COLCT,12,TOT_BO,"Total Backorders",,,,1,,,,{129,952};
    ACTIVITY,,,,,,,,,{34,36,,,};
    WRITE,"TabuOut",NO,,{TNOW,CCAVG(3),CCSTD(3),CCNUM(3)},2,,,,,,{252,
      952};
    ACTIVITY,,,NNRUN==NNRNS,"WRITE_FINAL_DATA",,,,,,{36,-1,,,370,952};
    ACTIVITY,,,,"TSBOLS_WRITE_1",,,,,,{36,-1,,,374,996};
WRITE_FINAL_DATA: WRITE,"C:/TEMP/work/Core_tabu_output.dat",NO,
      "%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf\t%9.4lf",
      {CCMAX(9),CCMAX(4),CCMAX(10),TTAVG(3),TNOW,CCMAX(8),
      TTAVG(1),TTAVG(2),CCMAX(12)},1,,,,,,{103,1074};
    ACTIVITY,,,,,,,,,{39,41,,,};
TSBOLS_TERMINATE_2: TERMINATE,INF,,,,,,,,,{381,1152};
TSBOLS_WRITE_1: WRITE,"C:/TEMP/work/OutputTabu",NO,,{NNRUN,TNOW,
      CCMAX(9),CCMAX(4),CCMAX(10),TTAVG(3),TNOW,CCMAX(8),
      TTAVG(1),TTAVG(2),CCMAX(12)},1,,,,,,{105,1206};
    ACTIVITY,,,,"TSBOLS_TERMINATE_2",,,,,,{42,41,,,};
SUB_INV_TERMINATE_1: TERMINATE,INF,,,,,,,,,{310,230};
Fill_Order: ASSIGN,{{INV,INV-1},{TS,TS+1}},1,,,,,,,,{200,230};
    ACTIVITY,,,,"SUB_INV_TERMINATE_1",,,,,,{45,44,,,};
Demand: GOON,2,,,,,,,,,{80,240};
    ACTIVITY,,RLOGN(MTBDEM,TBD_STD),,"Demand",,,,,,{47,47,,,100,210,60,
      210};
    ACTIVITY,,,,,,,,,{47,50,,,120,240,120,270,50,270,50,280};
    ASSIGN,{{Tot_Dem,Tot_Dem+1}},1,,,,,,,,{60,290};
```

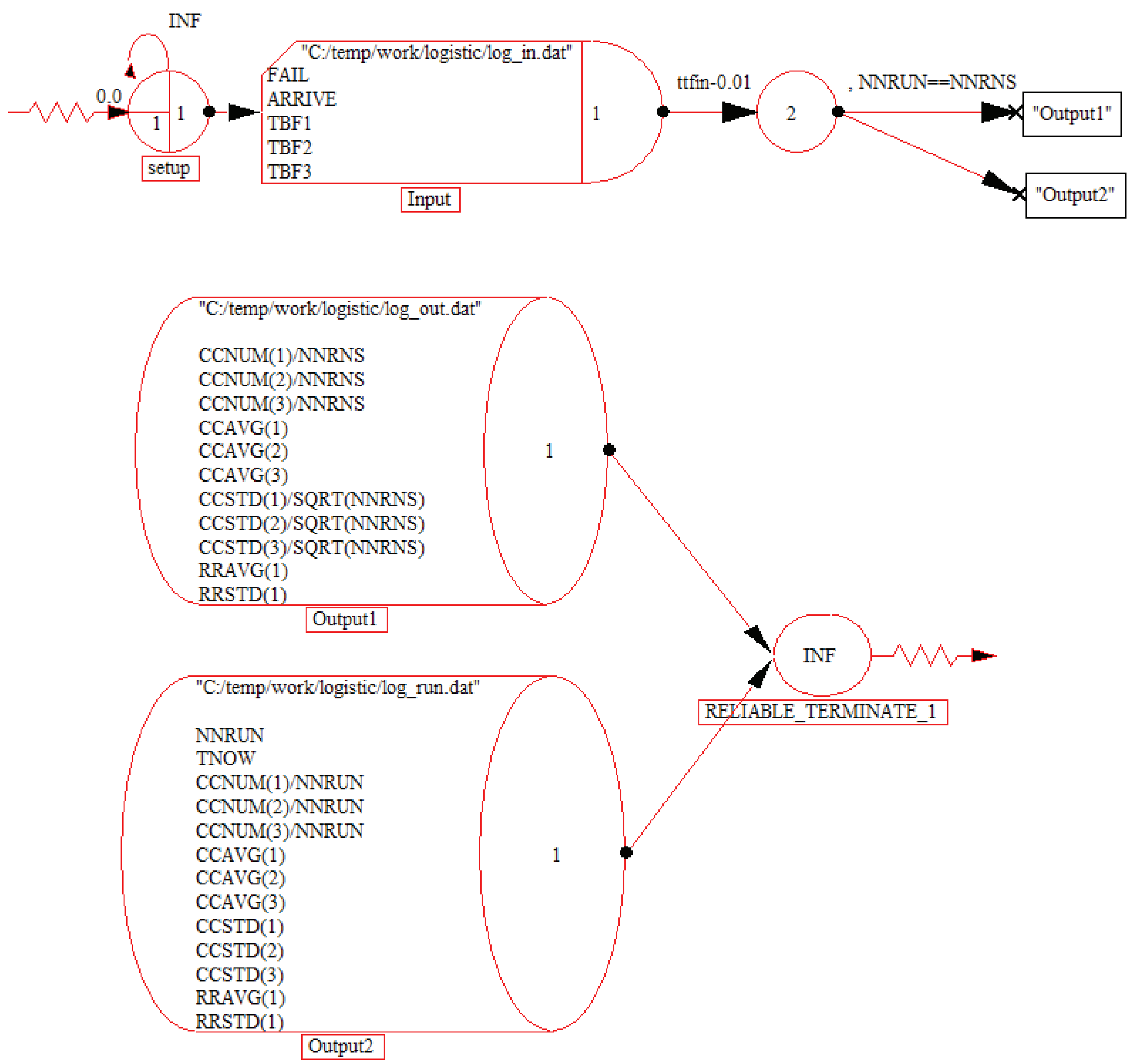
```
      ACTIVITY,,,INV>=1,"Fill_Order",,,,,,{50,45,,,};
      ACTIVITY,,,INV==0&&CLOCKFLAG==0,,,,,,,{50,54,,,};
      ACTIVITY,,,,"DECIDE",,,,,,{50,57,,,};
Clocks: ASSIGN,{{CLOCKSTART,TNOW},{CLOCKFLAG,1},{NUMCLOCKS,
         NUMCLOCKS+1}},1,,,,,,,,,{260,290};
      ACTIVITY,,,,,,,,,,{54,56,,,};
      TERMINATE,INF,,,,,,,,,,{400,290};
DECIDE: GOON,1,,,,,,,,,,{209,364};
      ACTIVITY,,,PROB(0.8),,,,,,,{57,60,,,};
      ACTIVITY,,,PROB(0.2),"Back_Orders",,,,,,{57,63,,,};
Lost_Sales: ASSIGN,{{LS,LS+1}},1,,,,,,,,,{269,344};
      ACTIVITY,,,,,,,,,,{60,62,,,};
TSBOLS_TERMINATE_1: TERMINATE,INF,,,,,,,,,,{389,364};
Back_Orders: ASSIGN,{{BO,BO+1},{TOT_BO,TOT_BO+1}},1,,,,,,,,,{269,384};
      ACTIVITY,,,,"TSBOLS_TERMINATE_1",,,,,,{63,62,,,};
Review: GOON,2,,,,,,,,,,{70,480};
      ACTIVITY,,TBREV,,"Review",,,,,,{65,65,,,100,460,40,460};
      ACTIVITY,,,,,,,,,,{65,68,,,110,480,110,510,18,510};
      ASSIGN,{{Tot_Rev,Tot_Rev+1}},1,,,,,,,,,{20,530};
      ACTIVITY,,,ORDERFLAG==1,"SUB_INV_ASSIGN_1",,,,,,{68,72,,,170,500};
      ACTIVITY,,,INV<=RP,"Order",,,,,,{68,75,,,};
      ACTIVITY,,,,"NO_ORDER_NEEDED",,,,,,{68,88,,,93,592};
SUB_INV_ASSIGN_1: ASSIGN,{{Non_Orders,Non_Orders+1}},1,,,,,,,,,{210,500};
      ACTIVITY,,,,"ORDER_PLACED",,,,,,{72,74,,,};
ORDER_PLACED: TERMINATE,INF,,,,,,,,,,{340,500};
Order: ASSIGN,{{ORDERFLAG,1},{NUMORDERS,NUMORDERS+1},
         {BINWAIT,TNOW}},1,,,,,,,,,{140,560};
      ACTIVITY,,RLOGN(LEADTIME,LT_STD),,,,,,,,{75,77,,,};
Receipt: ASSIGN,{{INV,INV+DQ-BO},{ORDERFLAG,0},{TBW,TBW+TNOW-
         BINWAIT}},1,,,,,,,,,{300,580};
      ACTIVITY,,,,,,,,,,{77,79,,,436,580,436,618,210,618,210,670};
      ASSIGN,{{BO,0}},1,,,,,,,,,{229,669};
      ACTIVITY,,,CLOCKFLAG==1,,,,,,,{79,82,,,};
      ACTIVITY,,,,"Terminate",,,,,,{79,87,,,};
      COLCT,3,TNOW-CLOCKSTART,"Clock Time",,,,1,,,,{317,643};
      ACTIVITY,,,,,,,,,,{82,84,,,444,643,444,709,192,709,192,730};
      ASSIGN,{{CLOCKFLAG,0},{TCW,TCW+TNOW-
         CLOCKSTART}},1,,,,,,,,,{244,730};
      ACTIVITY,,,,,,,,,,{84,86,,,};
      TERMINATE,INF,,,,,,,,,,{399,730};
Terminate: TERMINATE,INF,,,,,,,,,,{299,679};
NO_ORDER_NEEDED: TERMINATE,INF,,,,,,,,,,{133,608};
```

Logistics Domain

Network Model: Main Section



Network Model: Data Section



Control Statements

```
GEN,"Glenn Kuriger","Logistics Network",,400,YES,YES;
REPORT,,YES,YES,EVERY(400);
LIMITS,10,10,,10,10;
EQUIVALENCE,{{FAIL,XX[4]},{ARRIVE,XX[5]},{TBF1,XX[1]},{TBF2,XX[2]},
        {TBF3,XX[3]}};
INITIALIZE,0.0,108,NO,,NO;
MONTR,CLEAR,4;
NET;
FIN;
```

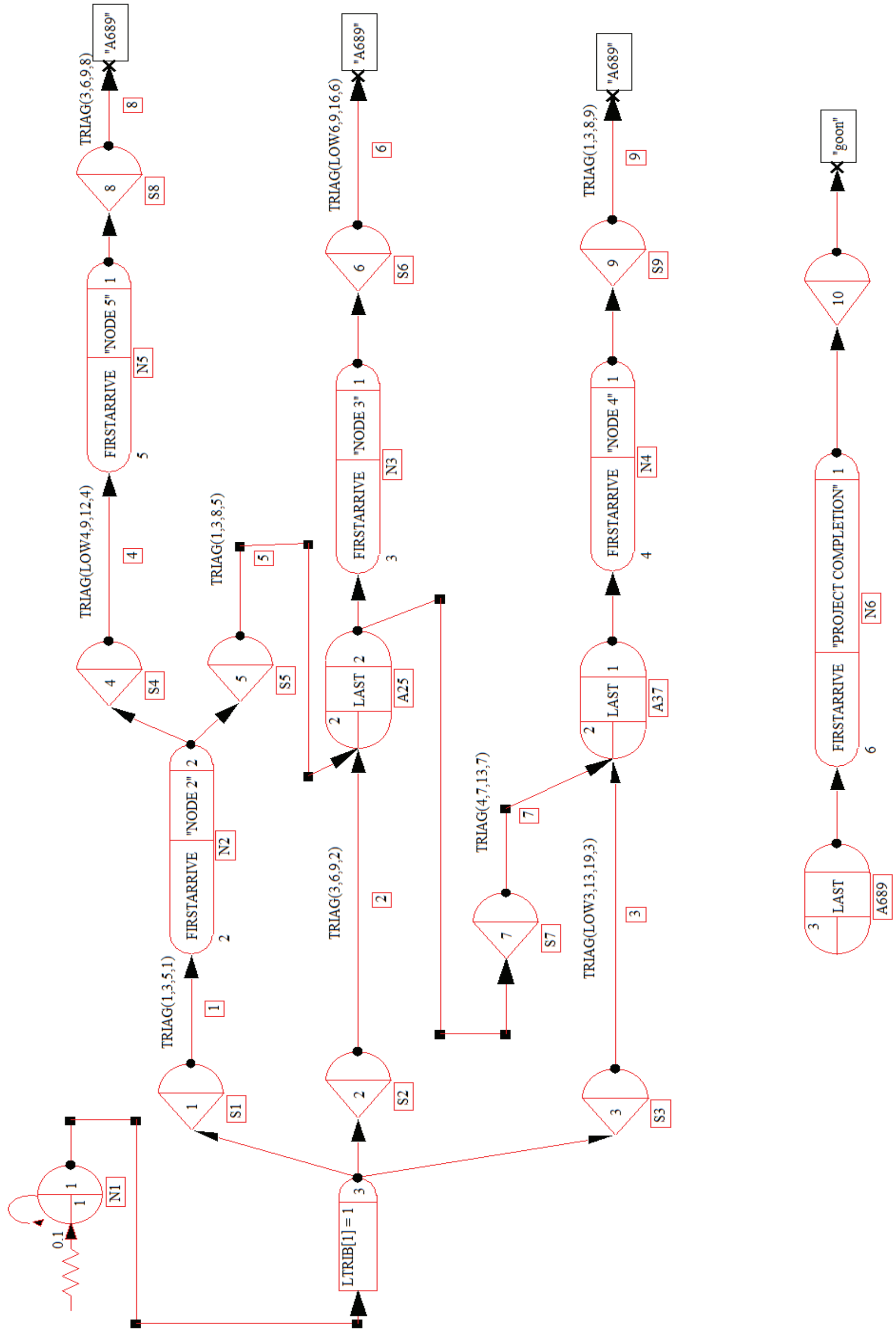
Network Statements

```
;DBF file created with Version 4
     RESOURCE,1,PART,100,{56},,,,,,,{240,20};
Base1: CREATE,EXPON(TBF1,1),0.1,ATRIB[3],,1,,,,,{80,70};
     ACTIVITY,,,,,,,,,{2,4,,,};
ASN1: ASSIGN,{{ATRIB[1],1}},,,,,,,,,{160,70};
     ACTIVITY,,,,,,,,,{4,7,,,};
     ACTIVITY,,,,"FRE0",,,,,{4,12,,,};
PORD: ACCUMULATE,FAIL,FAIL,LAST,1,,,,,,{300,70};
     ACTIVITY,9,EXPON(ARRIVE,9),,,,,,,{7,9,,,};
PREC: FREE,{{PART,FAIL}},1,,,,,,,,{420,70};
     ACTIVITY,,,,,,,,,{9,11,,,};
     TERMINATE,INF,,,,,,,,{500,70};
FRE0: AWAIT,56,{{PART,1}},ALL,,NONE,1,,,,{310,230};
     ACTIVITY,,,ATRIB[1]==1,,,,,,{12,16,,,370,190};
     ACTIVITY,,,ATRIB[1]==2,"COL2",,,,,{12,19,,,};
     ACTIVITY,,,ATRIB[1]==3,"COL3",,,,,{12,22,,,370,270};
COL1: COLCT,1,INT(ATRIB[3]),"TIME TO OR1",,,,1,,,,{440,190};
     ACTIVITY,,,,,,,,,{16,18,,,};
     TERMINATE,INF,,,,,,,,{574,190};
COL2: COLCT,2,INT(ATRIB[3]),"TIME TO OR2",,,,1,,,,{440,230};
     ACTIVITY,,,,,,,,,{19,21,,,};
     TERMINATE,INF,,,,,,,,{575,230};
COL3: COLCT,3,INT(ATRIB[3]),"TIME TO OR3",,,,1,,,,{440,270};
     ACTIVITY,,,,,,,,,{22,24,,,};
     TERMINATE,INF,,,,,,,,{575,270};
Base2: CREATE,EXPON(TBF2,2),0.1,ATRIB[3],,1,,,,,{80,150};
     ACTIVITY,,,,,,,,,{25,27,,,};
ASN2: ASSIGN,{{ATRIB[1],2}},,,,,,,,,{160,150};
     ACTIVITY,,,,"PORD",,,,,{27,7,,,};
     ACTIVITY,,,,"FRE0",,,,,{27,12,,,};
Base3: CREATE,EXPON(TBF3,3),0.1,ATRIB[3],,1,,,,,{80,230};
```

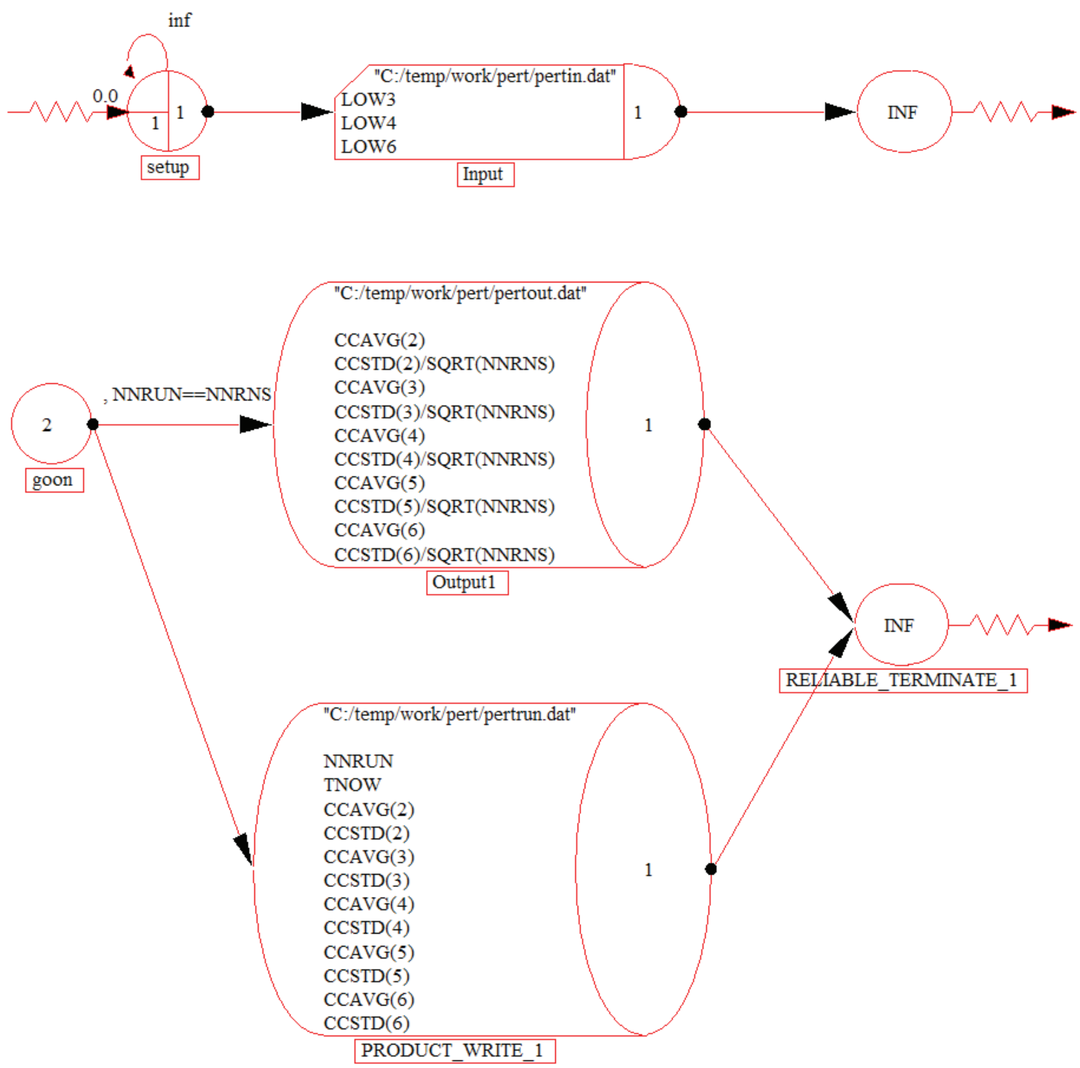
```
    ACTIVITY,,,,,,,,,{30,32,,,};
ASN3: ASSIGN,{{ATRIB[1],3}},,,,,,,,,{160,230};
    ACTIVITY,,,,"PORD",,,,,,{32,7,,,};
    ACTIVITY,,,,"FRE0",,,,,,{32,12,,,};
setup: CREATE,INF,0.0,,1,1,,,,,,{44,357};
    ACTIVITY,,,,,,,,,{35,37,,,};
Input: READ,"C:/temp/work/logistic/log_in.dat",YES,,,{FAIL,ARRIVE,
      TBF1,TBF2,TBF3},1,,,,,{71,357};
    ACTIVITY,,ttfin-0.01,,,,,,,,{37,39,,,};
    GOON,2,,,,,,,,,{250,357};
    ACTIVITY,,,NNRUN==NNRNS,,,,,,,{39,42,,,};
    ACTIVITY,,,,"Output2",,,,,,{39,45,,,};
Output1: WRITE,"C:/temp/work/logistic/log_out.dat",NO,,{CCNUM(1)/NNRNS,
      CCNUM(2)/NNRNS,CCNUM(3)/NNRNS,CCAVG(1),CCAVG(2),
      CCAVG(3),CCSTD(1)/SQRT(NNRNS),CCSTD(2)/SQRT(NNRNS),
      CCSTD(3)/SQRT(NNRNS),RRAVG(1) ,RRSTD(1)},1,,,,,,{360,356};
    ACTIVITY,,,,,,,,,{42,44,,,};
RELIABLE_TERMINATE_1: TERMINATE,INF,,,,,,,,,{553,356};
Output2: WRITE,"C:/temp/work/logistic/log_run.dat",NO,,{NNRUN,TNOW,
      CCNUM(1)/NNRUN,CCNUM(2)/NNRUN,CCNUM(3)/NNRUN,CCAVG(1),
      CCAVG(2),CCAVG(3),CCSTD(1),CCSTD(2),CCSTD(3),RRAVG(1) ,
      RRSTD(1)},1,,,,,,{357,498};
    ACTIVITY,,,,"RELIABLE_TERMINATE_1",,,,,,{45,44,,,};
```

PERT Domain

Network Model: Main Section



Network Model: Data Section



Control Statements

GEN,"Glenn Kuriger","PERT Network",,400,YES,YES;
REPORT,,YES,YES,EVERY(400);
EQUIVALENCE,{{LOW3,XX[3]},{LOW4,XX[4]},{LOW6,XX[6]}};
LIMITS,10,,,10,10;
NET;
INITIALIZE,,,NO,,NO;
FIN;

Network Statements

```
;DBF file created with Version 4
N1: CREATE,,0.1,,1,1,,,,,,{40,140};
      ACTIVITY,,,,,,,,,{1,3,,,};
      ASSIGN,{{LTRIB[1],1}},3,,,,,,,,{80,140};
      ACTIVITY,,,,,,,,,{3,7,,,};
      ACTIVITY,,,,"S2",,,,,,{3,35,,,};
      ACTIVITY,,,,"S3",,,,,,{3,37,,,};
S1: EVENT,1,,,,,,,,,{180,80};
      ACTIVITY,1,TRIAG(1,3,5,1),,,,,,,{7,9,,,};
N2: COLCT,2,FIRSTARRIVE,"NODE 2",,,,2,,,,{240,80};
      ACTIVITY,,,,,,,,,{9,12,,,};
      ACTIVITY,,,,"S5",,,,,,{9,18,,,};
S4: EVENT,4,,,,,,,,,{350,50};
      ACTIVITY,4,TRIAG(LOW4,9,12,4),,,,,,,{12,14,,,};
N5: COLCT,5,FIRSTARRIVE,"NODE 5",,,,1,,,,{467,50};
      ACTIVITY,,,,,,,,,{14,16,,,};
S8: EVENT,8,,,,,,,,,{590,50};
      ACTIVITY,8,TRIAG(3,6,9,8),,"A689",,,,,,{16,-1,,,642,50};
S5: EVENT,5,,,,,,,,,{360,110};
      ACTIVITY,5,TRIAG(1,3,8,5),,,,,,,{18,20,,,};
A25: ACCUMULATE,2,,LAST,2,,,,,,,{420,140};
      ACTIVITY,,,,,,,,,{20,23,,,};
      ACTIVITY,,,,"S7",,,,,,{20,27,,,470,170,280,170,280,190};
N3: COLCT,3,FIRSTARRIVE,"NODE 3",,,,1,,,,{480,140};
      ACTIVITY,,,,,,,,,{23,25,,,};
S6: EVENT,6,,,,,,,,,{600,140};
      ACTIVITY,6,TRIAG(LOW6,9,16,6),,"A689",,,,,,{25,-1,,,670,140};
S7: EVENT,7,,,,,,,,,{330,190};
      ACTIVITY,7,TRIAG(4,7,13,7),,,,,,,{27,29,,,};
A37: ACCUMULATE,2,,LAST,1,,,,,,,{390,230};
      ACTIVITY,,,,,,,,,{29,31,,,};
N4: COLCT,4,FIRSTARRIVE,"NODE 4",,,,1,,,,{460,230};
      ACTIVITY,,,,,,,,,{31,33,,,};
S9: EVENT,9,,,,,,,,,{580,230};
      ACTIVITY,9,TRIAG(1,3,8,9),,"A689",,,,,,{33,-1,,,631,230};
S2: EVENT,2,,,,,,,,,{180,140};
      ACTIVITY,2,TRIAG(3,6,9,2),,"A25",,,,,,{35,20,,,};
S3: EVENT,3,,,,,,,,,{190,230};
      ACTIVITY,3,TRIAG(LOW3,13,19,3),,"A37",,,,,,{37,29,,,};
A689: ACCUMULATE,3,,LAST,,,,,,,,{228,315};
      ACTIVITY,,,,,,,,,{39,41,,,};
N6: COLCT,6,FIRSTARRIVE,"PROJECT COMPLETION",,,,1,,,,{300,315};
      ACTIVITY,,,,,,,,,{41,43,,,};
      EVENT,10,,,,,,,,,{488,315};
```

```
      ACTIVITY,,,,"goon",,,,,,{43,-1,,,534,315};
setup: CREATE,inf,0.0,,1,1,,,,,,{216,398};
      ACTIVITY,,,,,,,,,{45,47,,,};
Input: READ,"C:/temp/work/pert/pertin.dat",YES,,,{LOW3,LOW4,LOW6},
         1,,,,,{265,398};
      ACTIVITY,,,,,,,,,{47,49,,,};
      TERMINATE,INF,,,,,,,,,{431,398};
goon: GOON,2,,,,,,,,,{223,495};
      ACTIVITY,,,NNRUN==NNRNS,,,,,,{50,53,,,};
      ACTIVITY,,,,"PRODUCT_WRITE_1",,,,,,{50,56,,,};
Output1: WRITE,"C:/temp/work/pert/pertout.dat",NO,,{CCAVG(2),
         CCSTD(2)/SQRT(NNRNS),CCAVG(3),CCSTD(3)/SQRT(NNRNS),
         CCAVG(4),CCSTD(4)/SQRT(NNRNS),CCAVG(5),
         CCSTD(5)/SQRT(NNRNS),CCAVG(6),CCSTD(6)/SQRT(NNRNS)},
         1,,,,,{326,493};
      ACTIVITY,,,,,,,,,{53,55,,,};
RELIABLE_TERMINATE_1: TERMINATE,INF,,,,,,,,,{512,563};
PRODUCT_WRITE_1:          WRITE,"C:/temp/work/pert/pertrun.dat",NO,,{NNRUN,
         TNOW,CCAVG(2),CCSTD(2),CCAVG(3),CCSTD(3),CCAVG(4),
         CCSTD(4),CCAVG(5),CCSTD(5),CCAVG(6),CCSTD(6)},1,,,,,,{323,632};
      ACTIVITY,,,,"RELIABLE_TERMINATE_1",,,,,,{56,55,,,};
```

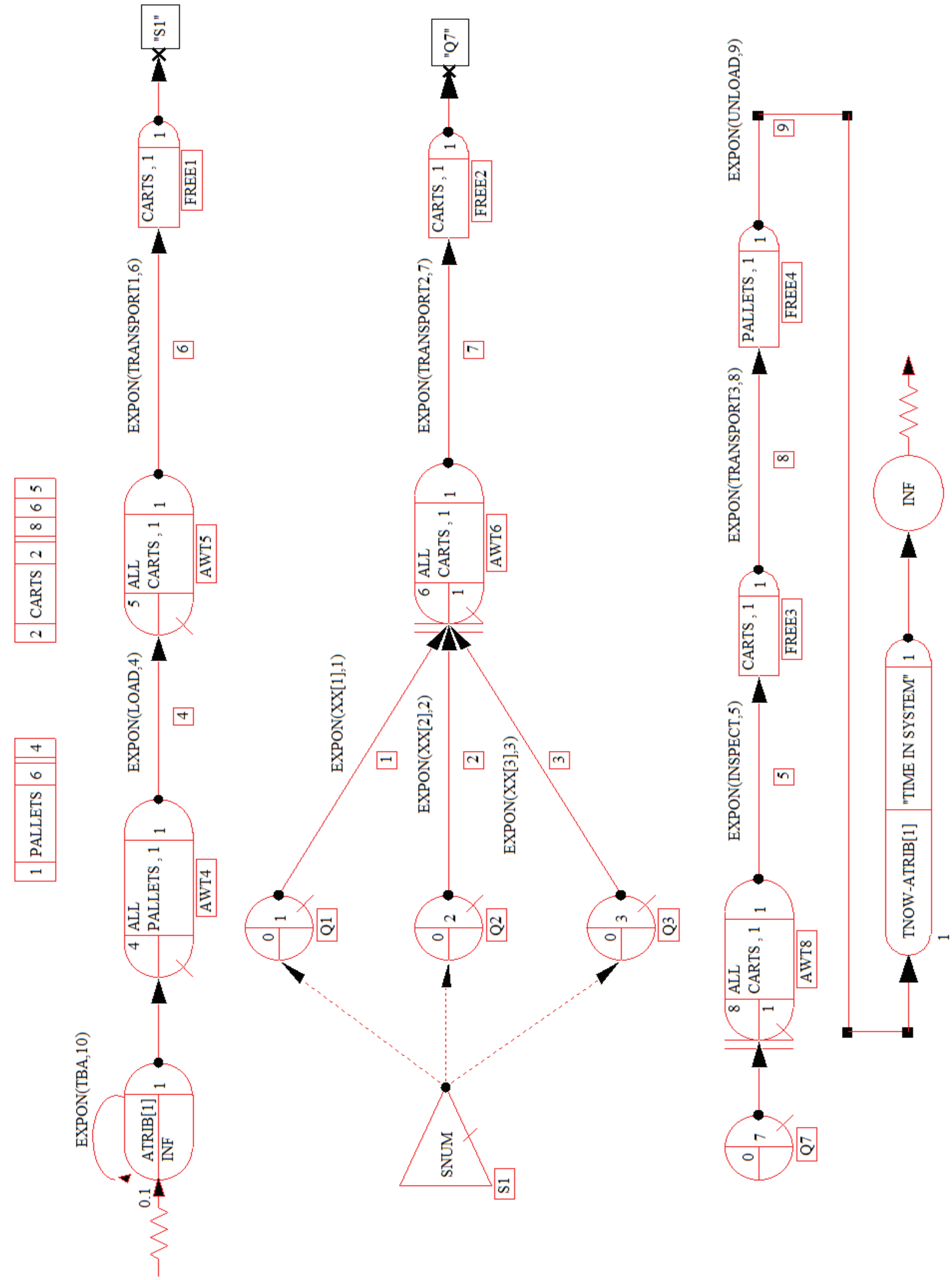
C++ Code for the Event Node: "pert"

```
#include "vslam.h"

double XCRIT[10] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

/*********************************************************************
* FUNCTION: EVENT
*
* DESCRIPTION: routine for user hook EVENT
*
* INPUTS:
*       iCode  - code number
*       peNode - current node
*
* RETURN: NONE
*********************************************************************/

void EVENT(int NXTACT, ENTITY * peNode)
{
        int I, INUM, IMAX;
        double YCRIT;
        int NEXT;
```

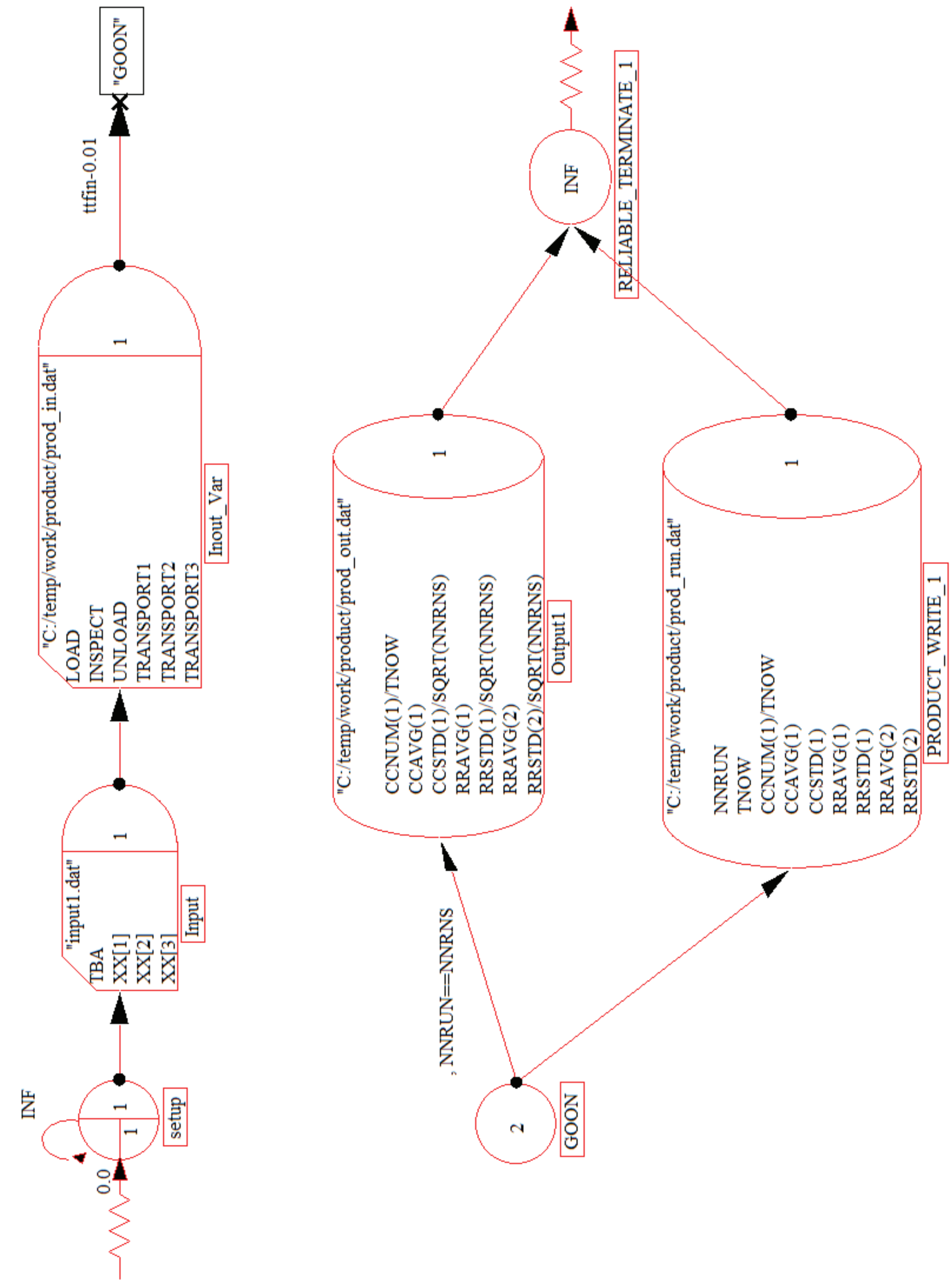
```
        if (NXTACT < 10)
        {
                peNode -> LTRIB[1] = peNode -> LTRIB[1] + 1;
                NEXT = peNode -> LTRIB[1];
                peNode -> LTRIB[NEXT] = NXTACT;
        }
        else
        {
                // ADD TO NUMBER OF TIMES AN ACTIVITY IS CRITICAL
                IMAX = peNode -> LTRIB[1];
                for (I = 2; I <= IMAX; I++)
                {
                        INUM = peNode -> LTRIB[I];
                        XCRIT[INUM] = XCRIT[INUM] + 1.0;
                }

                if (NNRUN >= 400)
                {
                        FILE *out;
                        out = fopen("C:/temp/work/pert/pertcrit.dat", "w");
                        for(I = 1; I <= 9; I++)
                        {
                                YCRIT = XCRIT[I] / NNRNS;
                                SU_OUT (TRUE,TRUE, "The criticality index for
                                     activity %d is %f\n", I, YCRIT);
                        }

                        fprintf(out, "%f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\t %f\n",
                            XCRIT[1] / NNRNS, XCRIT[2] / NNRNS, XCRIT[3] /
                            NNRNS, XCRIT[4] / NNRNS, XCRIT[5] / NNRNS,
                            XCRIT[6] / NNRNS, XCRIT[7] / NNRNS, XCRIT[8] /
                            NNRNS, XCRIT[9] / NNRNS);
                        fclose(out);
                }
        }
}
```

Production Domain

Network Model: Main Section



Network Model: Data Section



Control Statements

```
GEN,"Glenn Kuriger","Production Network",,400,YES,YES;
REPORT,,YES,YES,EVERY(400);
LIMITS,10,10,,1;
EQUIVALENCE,{{TBA,XX[0]},{LOAD,XX[4]},{INSPECT,XX[5]},{TRANSPORT1,
        XX[6]},{TRANSPORT2,XX[7]},{TRANSPORT3,XX[8]},{UNLOAD,XX[9]},
        {PALLET,LL[5]},{CART,LL[6]}};
SEEDS,{{428956419,1,NO},{1954324947,2,NO},{1145661099,3,NO},{1145661099,3,
        NO},{1145661099,4,NO},{794161987,5,NO},{1329531353,6,NO},{200496737,
        7,NO},{633816299,8,NO},{1410143363,9,NO}};
INITIALIZE,0.0,108,NO,,NO;
MONTR,CLEAR,4;
NET;
FIN;
```

Network Statements

```
;DBF file created with Version 4
     RESOURCE,1,PALLETS,6,{4},,,,,,,{166,56};
     RESOURCE,2,CARTS,2,{8,6,5},,,,,,,{253,56};
     CREATE,EXPON(TBA,10),0.1,ATRIB[1],INF,1,,,,,,{69,99};
     ACTIVITY,,,,,,,,,{3,5,,,};
AWT4: AWAIT,4,{{PALLETS,1}},ALL,,NONE,1,,,,,{143,99};
     ACTIVITY,4,EXPON(LOAD,4),,,,"Load Pallet",,,,{5,7,,,};
AWT5: AWAIT,5,{{CARTS,1}},ALL,,NONE,1,,,,,{268,99};
     ACTIVITY,6,EXPON(TRANSPORT1,6),,,,"Transport to machine",,,,{7,9,,,};
FREE1: FREE,{{CARTS,1}},1,,,,,,,,,{405,99};
     ACTIVITY,,,,"S1",,,,,,{9,-1,,,468,99};
S1: SELECT,SNUM,NONE,NONE,{Q1,Q2,Q3},,,,,,,{18,199};
     ;CONNECTOR{11,15}
     ;CONNECTOR{11,21}
     ;CONNECTOR{11,23}
Q1: QUEUE,1,0,,NONE,,,,,,,{150,141};
     ACTIVITY,1,EXPON(XX[1],1),,,,,,,,{15,17,,,};
AWT6: AWAIT,6,{{CARTS,1}},ALL,1,BLOCK,1,,,,,{272,200};
     ACTIVITY,7,EXPON(TRANSPORT2,7),,,,"Transport to inspection",,,,{17,19,,,};
FREE2: FREE,{{CARTS,1}},1,,,,,,,,,{401,200};
     ACTIVITY,,,,"Q7",,,,,,{19,-1,,,462,200};
Q2: QUEUE,2,0,,NONE,,,,,,,{150,200};
     ACTIVITY,2,EXPON(XX[2],2),,"AWT6",,,,,,{21,17,,,};
Q3: QUEUE,3,0,,NONE,,,,,,,{150,260};
     ACTIVITY,3,EXPON(XX[3],3),,"AWT6",,,,,,{23,17,,,};
Q7: QUEUE,7,0,,NONE,,,,,,,{70,308};
     ACTIVITY,,,,,,,,,{25,27,,,};
```

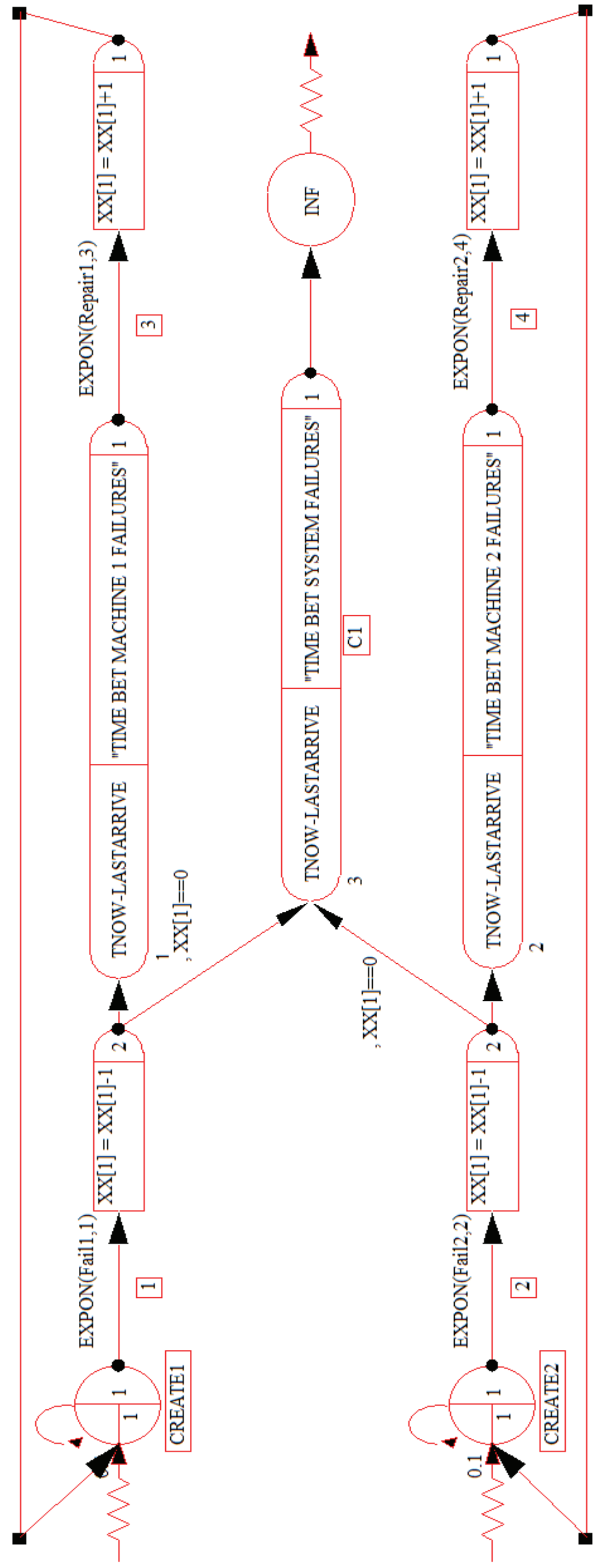
```
AWT8: AWAIT,8,{{CARTS,1}},ALL,1,BLOCK,1,,,,,{120,308};
      ACTIVITY,5,EXPON(INSPECT,5),,,,"Inspection",,,,{27,29,,,};
FREE3: FREE,{{CARTS,1}},1,,,,,,,,{241,308};
      ACTIVITY,8,EXPON(TRANSPORT3,8),,,,"Transport to unload",,,,{29,31,,,};
FREE4: FREE,{{PALLETS,1}},1,,,,,,,,{361,308};
      ACTIVITY,9,EXPON(UNLOAD,9),,,,"Unload
        pallet",,,,{31,33,,,446,308,446,339,111,339,111,360};
      COLCT,1,TNOW-ATRIB[1],"TIME IN SYSTEM",,,,1,,,,{147,360};
      ACTIVITY,,,,,,,,,{33,35,,,};
      TERMINATE,INF,,,,,,,,,{306,360};
setup: CREATE,INF,0.0,,1,1,,,,,{67,489};
      ACTIVITY,,,,,,,,,{36,38,,,};
Input: READ,"input1.dat",YES,,,{TBA,XX[1],XX[2],XX[3]},1,,,,,{105,489};
      ACTIVITY,,,,,,,,,{38,40,,,};
Inout_Var: READ,"C:/temp/work/product/prod_in.dat",YES,,,{LOAD,INSPECT,
        UNLOAD,TRANSPORT1,TRANSPORT2,TRANSPORT3},1,,,,,{194,489};
      ACTIVITY,,ttfin-0.01,,"GOON",,,,,,{40,-1,,,367,489};
GOON: GOON,2,,,,,,,,,{66,606};
      ACTIVITY,,,NNRUN==NNRNS,,,,,,,{42,45,,,};
      ACTIVITY,,,,"PRODUCT_WRITE_1",,,,,,{42,48,,,};
Output1: WRITE,"C:/temp/work/product/prod_out.dat",NO,,
        {CCNUM(1)/TNOW,CCAVG(1),CCSTD(1)/SQRT(NNRNS),RRAVG(1),
        RRSTD(1)/SQRT(NNRNS),RRAVG(2),RRSTD(2)/SQRT(NNRNS)},1,,,,,,
        {163,583};
      ACTIVITY,,,,,,,,,{45,47,,,};
RELIABLE_TERMINATE_1: TERMINATE,INF,,,,,,,,,{343,622};
PRODUCT_WRITE_1: WRITE,"C:/temp/work/product/prod_run.dat",NO,,{NNRUN,
        TNOW,CCNUM(1)/TNOW,CCAVG(1),CCSTD(1),RRAVG(1),RRSTD(1),
        RRAVG(2),RRSTD(2)},1,,,,,,{157,687};
      ACTIVITY,,,,"RELIABLE_TERMINATE_1",,,,,,{48,47,,,};
```

Input File: ”input1.dat”

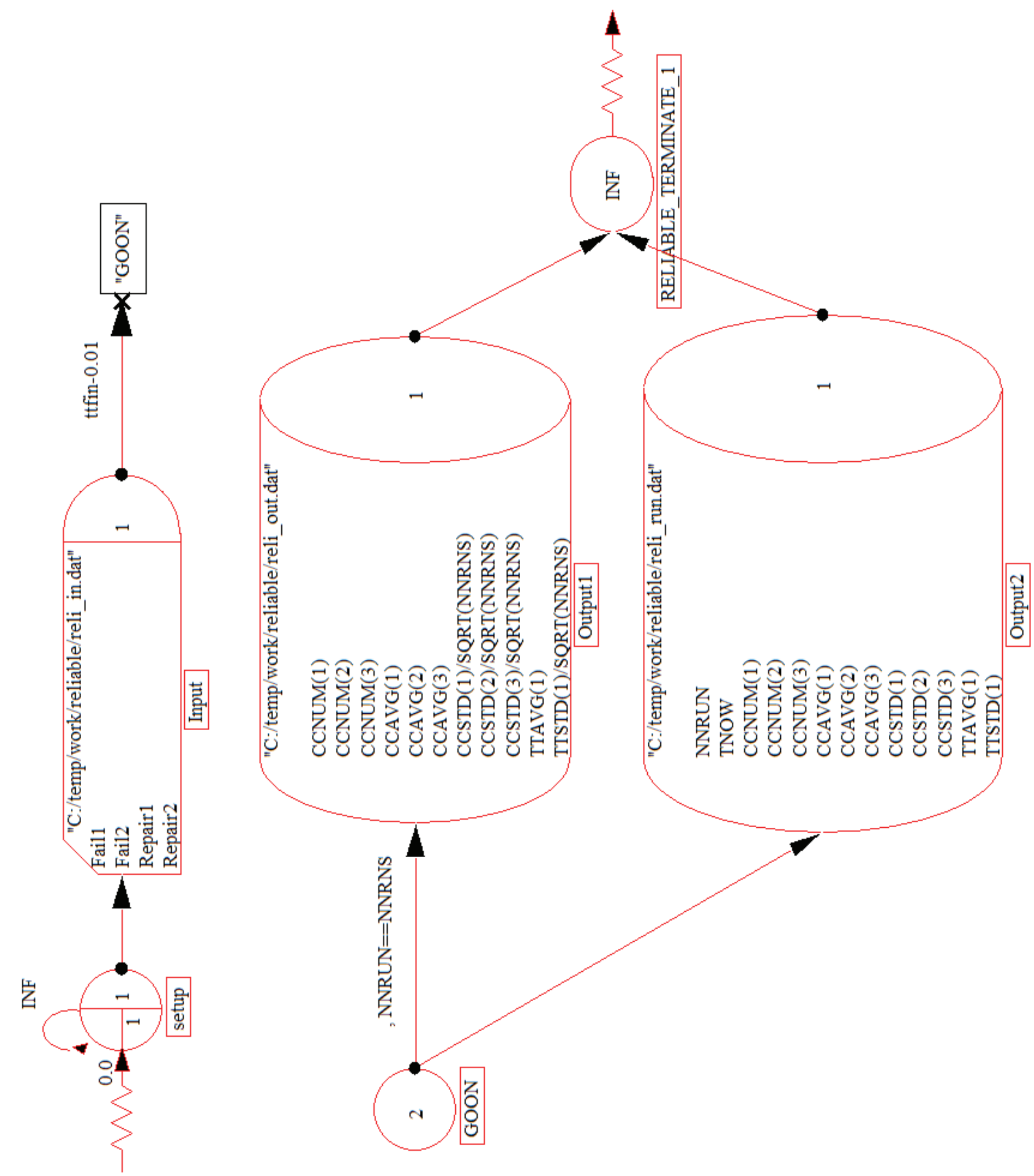
```
5        8        10        12
```

Reliability Domain

Network Model: Main Section



Network Model: DataSection



Control Statements

```
GEN,"Glenn Kuriger","Reliability Domain",,400,YES,YES;
REPORT,,YES,YES,EVERY(400);
LIMITS,10,10,,10,10;
EQUIVALENCE,{{Fail1,XX[5]},{Fail2,XX[6]},{Repair1,XX[7]},{Repair2,XX[8]}};
TIMST,1,XX[1],"DOWN TIME",0,0.0,1.0;
INITIALIZE,0.0,108,NO,,NO;
MONTR,CLEAR,4;
INTLC,{{XX[1],2}};
NET;
FIN;
```

Network Statements

```
;DBF file created with Version 4
setup: CREATE,INF,0.0,,1,1,,,,,,{71,246};
    ACTIVITY,,,,,,,,,{1,3,,,};
Input: READ,"C:/temp/work/reliable/reli_in.dat",YES,,,{Fail1,Fail2,Repair1,
      Repair2},1,,,,,{98,245};
    ACTIVITY,,ttfin-0.01,,,,,,,{3,5,,,};
    GOON,2,,,,,,,,,{277,246};
    ACTIVITY,,,NNRUN==NNRNS,,,,,,{5,8,,,};
    ACTIVITY,,,,"Output2",,,,,{5,11,,,};
Output1: WRITE,"C:/temp/work/reliable/reli_out.dat",NO,,{CCNUM(1),
      CCNUM(2),CCNUM(3),CCAVG(1),CCAVG(2),CCAVG(3),
      CCSTD(1)/SQRT(NNRNS),CCSTD(2)/SQRT(NNRNS),
      CCSTD(3)/SQRT(NNRNS),TTAVG(1),TTSTD(1)/SQRT(NNRNS)},
      1,,,,,{387,244};
    ACTIVITY,,,,,,,,{8,10,,,};
RELIABLE_TERMINATE_1: TERMINATE,INF,,,,,,,,,{567,246};
Output2: WRITE,"C:/temp/work/reliable/reli_run.dat",NO,,{NNRUN,TNOW,
      CCNUM(1),CCNUM(2),CCNUM(3),CCAVG(1),CCAVG(2),CCAVG(3),
      CCSTD(1),CCSTD(2),CCSTD(3),TTAVG(1),TTSTD(1)},1,,,,,{371,402};
    ACTIVITY,,,,"RELIABLE_TERMINATE_1",,,,,{11,10,,,};
CREATE1: CREATE,,0.1,,1,1,,,,,{68,43};
    ACTIVITY,1,EXPON(Fail1,1),,,,,,,{13,15,,,};
    ASSIGN,{{XX[1],XX[1]-1}},2,,,,,,,,{148,43};
    ACTIVITY,,,,,,,,,{15,18,,,};
    ACTIVITY,,,XX[1]==0,"C1",,,,,{15,22,,,};
    COLCT,1,TNOW-LASTARRIVE,"TIME BET MACHINE 1
      FAILURES",,,,1,,,,{248,43};
    ACTIVITY,3,EXPON(Repair1,3),,,,,,,{18,20,,,};
    ASSIGN,{{XX[1],XX[1]+1}},1,,,,,,,,{498,43};
    ACTIVITY,,,,"CREATE1",,,,,{20,13,,,568,13,19,13};
```

```
C1: COLCT,3,TNOW-LASTARRIVE,"TIME BET SYSTEM
      FAILURES",,,,1,,,,{268,93};
    ACTIVITY,,,,,,,,,{22,24,,,};
    TERMINATE,INF,,,,,,,,,{498,93};
CREATE2: CREATE,,0.1,,1,1,,,,,,{68,143};
    ACTIVITY,2,EXPON(Fail2,2),,,,,,,,{25,27,,,};
    ASSIGN,{{XX[1],XX[1]-1}},2,,,,,,,,{148,143};
    ACTIVITY,,,XX[1]==0,"C1",,,,,,{27,22,,,};
    ACTIVITY,,,,,,,,,{27,30,,,};
    COLCT,2,TNOW-LASTARRIVE,"TIME BET MACHINE 2
      FAILURES",,,,1,,,,{248,143};
    ACTIVITY,4,EXPON(Repair2,4),,,,,,,,{30,32,,,};
    ASSIGN,{{XX[1],XX[1]+1}},1,,,,,,,,{498,143};
    ACTIVITY,,,,"CREATE2",,,,,,{32,25,,,568,173,28,173};
```

APPENDIX C

SAS RESULTS

SAS Results: ANOVA and Duncan's Multiple Range Test Results

00:21 Friday, October 27, 2006 1

The GLM Procedure

Class Level Information

Class	Levels	Values
Method	3	GA-SO LNM-SO TS-SO
Domain	5	INV LOG PERT PROD RELI

Number of Observations Read	301
Number of Observations Used	300

00:21 Friday, October 27, 2006 2

The GLM Procedure

Dependent Variable: Count

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	89998491.55	6428463.68	2405.50	<.0001
Error	285	761634.25	2672.40		
Corrected Total	299	90760125.80			

R-Square	Coeff Var	Root MSE	Count Mean
0.991608	3.431717	51.69527	1506.397

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	11383876.38	2845969.10	1064.95	<.0001
Method	2	67570388.25	33785194.12	12642.3	<.0001
Method*Domain	8	11044226.92	1380528.37	516.59	<.0001

The GLM Procedure

Dependent Variable: ROCCM

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	8018220.520	572730.037	162.59	<.0001
Error	285	1003908.050	3522.484		
Corrected Total	299	9022128.570			

R-Square	Coeff Var	Root MSE	ROCCM Mean
0.888728	29.24968	59.35052	202.9100

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	1443760.687	360940.172	102.47	<.0001
Method	2	4082706.620	2041353.310	579.52	<.0001
Method*Domain	8	2491753.213	311469.152	88.42	<.0001

00:21 Friday, October 27, 2006 4

The GLM Procedure

Dependent Variable: CPU_CM

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	1130325075	80737505	55.18	<.0001
Error	285	417039530	1463297		
Corrected Total	299	1547364605			

R-Square	Coeff Var	Root MSE	CPU_CM Mean
0.730484	65.53398	1209.668	1845.864

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	346268097.9	86567024.5	59.16	<.0001
Method	2	389187840.1	194593920.0	132.98	<.0001
Method*Domain	8	394869136.6	49358642.1	33.73	<.0001

The GLM Procedure

Dependent Variable: ROCBS

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	3874323.680	276737.406	89.77	<.0001
Error	285	878564.250	3082.682		
Corrected Total	299	4752887.930			

R-Square	Coeff Var	Root MSE	ROCBS Mean
0.815151	43.84577	55.52190	126.6300

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	492360.880	123090.220	39.93	<.0001
Method	2	1956098.580	978049.290	317.27	<.0001
Method*Domain	8	1425864.220	178233.028	57.82	<.0001

The GLM Procedure

Dependent Variable: CPU_BS

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	395587147.8	28256224.8	31.98	<.0001
Error	285	251818455.8	883573.5		
Corrected Total	299	647405603.5			

R-Square	Coeff Var	Root MSE	CPU_BS Mean
0.611034	86.99812	939.9859	1080.467

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	88269217.8	22067304.5	24.98	<.0001
Method	2	152193071.6	76096535.8	86.12	<.0001
Method*Domain	8	155124858.4	19390607.3	21.95	<.0001

00:21 Friday, October 27, 2006 7

The GLM Procedure

Dependent Variable: BSF_Z

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	20515575.71	1465398.26	27.70	<.0001
Error	285	15079708.25	52911.26		
Corrected Total	299	35595283.96			

R-Square	Coeff Var	Root MSE	BSF_Z Mean
0.576357	105.4898	230.0245	218.0537

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	10918085.32	2729521.33	51.59	<.0001
Method	2	3367063.10	1683531.55	31.82	<.0001
Method*Domain	8	6230427.29	778803.41	14.72	<.0001

00:21 Friday, October 27, 2006 8

The GLM Procedure

Dependent Variable: BSF_Zdif

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	17713424.57	1265244.61	23.91	<.0001
Error	285	15079708.25	52911.26		
Corrected Total	299	32793132.83			

R-Square	Coeff Var	Root MSE	BSF_Zdif Mean
0.540157	154.4359	230.0245	148.9449

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	8115934.183	2028983.546	38.35	<.0001
Method	2	3367063.101	1683531.550	31.82	<.0001
Method*Domain	8	6230427.288	778803.411	14.72	<.0001

The GLM Procedure

Dependent Variable: Init_Zdif

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	26386845941	1884774710	59.33	<.0001
Error	285	9054073158	31768678		
Corrected Total	299	35440919099			

R-Square	Coeff Var	Root MSE	Init_Zdif Mean
0.744531	100.6153	5636.371	5601.900

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Domain	4	24960996195	6240249049	196.43	<.0001
Method	2	281312487	140656244	4.43	0.0128
Method*Domain	8	1144537259	143067157	4.50	<.0001

The GLM Procedure

Duncan's Multiple Range Test for Count

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	2672.401

Number of Means	2	3	4	5
Critical Range	18.58	19.56	20.21	20.69

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	1809.283	60	PROD
B	1630.667	60	LOG
C	1478.683	60	RELI
D	1338.317	60	INV
E	1275.033	60	PERT

The GLM Procedure

Duncan's Multiple Range Test for ROCCM

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	3522.484

Number of Means	2	3	4	5
Critical Range	21.33	22.45	23.20	23.76

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	283.87	60	PROD
B	253.22	60	LOG
C	217.28	60	INV
D	175.73	60	RELI
E	84.45	60	PERT

The GLM Procedure

Duncan's Multiple Range Test for CPU_CM

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	1463297

Number of Means	2	3	4	5
Critical Range	434.7	457.6	472.9	484.2

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	3778.9	60	PROD
B	1707.7	60	RELI
B	1680.5	60	LOG
B	1605.8	60	INV
C	456.4	60	PERT

The GLM Procedure

Duncan's Multiple Range Test for ROCBS

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	3082.682

Number of Means	2	3	4	5
Critical Range	19.95	21.00	21.71	22.22

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	173.98	60	LOG
B	148.10	60	PROD
B	144.10	60	INV
C	110.63	60	RELI
D	56.33	60	PERT

The GLM Procedure

Duncan's Multiple Range Test for CPU_BS

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	883573.5

Number of Means	2	3	4	5
Critical Range	337.8	355.6	367.5	376.3

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	1955.0	60	PROD
B	1092.5	60	LOG
B	1064.4	60	INV
B	1049.7	60	RELI
C	240.8	60	PERT

00:21 Friday, October 27, 2006 15

The GLM Procedure

Duncan's Multiple Range Test for BSF_Z

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	52911.26

Number of Means	2	3	4	5
Critical Range	82.66	87.02	89.93	92.08

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	514.81	60	INV
B	290.76	60	RELI
B	264.03	60	LOG
C	20.54	60	PROD
C	0.13	60	PERT

00:21 Friday, October 27, 2006 16

The GLM Procedure

Duncan's Multiple Range Test for BSF_Zdif

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	52911.26

Number of Means	2	3	4	5
Critical Range	82.66	87.02	89.93	92.08

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	400.40	60	INV
B	288.22	60	RELI
C	52.78	60	LOG
C	3.32	60	PROD
C	0.00	60	PERT

00:21 Friday, October 27, 2006 17

The GLM Procedure

Duncan's Multiple Range Test for Init_Zdif

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	31768678

Number of Means	2	3	4	5
Critical Range	2026	2132	2204	2256

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Domain
A	23545	60	RELI
B	4307	60	INV
C	123	60	LOG
C	34	60	PROD
C	0	60	PERT

The GLM Procedure

Duncan's Multiple Range Test for Count

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	2672.401

Number of Means	2	3
Critical Range	14.39	15.15

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	1993.160	100	GA-SO
B	1663.200	100	TS-SO
C	862.830	100	LNM-SO

The GLM Procedure

Duncan's Multiple Range Test for ROCCM

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	3522.484

Number of Means	2	3
Critical Range	16.52	17.39

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	343.900	100	GA-SO
B	206.610	100	TS-SO
C	58.220	100	LNM-SO

00:21 Friday, October 27, 2006 20

The GLM Procedure

Duncan's Multiple Range Test for CPU_CM

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	1463297

Number of Means	2	3
Critical Range	336.7	354.5

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	3063.6	100	GA-SO
B	2150.2	100	TS-SO
C	323.9	100	LNM-SO

The GLM Procedure

Duncan's Multiple Range Test for ROCBS

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	3082.682

Number of Means	2	3
Critical Range	15.46	16.27

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	231.800	100	GA-SO
B	112.580	100	TS-SO
C	35.510	100	LNM-SO

The GLM Procedure

Duncan's Multiple Range Test for CPU_BS

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	883573.5

Number of Means	2	3
Critical Range	261.7	275.5

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	1935.3	100	GA-SO
B	1114.4	100	TS-SO
C	191.7	100	LNM-SO

The GLM Procedure

Duncan's Multiple Range Test for BSF_Z

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	52911.26

Number of Means	2	3
Critical Range	64.03	67.41

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	367.85	100	TS-SO
B	145.67	100	LNM-SO
B	140.64	100	GA-SO

The GLM Procedure

Duncan's Multiple Range Test for BSF_Zdif

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	285
Error Mean Square	52911.26

Number of Means	2	3
Critical Range	64.03	67.41

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	298.74	100	TS-SO
B	76.56	100	LNM-SO
B	71.53	100	GA-SO

The GLM Procedure

Duncan's Multiple Range Test for Init_Zdif

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05	
Error Degrees of Freedom	285	
Error Mean Square	31768678	
Number of Means	2	3
Critical Range	1569	1652

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	Method
A	6934.3	100	GA-SO
B	5209.9	100	TS-SO
B	4661.6	100	LNM-SO

APPENDIX D

SUMMARY DATA TABLES

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
GA-SO	INV	1	15364.65	1956	100	196	1283	166	1056	640.29	33.33	525.88	15250.24
GA-SO	INV	2	6834.55	1956	100	151	978	62	383	1075.70	33.33	961.29	6720.14
GA-SO	INV	3	2060.34	1968	100	157	1035	128	810	782.91	33.33	668.50	1945.93
GA-SO	INV	4	1282.95	1905	100	131	860	97	605	114.41	33.33	0.00	1168.54
GA-SO	INV	5	8789.61	1859	100	163	1055	138	857	114.41	33.33	0.00	8675.20
GA-SO	INV	6	2738.43	1945	100	148	974	119	748	672.56	33.33	558.15	2624.02
GA-SO	INV	7	4428.45	1965	100	184	1196	132	826	640.29	33.33	525.88	4314.04
GA-SO	INV	8	2244.65	1958	100	152	985	68	421	114.41	33.33	0.00	2130.24
GA-SO	INV	9	3668.44	1957	100	131	867	95	597	640.29	33.33	525.88	3554.03
GA-SO	INV	10	3010.99	1932	100	211	1360	150	934	640.29	33.33	525.88	2896.58
GA-SO	INV	11	3871.14	1917	100	136	873	46	286	640.29	33.33	525.88	3756.73
GA-SO	INV	12	3063.61	1976	100	116	758	78	480	114.41	33.33	0.00	2949.19
GA-SO	INV	13	4343.56	1973	100	167	1075	126	777	782.91	33.33	668.50	4229.15
GA-SO	INV	14	4256.10	1983	100	205	1308	168	1037	672.56	33.33	558.15	4141.69
GA-SO	INV	15	2031.39	1941	100	136	877	51	313	672.56	33.33	558.15	1916.98
GA-SO	INV	16	6219.04	1897	100	139	900	95	586	114.41	33.33	0.00	6104.63
GA-SO	INV	17	3116.09	1919	100	189	1217	161	1002	114.41	33.33	0.00	3001.68
GA-SO	INV	18	2318.39	1974	100	176	1122	12	74	114.41	33.33	0.00	2203.97
GA-SO	INV	19	5393.33	1965	100	155	1003	122	752	640.29	33.33	525.88	5278.92
GA-SO	INV	20	5235.09	2001	100	154	989	56	344	114.41	33.33	0.00	5120.68
GA-SO	LOG	1	213.22	2022	100	544	9378	459	7666	211.25	0.00	0.00	1.97
GA-SO	LOG	2	215.97	2060	100	550	8882	407	6382	211.25	0.00	0.00	4.73
GA-SO	LOG	3	215.62	1994	100	580	7754	369	4829	211.25	0.00	0.00	4.37
GA-SO	LOG	4	217.03	1964	100	642	8223	544	6940	211.25	0.00	0.00	5.78
GA-SO	LOG	5	214.89	2049	100	488	6187	341	4278	211.25	0.00	0.00	3.64
GA-SO	LOG	6	214.37	2031	100	604	3001	461	2235	211.25	0.00	0.00	3.12
GA-SO	LOG	7	216.50	2027	100	509	1645	356	1114	211.25	0.00	0.00	5.26
GA-SO	LOG	8	217.42	1973	100	542	1760	450	1424	211.25	0.00	0.00	6.18
GA-SO	LOG	9	215.59	2047	100	456	1461	265	811	211.25	0.00	0.00	4.34
GA-SO	LOG	10	216.74	2037	100	620	1997	483	1520	211.25	0.00	0.00	5.49

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
GA-SO	LOG	11	214.67	2048	100	564	1532	414	1083	211.25	0.00	0.00	3.43
GA-SO	LOG	12	216.06	2068	100	501	1407	333	895	211.25	0.00	0.00	4.81
GA-SO	LOG	13	218.18	2027	100	436	1242	339	926	211.25	0.00	0.00	6.93
GA-SO	LOG	14	214.63	2028	100	580	1634	357	972	211.25	0.00	0.00	3.39
GA-SO	LOG	15	212.28	1990	100	594	1675	523	1458	211.25	0.00	0.00	1.03
GA-SO	LOG	16	216.11	2006	100	560	1575	510	1408	211.25	0.00	0.00	4.86
GA-SO	LOG	17	213.20	2032	100	505	1438	435	1207	211.25	0.00	0.00	1.95
GA-SO	LOG	18	216.64	2058	100	467	1324	402	1113	211.25	0.00	0.00	5.40
GA-SO	LOG	19	213.60	1978	100	525	1487	356	969	211.25	0.00	0.00	2.35
GA-SO	LOG	20	216.45	1962	100	432	1234	368	1031	211.25	0.00	0.00	5.20
GA-SO	PERT	1	0.1559	1955	100	123	277	64	128	0.1324	50.00	0.00	0.02
GA-SO	PERT	2	0.1495	1926	100	170	415	122	266	0.1324	50.00	0.00	0.02
GA-SO	PERT	3	0.1521	1906	100	115	308	87	194	0.1324	50.00	0.00	0.02
GA-SO	PERT	4	0.1412	1898	100	160	350	123	238	0.1324	50.00	0.00	0.01
GA-SO	PERT	5	0.1399	1955	100	123	280	123	280	0.1324	50.00	0.00	0.01
GA-SO	PERT	6	0.1516	1950	100	163	360	135	262	0.1324	50.00	0.00	0.02
GA-SO	PERT	7	0.1390	1879	100	158	346	126	245	0.1324	50.00	0.00	0.01
GA-SO	PERT	8	0.1450	1961	100	140	317	98	192	0.1324	50.00	0.00	0.01
GA-SO	PERT	9	0.1414	1975	100	170	370	119	231	0.1324	50.00	0.00	0.01
GA-SO	PERT	10	0.1418	1871	100	162	350	98	191	0.1324	50.00	0.00	0.01
GA-SO	PERT	11	0.1456	1945	100	140	310	99	192	0.1324	50.00	0.00	0.01
GA-SO	PERT	12	0.1489	2046	100	186	410	135	265	0.1324	50.00	0.00	0.02
GA-SO	PERT	13	0.1556	1997	100	143	324	115	225	0.1324	50.00	0.00	0.02
GA-SO	PERT	14	0.1503	1957	100	134	315	109	217	0.1324	50.00	0.00	0.02
GA-SO	PERT	15	0.1496	1928	100	163	411	127	281	0.1324	50.00	0.00	0.02
GA-SO	PERT	16	0.1432	1935	100	195	488	158	352	0.1324	50.00	0.00	0.01
GA-SO	PERT	17	0.1533	2005	100	195	490	166	372	0.1324	50.00	0.00	0.02
GA-SO	PERT	18	0.1431	2007	100	172	428	126	279	0.1324	50.00	0.00	0.01
GA-SO	PERT	19	0.1504	1839	100	133	342	99	222	0.1324	50.00	0.00	0.02
GA-SO	PERT	20	0.1418	1928	100	174	439	137	305	0.1324	50.00	0.00	0.01

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
GA-SO	PROD	1	51.25	2102	100	605	8103	5	84	17.86	0.00	0.65	34.03
GA-SO	PROD	2	56.67	2025	100	477	6338	296	3877	17.86	0.00	0.65	39.45
GA-SO	PROD	3	35.11	2061	100	525	7621	331	4951	17.86	0.00	0.65	17.90
GA-SO	PROD	4	43.11	2073	100	571	7665	359	4775	17.86	0.00	0.65	25.89
GA-SO	PROD	5	53.39	2004	100	461	6157	279	3686	17.86	0.00	0.65	36.18
GA-SO	PROD	6	50.64	2134	100	525	6683	347	4360	17.86	0.00	0.65	33.43
GA-SO	PROD	7	48.80	2069	100	556	7510	386	5146	17.86	0.00	0.65	31.59
GA-SO	PROD	8	50.49	2097	100	512	6863	186	2424	17.86	0.00	0.65	33.28
GA-SO	PROD	9	46.10	2106	100	538	7308	310	4135	17.86	0.00	0.65	28.89
GA-SO	PROD	10	47.70	2010	100	473	6331	262	3437	17.86	0.00	0.65	30.49
GA-SO	PROD	11	45.52	2071	100	549	7455	356	4723	17.86	0.00	0.65	28.30
GA-SO	PROD	12	63.54	2004	100	585	7371	424	5317	17.86	0.00	0.65	46.32
GA-SO	PROD	13	50.91	2082	100	513	6655	281	3585	17.86	0.00	0.65	33.69
GA-SO	PROD	14	45.25	2147	100	437	5848	199	2564	17.86	0.00	0.65	28.04
GA-SO	PROD	15	63.01	2126	100	538	7297	273	3666	17.86	0.00	0.65	45.80
GA-SO	PROD	16	62.14	2100	100	587	7825	351	4611	17.86	0.00	0.65	44.93
GA-SO	PROD	17	50.30	2107	100	455	6087	312	4124	17.86	0.00	0.65	33.08
GA-SO	PROD	18	58.89	2033	100	617	11427	334	5801	17.86	0.00	0.65	41.68
GA-SO	PROD	19	42.21	1970	100	522	11105	324	7167	17.86	0.00	0.65	25.00
GA-SO	PROD	20	55.26	2102	100	628	11796	444	7871	17.86	0.00	0.65	38.04
GA-SO	RELI	1	36047.01	1970	100	348	8056	261	6064	2.95	66.67	0.41	36044.47
GA-SO	RELI	2	42248.44	2013	100	353	6799	235	4839	2.95	66.67	0.41	42245.90
GA-SO	RELI	3	17039.57	1948	100	271	5077	188	3243	2.95	66.67	0.41	17037.04
GA-SO	RELI	4	30483.82	1986	100	238	3155	143	1851	2.95	66.67	0.41	30481.28
GA-SO	RELI	5	25014.06	1957	100	345	4561	223	2908	2.95	66.67	0.41	25011.52
GA-SO	RELI	6	13941.52	2007	100	351	4454	230	2850	3.54	66.67	1.00	13938.98
GA-SO	RELI	7	24872.34	1980	100	344	3419	298	2960	2.95	66.67	0.41	24869.80
GA-SO	RELI	8	32793.57	1984	100	397	3805	289	2764	2.95	66.67	0.41	32791.03
GA-SO	RELI	9	48569.31	1987	100	388	3586	203	1849	2.95	66.67	0.41	48566.77
GA-SO	RELI	10	32113.11	1989	100	341	3108	263	2351	2.95	66.67	0.41	32110.57

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
GA-SO	RELI	11	53454.47	1944	100	295	3350	227	2572	3.54	66.67	1.00	53451.94
GA-SO	RELI	12	17538.26	1995	100	376	4137	268	2903	2.95	66.67	0.41	17535.72
GA-SO	RELI	13	28410.38	1975	100	370	858	196	422	2.95	66.67	0.41	28407.84
GA-SO	RELI	14	35368.66	1958	100	342	777	252	533	3.54	66.67	1.00	35366.12
GA-SO	RELI	15	23607.30	1981	100	332	787	244	547	3.54	66.67	1.00	23604.76
GA-SO	RELI	16	23582.04	1937	100	340	797	191	418	2.95	66.67	0.41	23579.50
GA-SO	RELI	17	37754.50	1986	100	265	636	187	412	3.54	66.67	1.00	37751.96
GA-SO	RELI	18	14740.98	2007	100	370	1191	243	855	3.54	66.67	1.00	14738.44
GA-SO	RELI	19	32627.02	2020	100	315	730	173	372	2.95	66.67	0.41	32624.48
GA-SO	RELI	20	34527.96	2058	100	320	748	199	430	3.54	66.67	1.00	34525.42
TS-SO	INV	1	2408.68	1188	100	265	1706.503	116	724.16	521.68	66.67	407.27	2294.27
TS-SO	INV	2	12516.10	1188	100	398	2565.852	329	2091.404	522.60	66.67	408.19	12401.68
TS-SO	INV	3	5913.89	1188	100	641	4052.316	496	3131.409	480.64	66.67	366.22	5799.47
TS-SO	INV	4	2375.05	1188	100	245	1567.868	146	914.097	428.10	33.33	313.69	2260.64
TS-SO	INV	5	5659.14	1188	100	344	4840.86	264	3701.86	1308.63	66.67	1194.21	5544.72
TS-SO	INV	6	1559.19	1188	100	303	4220.328	204	2840.718	451.70	0.00	337.29	1444.78
TS-SO	INV	7	4132.60	1188	100	542	7302.672	415	5605.609	436.71	66.67	322.29	4018.19
TS-SO	INV	8	3523.85	1188	100	236	3127.203	88	1160.563	347.08	66.67	232.67	3409.44
TS-SO	INV	9	2547.77	1188	100	380	4969.203	316	4125.484	470.09	66.67	355.68	2433.36
TS-SO	INV	10	7275.79	1188	100	241	1541.736	207	1289.092	1602.96	33.33	1488.55	7161.37
TS-SO	INV	11	1969.29	1188	100	161	1036.469	83	513.799	296.62	66.67	182.21	1854.88
TS-SO	INV	12	2907.54	1188	100	471	2960.128	358	2249.254	516.88	66.67	402.46	2793.13
TS-SO	INV	13	1686.18	1188	100	346	2190.764	226	1403.808	303.77	33.33	189.35	1571.77
TS-SO	INV	14	8952.61	1188	100	573	3612.303	452	2837.214	1239.72	33.33	1125.31	8838.20
TS-SO	INV	15	3301.83	1188	100	372	2387.906	318	2015.358	1040.54	33.33	926.12	3187.41
TS-SO	INV	16	2499.01	1188	100	273	1663.662	125	746.639	516.27	66.67	401.86	2384.60
TS-SO	INV	17	2123.57	1188	100	136	839.733	83	505.031	634.93	66.67	520.51	2009.16
TS-SO	INV	18	2353.56	1188	100	138	849.155	125	755.974	448.41	66.67	334.00	2239.14
TS-SO	INV	19	7898.51	1188	100	575	3503.543	300	1816.658	1150.85	33.33	1036.44	7784.09
TS-SO	INV	20	4911.45	1188	100	502	3064.587	423	2573.245	1025.57	33.33	911.16	4797.03

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
TS-SO	LOG	1	312.71	1980	100	181	860.372	107	477.753	290.64	0.00	79.39	101.47
TS-SO	LOG	2	314.77	1980	100	117	576.562	26	115.423	289.84	0.00	78.59	103.53
TS-SO	LOG	3	379.21	1980	100	202	948.535	67	294.194	288.55	0.00	77.31	167.96
TS-SO	LOG	4	345.31	1980	100	190	898.545	109	483.44	292.23	0.00	80.98	134.07
TS-SO	LOG	5	454.64	1980	100	175	832.563	48	213.471	308.03	0.00	96.78	243.40
TS-SO	LOG	6	318.70	1980	100	162	771.426	93	412.781	292.45	0.00	81.21	107.46
TS-SO	LOG	7	380.16	1980	100	164	783.783	76	338.326	313.73	0.00	102.49	168.92
TS-SO	LOG	8	387.03	1980	100	153	736.427	50	223.144	316.93	0.00	105.69	175.78
TS-SO	LOG	9	370.55	1980	100	179	847.714	102	452.487	294.77	0.00	83.52	159.30
TS-SO	LOG	10	477.10	1980	100	121	591.533	38	169.369	308.97	0.00	97.73	265.86
TS-SO	LOG	11	367.47	1980	100	208	2287.703	129	1408.219	311.64	0.00	100.40	156.22
TS-SO	LOG	12	365.64	1980	100	190	2047.703	111	1179.609	294.84	0.00	83.60	154.40
TS-SO	LOG	13	428.94	1980	100	374	4552.578	167	1807.109	304.51	0.00	93.26	217.69
TS-SO	LOG	14	466.43	1980	100	221	2414.531	161	1724.343	312.28	0.00	101.03	255.19
TS-SO	LOG	15	371.14	1980	100	184	2061.391	106	1165.407	294.28	0.00	83.03	159.89
TS-SO	LOG	16	316.16	1980	100	185	2114.297	45	491.922	287.11	0.00	75.86	104.91
TS-SO	LOG	17	353.96	1980	100	172	2013.375	49	569.89	313.71	0.00	102.47	142.71
TS-SO	LOG	18	356.76	1980	100	210	2489.172	99	1184.156	302.63	0.00	91.39	145.52
TS-SO	LOG	19	372.05	1980	100	226	2716.703	135	1594.297	313.97	0.00	102.72	160.81
TS-SO	LOG	20	442.12	1980	100	198	2342.781	81	936.969	307.51	0.00	96.26	230.87
TS-SO	PERT	1	0.1452	1188	100	58	699.704	32	377.516	0.1341	50.00	0.00	0.01
TS-SO	PERT	2	0.1480	1188	100	106	1256.313	53	608.516	0.1325	50.00	0.00	0.02
TS-SO	PERT	3	0.1493	1188	100	94	1107.812	47	563.672	0.1336	50.00	0.00	0.02
TS-SO	PERT	4	0.1505	1188	100	89	1070.344	52	615.922	0.1337	50.00	0.00	0.02
TS-SO	PERT	5	0.1402	1188	100	51	629.907	22	261.391	0.1334	50.00	0.00	0.01
TS-SO	PERT	6	0.1440	1188	100	78	977.719	24	288.625	0.1328	50.00	0.00	0.01
TS-SO	PERT	7	0.1432	1188	100	76	882.891	35	397.984	0.1331	50.00	0.00	0.01
TS-SO	PERT	8	0.1403	1188	100	60	712.093	23	265.656	0.1344	50.00	0.00	0.01
TS-SO	PERT	9	0.1394	1188	100	64	773.734	21	248.875	0.1338	50.00	0.00	0.01
TS-SO	PERT	10	0.1450	1188	100	62	750.468	27	313.859	0.1337	50.00	0.00	0.01

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
TS-SO	PERT	11	0.1421	1188	100	63	780	27	330.188	0.1348	50.00	0.00	0.01
TS-SO	PERT	12	0.1432	1188	100	55	694.704	16	200.5	0.1338	50.00	0.00	0.01
TS-SO	PERT	13	0.1399	1188	100	47	601.469	14	170.672	0.1348	50.00	0.00	0.01
TS-SO	PERT	14	0.1391	1188	100	70	860.406	19	229.562	0.1331	50.00	0.00	0.01
TS-SO	PERT	15	0.1523	1188	100	86	1083.266	46	568.719	0.1336	50.00	0.00	0.02
TS-SO	PERT	16	0.1524	1188	100	95	1208.625	54	682.422	0.1342	50.00	0.00	0.02
TS-SO	PERT	17	0.1452	1188	100	73	942.422	32	400.188	0.1339	50.00	0.00	0.01
TS-SO	PERT	18	0.1453	1188	100	52	669.188	14	174.266	0.1344	50.00	0.00	0.01
TS-SO	PERT	19	0.1506	1188	100	73	953.359	52	669.703	0.1330	50.00	0.00	0.02
TS-SO	PERT	20	0.1518	1188	100	91	1169.828	51	646.688	0.1341	50.00	0.00	0.02
TS-SO	PROD	1	50.33	2376	100	218	2829.609	114	1465.75	28.70	0.00	11.49	33.12
TS-SO	PROD	2	58.74	2376	100	261	3359.656	139	1775.125	24.50	0.00	7.28	41.53
TS-SO	PROD	3	57.06	2376	100	339	4351.61	180	2270.141	24.53	50.00	7.32	39.85
TS-SO	PROD	4	53.41	2376	100	205	2632.844	111	1406.39	27.48	0.00	10.27	36.19
TS-SO	PROD	5	46.19	2376	100	167	2196.547	88	1132.015	26.36	0.00	9.15	28.98
TS-SO	PROD	6	43.78	2376	100	191	2497.25	85	1093.547	25.88	0.00	8.67	26.57
TS-SO	PROD	7	41.80	2376	100	341	4433.578	58	752.344	24.98	0.00	7.77	24.59
TS-SO	PROD	8	56.13	2376	100	396	5119.797	137	1759.891	24.37	0.00	7.15	38.91
TS-SO	PROD	9	57.51	2376	100	427	5462.375	161	2064.344	27.80	50.00	10.59	40.30
TS-SO	PROD	10	52.72	2376	100	257	3213.047	138	1695.344	24.50	0.00	7.28	35.51
TS-SO	PROD	11	50.87	2376	100	365	4683.406	136	1723.984	27.14	50.00	9.92	33.66
TS-SO	PROD	12	39.69	2376	100	166	2153.078	76	969.703	27.51	0.00	10.29	22.48
TS-SO	PROD	13	55.33	2376	100	258	3321.532	127	1602.047	25.98	50.00	8.76	38.12
TS-SO	PROD	14	35.75	2376	100	158	2094.797	33	433.36	25.52	0.00	8.31	18.54
TS-SO	PROD	15	60.01	2376	100	403	5262.453	131	1678.984	28.34	0.00	11.13	42.80
TS-SO	PROD	16	58.73	2376	100	257	3384.203	149	1937.5	27.66	0.00	10.45	41.52
TS-SO	PROD	17	50.41	2376	100	254	3357.937	107	1387.578	26.02	0.00	8.81	33.20
TS-SO	PROD	18	40.19	2376	100	186	2426.188	37	455.735	24.04	0.00	6.83	22.98
TS-SO	PROD	19	46.15	2376	100	171	2206.719	95	1202.719	26.77	0.00	9.56	28.93
TS-SO	PROD	20	55.53	2376	100	220	2820.343	112	1409.859	23.55	0.00	6.34	38.32

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
TS-SO	RELI	1	9570.51	1584	100	114	1452.25	45	563.234	1536.71	0.00	1534.17	9567.97
TS-SO	RELI	2	11656.55	1584	100	160	2029.859	59	728.297	1330.08	0.00	1327.54	11654.01
TS-SO	RELI	3	18131.28	1584	100	140	1768.641	67	835.203	1286.31	0.00	1283.77	18128.74
TS-SO	RELI	4	17806.98	1584	100	163	1997.891	98	1198.563	85.82	33.33	83.28	17804.44
TS-SO	RELI	5	3507.21	1584	100	124	1502.297	40	464.718	112.54	33.33	110.00	3504.67
TS-SO	RELI	6	8337.71	1584	100	113	1412.578	55	672.515	187.77	33.33	185.23	8335.17
TS-SO	RELI	7	69591.39	1584	100	172	2153.5	131	1623.859	1664.63	0.00	1662.09	69588.85
TS-SO	RELI	8	12347.32	1584	100	166	2064.562	89	1083.828	654.79	0.00	652.25	12344.78
TS-SO	RELI	9	18852.50	1584	100	149	1889.063	79	982.797	1862.33	0.00	1859.79	18849.96
TS-SO	RELI	10	24195.69	1584	100	150	1932.281	84	1083.296	883.76	0.00	881.23	24193.15
TS-SO	RELI	11	32232.39	1584	100	147	1846.235	88	1090.172	55.97	33.33	53.43	32229.85
TS-SO	RELI	12	32485.83	1584	100	179	2204.266	93	1133	327.28	0.00	324.74	32483.29
TS-SO	RELI	13	3356.82	1584	100	114	1405.125	40	482.781	945.94	0.00	943.40	3354.28
TS-SO	RELI	14	13245.77	1584	100	115	1450.484	40	499.297	1608.27	0.00	1605.73	13243.23
TS-SO	RELI	15	27944.34	1584	100	240	3209.469	96	1313	187.89	33.33	185.35	27941.80
TS-SO	RELI	16	35079.54	1584	100	185	2406.438	83	1058.125	126.64	33.33	124.10	35077.00
TS-SO	RELI	17	16899.65	1584	100	124	1614.266	65	806.937	1718.93	0.00	1716.39	16897.11
TS-SO	RELI	18	25926.39	1584	100	160	2076.375	76	980.437	1666.63	0.00	1664.09	25923.85
TS-SO	RELI	19	27777.06	1584	100	154	2004.265	89	1137.421	233.39	0.00	230.85	27774.52
TS-SO	RELI	20	23843.71	1584	100	155	2073.969	93	1229.156	2.54	66.67	0.00	23841.17
LNM-SO	INV	1	9130.28	790	110	145	913.957	60	374.878	348.68	66.67	234.27	9015.87
LNM-SO	INV	2	1784.24	940	129	197	1242.719	108	673.638	234.18	66.67	119.77	1669.83
LNM-SO	INV	3	7036.38	1084	126	145	917.652	117	729.447	337.65	66.67	223.24	6921.96
LNM-SO	INV	4	4583.64	730	102	101	638.288	7	43.871	172.41	66.67	57.99	4469.23
LNM-SO	INV	5	2357.24	845	119	73	778.856	69	746	148.83	66.67	34.42	2242.83
LNM-SO	INV	6	5346.89	743	106	42	268.368	37	230.915	690.13	66.67	575.72	5232.48
LNM-SO	INV	7	2417.72	848	115	181	1139.955	75	467.097	540.71	66.67	426.29	2303.31
LNM-SO	INV	8	7374.40	730	104	35	224.266	15	93.943	148.83	66.67	34.42	7259.99
LNM-SO	INV	9	1214.89	994	127	195	1226.276	125	777.564	271.15	66.67	156.74	1100.48
LNM-SO	INV	10	6146.86	735	101	198	1240.026	14	91.008	1131.05	33.33	1016.63	6032.45

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
LNM-SO	INV	11	5585.78	1009	127	189	1183.866	115	712.022	242.03	66.67	127.61	5471.37
LNM-SO	INV	12	5275.37	715	102	24	154.758	5	31.474	706.01	33.33	591.59	5160.96
LNM-SO	INV	13	2662.44	1028	116	226	1430.394	74	459.668	395.54	66.67	281.13	2548.03
LNM-SO	INV	14	3227.48	967	133	195	1226.677	170	1058.841	185.15	66.67	70.73	3113.07
LNM-SO	INV	15	2512.39	1037	143	185	1474.386	182	1443.212	234.25	66.67	119.84	2397.98
LNM-SO	INV	16	3982.85	713	101	35	222.353	8	50.271	549.91	66.67	435.50	3868.43
LNM-SO	INV	17	1884.68	707	101	29	185.581	28	173.825	493.62	66.67	379.21	1770.26
LNM-SO	INV	18	4020.68	918	109	199	1249.649	31	192.261	526.24	66.67	411.83	3906.27
LNM-SO	INV	19	4956.10	1184	140	195	1227.639	177	1100.069	228.35	66.67	113.93	4841.69
LNM-SO	INV	20	7021.67	875	123	109	683.172	85	525.35	143.98	33.33	29.56	6907.25
LNM-SO	LOG	1	356.75	894	100	28	130.135	14	62.567	278.90	0.00	67.65	145.50
LNM-SO	LOG	2	383.21	891	100	32	145.739	14	61.394	278.90	0.00	67.65	171.96
LNM-SO	LOG	3	410.94	890	100	39	177.094	31	135.783	278.90	0.00	67.66	199.70
LNM-SO	LOG	4	390.51	893	100	43	193.075	15	65.346	278.90	0.00	67.65	179.26
LNM-SO	LOG	5	511.02	886	100	39	176.431	38	165.134	278.90	0.00	67.65	299.78
LNM-SO	LOG	6	474.80	884	100	48	215.479	43	185.935	278.91	0.00	67.66	263.55
LNM-SO	LOG	7	355.66	894	100	23	106.227	14	60.998	278.90	0.00	67.65	144.41
LNM-SO	LOG	8	412.89	893	100	21	97.356	19	82.23	278.90	0.00	67.65	201.64
LNM-SO	LOG	9	386.93	892	100	30	136.919	28	121.492	278.90	0.00	67.65	175.68
LNM-SO	LOG	10	339.21	891	100	26	120.315	20	87.809	278.90	0.00	67.65	127.96
LNM-SO	LOG	11	316.03	892	100	38	171.099	9	39.305	278.90	0.00	67.65	104.79
LNM-SO	LOG	12	331.39	893	100	28	127.668	24	103.815	278.90	0.00	67.65	120.14
LNM-SO	LOG	13	422.08	897	100	39	175.806	14	61.216	278.90	0.00	67.65	210.84
LNM-SO	LOG	14	399.65	893	100	30	136.241	28	120.839	278.90	0.00	67.65	188.41
LNM-SO	LOG	15	490.69	886	100	39	175.674	38	164.288	278.90	0.00	67.65	279.44
LNM-SO	LOG	16	432.13	891	100	33	151.981	28	123.221	278.90	0.00	67.65	220.89
LNM-SO	LOG	17	436.46	905	100	53	241.107	14	62.457	278.90	0.00	67.65	225.22
LNM-SO	LOG	18	462.56	893	100	27	125.756	26	114.65	278.90	0.00	67.65	251.32
LNM-SO	LOG	19	466.62	894	100	29	134.728	15	66.312	278.90	0.00	67.65	255.37
LNM-SO	LOG	20	404.00	887	100	37	170.268	36	158.903	278.90	0.00	67.65	192.76

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
LNM-SO	PERT	1	0.1561	696	100	17	77.92	10	43.571	0.1324	50.00	0.00	0.02
LNM-SO	PERT	2	0.1410	698	100	11	51.313	7	29.612	0.1324	50.00	0.00	0.01
LNM-SO	PERT	3	0.1616	699	100	9	43.031	7	30.042	0.1324	50.00	0.00	0.03
LNM-SO	PERT	4	0.1479	695	100	19	85.651	10	42.951	0.1324	50.00	0.00	0.02
LNM-SO	PERT	5	0.1460	696	100	17	76.348	10	42.08	0.1324	50.00	0.00	0.01
LNM-SO	PERT	6	0.1544	673	100	55	237.294	19	80.553	0.1324	50.00	0.00	0.02
LNM-SO	PERT	7	0.1425	698	100	11	50.792	7	29.332	0.1324	50.00	0.00	0.01
LNM-SO	PERT	8	0.1399	698	100	11	50.942	7	29.652	0.1324	50.00	0.00	0.01
LNM-SO	PERT	9	0.1457	695	100	22	97.998	10	42.079	0.1324	50.00	0.00	0.01
LNM-SO	PERT	10	0.1439	697	100	13	59.354	7	29.412	0.1324	50.00	0.00	0.01
LNM-SO	PERT	11	0.1421	676	100	154	653.2	153	644.217	0.1324	50.00	0.00	0.01
LNM-SO	PERT	12	0.2252	695	100	22	98.018	15	63.709	0.1324	50.00	0.00	0.09
LNM-SO	PERT	13	0.1358	698	100	11	50.802	7	29.401	0.1324	50.00	0.00	0.00
LNM-SO	PERT	14	0.1546	695	100	17	76.938	9	38.094	0.1324	50.00	0.00	0.02
LNM-SO	PERT	15	0.1503	692	100	25	110.025	9	38.184	0.1324	50.00	0.00	0.02
LNM-SO	PERT	16	0.1511	694	100	24	106.479	21	88.895	0.1324	50.00	0.00	0.02
LNM-SO	PERT	17	0.1517	697	100	16	72.092	12	50.461	0.1324	50.00	0.00	0.02
LNM-SO	PERT	18	0.1398	696	100	15	67.735	7	29.552	0.1324	50.00	0.00	0.01
LNM-SO	PERT	19	0.1504	694	100	21	97.517	16	71.541	0.1324	50.00	0.00	0.02
LNM-SO	PERT	20	0.1438	697	100	15	67.084	10	41.568	0.1324	50.00	0.00	0.01
LNM-SO	PROD	1	47.28	986	100	47	232.227	36	164.812	17.50	0.00	0.29	30.07
LNM-SO	PROD	2	60.13	974	100	82	387.776	58	263.111	17.61	0.00	0.40	42.92
LNM-SO	PROD	3	57.92	980	100	61	292.412	53	239.908	17.21	50.00	0.00	40.71
LNM-SO	PROD	4	33.99	983	100	60	288.807	10	44.974	17.86	0.00	0.65	16.78
LNM-SO	PROD	5	49.46	981	100	45	232.097	45	204.649	17.86	0.00	0.65	32.24
LNM-SO	PROD	6	44.11	966	100	87	416.066	73	330.986	17.39	0.00	0.17	26.90
LNM-SO	PROD	7	61.10	982	100	53	262.53	45	204.348	17.21	50.00	0.00	43.88
LNM-SO	PROD	8	58.19	987	100	35	187.504	10	44.663	17.86	0.00	0.65	40.98
LNM-SO	PROD	9	58.39	981	100	60	288.286	10	44.453	17.86	0.00	0.65	41.18
LNM-SO	PROD	10	47.04	980	100	51	258.595	46	208.674	17.47	0.00	0.26	29.82

Method	Domain	Run	Init_Z	Count	Iterations	ROCCM	CPU_CM	ROCBS	CPU_BS	BSF_Z	% Goals Sat	BSF_Zdif	Init_Zdif
LNM-SO	PROD	11	62.99	981	100	59	294.064	53	240.349	17.39	0.00	0.17	45.78
LNM-SO	PROD	12	61.69	981	100	58	279.163	10	44.743	17.86	0.00	0.65	44.48
LNM-SO	PROD	13	56.77	980	100	57	274.887	10	44.633	17.86	0.00	0.65	39.56
LNM-SO	PROD	14	46.63	987	100	37	198.95	34	156.47	17.41	0.00	0.19	29.42
LNM-SO	PROD	15	44.55	984	100	52	251.894	10	44.432	17.86	0.00	0.65	27.33
LNM-SO	PROD	16	46.41	986	100	51	247.709	10	44.653	17.86	0.00	0.65	29.19
LNM-SO	PROD	17	47.45	975	100	46	236.583	35	158.684	17.71	0.00	0.50	30.24
LNM-SO	PROD	18	57.24	980	100	60	288.035	10	44.603	17.86	0.00	0.65	40.02
LNM-SO	PROD	19	46.28	979	100	59	283.719	10	44.673	17.86	0.00	0.65	29.07
LNM-SO	PROD	20	59.07	981	100	58	278.803	45	203.296	17.64	0.00	0.42	41.85
LNM-SO	RELI	1	14730.00	830	104	27	121.411	17	72.843	54.01	33.33	51.47	14727.46
LNM-SO	RELI	2	18168.16	829	104	20	90.487	16	67.104	68.39	33.33	65.85	18165.62
LNM-SO	RELI	3	13660.24	860	108	28	124.114	25	104.827	15.75	33.33	13.21	13657.70
LNM-SO	RELI	4	22835.04	826	104	19	86.081	15	63.249	67.12	33.33	64.58	22832.50
LNM-SO	RELI	5	10563.76	794	100	19	85.63	18	75.576	137.78	33.33	135.24	10561.22
LNM-SO	RELI	6	14471.94	819	103	17	77.559	15	62.878	84.45	33.33	81.91	14469.40
LNM-SO	RELI	7	11334.02	1193	134	194	829.668	122	510.448	32.56	33.33	30.02	11331.48
LNM-SO	RELI	8	13570.83	830	104	25	111.557	17	71.511	73.70	33.33	71.16	13568.29
LNM-SO	RELI	9	32849.10	832	104	17	77.529	15	62.948	30.88	33.33	28.34	32846.56
LNM-SO	RELI	10	33499.59	1142	140	165	706.314	136	569.221	3.15	66.67	0.61	33497.05
LNM-SO	RELI	11	3008.98	806	102	35	156.77	32	137.443	38.88	33.33	36.34	3006.44
LNM-SO	RELI	12	44039.00	852	107	36	162.318	32	138.265	50.22	33.33	47.68	44036.46
LNM-SO	RELI	13	29330.84	836	105	27	121.942	18	76.898	34.82	33.33	32.28	29328.30
LNM-SO	RELI	14	13405.15	851	107	24	108.984	22	94.083	23.97	33.33	21.43	13402.61
LNM-SO	RELI	15	16074.05	844	106	33	147.438	19	80.984	27.55	33.33	25.01	16071.51
LNM-SO	RELI	16	19843.36	838	105	24	108.734	17	72.523	31.28	33.33	28.74	19840.82
LNM-SO	RELI	17	30644.72	845	106	30	134.85	21	89.496	69.16	33.33	66.62	30642.18
LNM-SO	RELI	18	2876.39	830	104	25	115.612	17	72.852	19.31	33.33	16.77	2873.85
LNM-SO	RELI	19	26155.84	846	106	24	327.05	19	274.827	25.74	33.33	23.20	26153.30
LNM-SO	RELI	20	4250.57	856	108	30	243.994	22	176.419	15.75	33.33	13.21	4248.03

APPENDIX E

SUMMARY OF INITIAL AND OPTIMAL SOLUTIONS

NOTE: Runs shown in boldface type correspond to the runs that generated the BSF for each domain overall methods.

Method	Domain	Run	Initial Solution	Init_Z	BSF	BSF_Z	BSF_Zdif	Init_Zdif
GA-SO	INV	1	[53.33, 0.00, 7.60]	15364.65	[95.33, 28.00, 2.80]	640.29	525.88	15250.24
GA-SO	INV	2	[48.67, 8.00, 7.60]	6834.55	[100.00, 22.00, 5.80]	1075.70	961.29	6720.14
GA-SO	INV	3	[62.67, 22.00, 5.80]	2060.34	[100.00, 26.00, 4.00]	782.91	668.50	1945.93
GA-SO	**INV**	**4**	**[67.33, 30.00, 5.80]**	**1282.95**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**1168.54**
GA-SO	**INV**	**5**	**[30.00, 0.00, 3.40]**	**8789.61**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**8675.20**
GA-SO	INV	6	[86.00, 16.00, 3.40]	2738.43	[86.00, 28.00, 4.00]	672.56	558.15	2624.02
GA-SO	INV	7	[58.00, 8.00, 3.40]	4428.45	[95.33, 28.00, 2.80]	640.29	525.88	4314.04
GA-SO	**INV**	**8**	**[53.33, 24.00, 3.40]**	**2244.65**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**2130.24**
GA-SO	INV	9	[81.33, 4.00, 3.40]	3668.44	[95.33, 28.00, 2.80]	640.29	525.88	3554.03
GA-SO	INV	10	[44.00, 20.00, 3.40]	3010.99	[95.33, 28.00, 2.80]	640.29	525.88	2896.58
GA-SO	INV	11	[90.67, 4.00, 8.20]	3871.14	[95.33, 28.00, 2.80]	640.29	525.88	3756.73
GA-SO	**INV**	**12**	**[90.67, 20.00, 8.20]**	**3063.61**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**2949.19**
GA-SO	INV	13	[72.00, 12.00, 8.20]	4343.56	[100.00, 26.00, 4.00]	782.91	668.50	4229.15
GA-SO	INV	14	[76.67, 8.00, 2.20]	4256.10	[48.67, 28.00, 3.40]	672.56	558.15	4141.69
GA-SO	INV	15	[90.67, 24.00, 2.20]	2031.39	[86.00, 28.00, 4.00]	672.56	558.15	1916.98
GA-SO	**INV**	**16**	**[53.33, 4.00, 2.20]**	**6219.04**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**6104.63**
GA-SO	**INV**	**17**	**[72.00, 20.00, 2.20]**	**3116.09**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**3001.68**
GA-SO	**INV**	**18**	**[81.33, 12.00, 2.20]**	**2318.39**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**2203.97**
GA-SO	INV	19	[67.33, 2.00, 2.20	5393.33	[95.33, 28.00, 2.80]	640.29	525.88	5278.92
GA-SO	**INV**	**20**	**[48.67, 18.00, 2.20]**	**5235.09**	**[100.00, 30.00, 4.60]**	**114.41**	**0.00**	**5120.68**
GA-SO	**LOG**	**1**	**[3, 2.87, 8.20, 9.33, 9.67]**	**213.22**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**1.97**
GA-SO	**LOG**	**2**	**[14, 2.33, 3.40, 8.00, 8.33]**	**215.97**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**4.73**
GA-SO	LOG	3	[4, 5.00, 4.00, 8.33, 7.33]	215.62	[13, 5.00, 10.00, 10.00, 7.00]	211.25	0.00	4.37
GA-SO	**LOG**	**4**	**[11, 5.00, 1.60, 7.33, 7.33]**	**217.03**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**5.78**
GA-SO	**LOG**	**5**	**[2, 5.00, 5.80, 9.67, 10.00]**	**214.89**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**3.64**
GA-SO	LOG	6	[6, 1.27, 7.60, 7.67, 9.33]	214.37	[15, 4.73, 10.00, 10.00, 7.00]	211.25	0.00	3.12
GA-SO	**LOG**	**7**	**[9, 3.93, 3.40, 7.00, 9.33]**	**216.50**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**5.26**
GA-SO	**LOG**	**8**	**[5, 2.07, 1.00, 8.00, 9.33]**	**217.42**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**6.18**
GA-SO	**LOG**	**9**	**[12, 1.00, 3.40, 9.33, 9.33]**	**215.59**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**4.34**
GA-SO	**LOG**	**10**	**[9, 2.07, 2.80, 7.33, 9.33]**	**216.74**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**5.49**

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
GA-SO	**LOG**	**11**	**[4, 1.00, 8.20, 6.00, 11.00]**	**214.67**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**3.43**
GA-SO	**LOG**	**12**	**[15, 4.73, 3.40, 7.67, 7.67]**	**216.06**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**4.81**
GA-SO	**LOG**	**13**	**[8, 3.67, 1.00, 5.67, 7.67]**	**218.18**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**6.93**
GA-SO	LOG	14	[11, 1.80, 6.40, 8.33, 7.67]	214.63	[13, 5.00, 10.00, 10.00, 7.00]	211.25	0.00	3.39
GA-SO	**LOG**	**15**	**[2, 1.53, 9.40, 9.67, 7.67]**	**212.28**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**1.03**
GA-SO	**LOG**	**16**	**[11, 1.00, 2.20, 9.67, 10.33]**	**216.11**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**4.86**
GA-SO	**LOG**	**17**	**[4, 4.47, 9.40, 7.67, 9.00]**	**213.20**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**1.95**
GA-SO	**LOG**	**18**	**[8, 4.47, 1.60, 9.00, 9.00]**	**216.64**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**5.40**
GA-SO	**LOG**	**19**	**[7, 1.00, 8.20, 8.33, 9.00]**	**213.60**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**2.35**
GA-SO	**LOG**	**20**	**[4, 2.07, 2.80, 8.00, 9.00]**	**216.45**	**[15, 5.00, 10.00, 10.00, 7.00]**	**211.25**	**0.00**	**5.20**
GA-SO	**PERT**	**1**	**[7.80, 7.53, 6.60]**	**0.1559**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0235**
GA-SO	**PERT**	**2**	**[12.00, 4.27, 1.00]**	**0.1495**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0171**
GA-SO	**PERT**	**3**	**[7.33, 8.00, 1.00]**	**0.1521**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0198**
GA-SO	**PERT**	**4**	**[8.73, 1.00, 4.73]**	**0.1412**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0089**
GA-SO	**PERT**	**5**	**[5.93, 2.87, 4.73]**	**0.1399**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0075**
GA-SO	**PERT**	**6**	**[9.20, 6.60, 4.73]**	**0.1516**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0192**
GA-SO	**PERT**	**7**	**[6.40, 1.93, 4.73]**	**0.1390**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0067**
GA-SO	**PERT**	**8**	**[5.47, 5.67, 4.73]**	**0.1450**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0127**
GA-SO	**PERT**	**9**	**[5.93, 3.80, 4.73]**	**0.1414**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0090**
GA-SO	**PERT**	**10**	**[6.40, 3.80, 4.73]**	**0.1418**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0095**
GA-SO	**PERT**	**11**	**[10.60, 1.47, 4.73]**	**0.1456**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0132**
GA-SO	**PERT**	**12**	**[11.07, 3.33, 4.73]**	**0.1489**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0165**
GA-SO	**PERT**	**13**	**[10.60, 7.07, 4.73]**	**0.1556**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0232**
GA-SO	**PERT**	**14**	**[6.40, 7.07, 4.73]**	**0.1503**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0180**
GA-SO	**PERT**	**15**	**[8.73, 6.13, 4.73]**	**0.1496**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0172**
GA-SO	**PERT**	**16**	**[6.87, 4.27, 4.73]**	**0.1432**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0109**
GA-SO	**PERT**	**17**	**[5.00, 8.00, 4.73]**	**0.1533**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0209**
GA-SO	**PERT**	**18**	**[10.60, 1.00, 2.87]**	**0.1431**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0107**
GA-SO	**PERT**	**19**	**[11.53, 4.73, 2.87]**	**0.1504**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0180**
GA-SO	**PERT**	**20**	**[9.67, 1.93, 2.87]**	**0.1418**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0094**

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
GA-SO	PROD	1	[8.33, 14.00, 7.00, 4.33, 3.60, 6.00]	51.25	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	34.03
GA-SO	PROD	2	[10.00, 17.33, 7.00, 5.40, 2.80, 4.13]	56.67	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	39.45
GA-SO	PROD	3	[5.33, 10.67, 12.33, 3.27, 2.20, 3.07]	35.11	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	17.90
GA-SO	PROD	4	[7.33, 12.00, 5.00, 4.33, 4.00, 3.07]	43.11	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	25.89
GA-SO	PROD	5	[8.00, 18.67, 9.00, 3.00, 2.00, 5.20]	53.39	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	36.18
GA-SO	PROD	6	[7.67, 14.00, 13.00, 5.13, 1.60, 3.60]	50.64	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	33.43
GA-SO	PROD	7	[9.33, 14.67, 14.33, 3.00, 1.40, 4.93]	48.80	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	31.59
GA-SO	PROD	8	[8.00, 12.67, 13.67, 4.33, 3.20, 4.93]	50.49	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	33.28
GA-SO	PROD	9	[5.67, 11.33, 11.00, 5.40, 2.20, 6.00]	46.10	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	28.89
GA-SO	PROD	10	[9.67, 11.33, 5.67, 6.20, 3.40, 4.13]	47.70	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	30.49
GA-SO	PROD	11	[8.00, 12.00, 10.33, 3.53, 4.00, 3.07]	45.52	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	28.30
GA-SO	PROD	12	[8.33, 18.67, 10.33, 6.47, 3.60, 5.20]	63.54	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	46.32
GA-SO	PROD	13	[7.33, 11.33, 10.33, 6.73, 3.80, 2.53]	50.91	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	33.69
GA-SO	PROD	14	[9.67, 14.00, 6.33, 4.33, 1.20, 4.67]	45.25	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	28.04
GA-SO	PROD	15	[9.00, 18.67, 11.00, 5.93, 3.80, 3.60]	63.01	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	45.80
GA-SO	PROD	16	[9.33, 19.33, 9.67, 5.13, 4.00, 3.60]	62.14	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	44.93
GA-SO	PROD	17	[8.67, 14.67, 7.00, 3.27, 3.80, 5.73]	50.30	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	33.08
GA-SO	PROD	18	[8.67, 18.67, 13.67, 5.13, 2.00, 2.27]	58.89	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	41.68
GA-SO	PROD	19	[5.67, 10.67, 9.67, 4.33, 3.80, 4.40]	42.21	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	25.00
GA-SO	PROD	20	[6.00, 18.00, 5.00, 6.73, 1.40, 3.33]	55.26	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	38.04
GA-SO	RELI	1	[15.00, 4.00, 8.80, 7.20]	36047.01	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	36044.47
GA-SO	RELI	2	[11.53, 7.60, 7.00, 14.13]	42248.44	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	42245.90
GA-SO	RELI	3	[4.60, 5.20, 10.00, 14.13]	17039.57	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	17037.04
GA-SO	RELI	4	[11.53, 3.40, 5.20, 3.73]	30483.82	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	30481.28
GA-SO	RELI	5	[8.07, 5.80, 9.40, 10.67]	25014.06	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	25011.52
GA-SO	RELI	6	[7.20, 2.80, 6.40, 6.33]	13941.52	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	13938.98
GA-SO	RELI	7	[3.73, 8.20, 7.00, 8.07]	24872.34	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	24869.80
GA-SO	RELI	8	[13.27, 4.60, 8.80, 8.07]	32793.57	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	32791.03
GA-SO	RELI	9	[8.93, 9.40, 3.40, 7.20]	48569.31	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	48566.77
GA-SO	RELI	10	[2.00, 10.00, 6.40, 13.27]	32113.11	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	32110.57

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
GA-SO	RELI	11	[11.53, 9.40, 4.60, 15.00]	53454.47	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	53451.94
GA-SO	RELI	12	[4.60, 5.20, 10.00, 15.00]	17538.26	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	17535.72
GA-SO	RELI	13	[7.20, 3.40, 1.60, 14.13]	28410.38	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	28407.84
GA-SO	RELI	14	[5.47, 8.20, 2.80, 14.13]	35368.66	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	35366.12
GA-SO	RELI	15	[5.47, 7.00, 7.60, 7.20]	23607.30	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	23604.76
GA-SO	RELI	16	[2.00, 5.80, 1.00, 14.13]	23582.04	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	23579.50
GA-SO	RELI	17	[14.13, 4.60, 5.80, 14.13]	37754.50	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	37751.96
GA-SO	RELI	18	[6.33, 2.80, 3.40, 14.13]	14740.98	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	14738.44
GA-SO	RELI	19	[6.33, 8.20, 7.00, 14.13]	32627.02	[2.00, 1.00, 9.40, 9.80]	2.95	0.41	32624.48
GA-SO	RELI	20	[11.53, 5.20, 4.60, 14.13]	34527.96	[2.00, 1.00, 9.40, 14.13]	3.54	1.00	34525.42
LNM-SO	INV	1	[32.14, 1.11, 7.33]	9130.28	[99.69, 29.87, 4.84]	348.68	234.27	9015.87
LNM-SO	INV	2	[72.52, 27.74, 6.24]	1784.24	[92.22, 27.10, 3.42]	234.18	119.77	1669.83
LNM-SO	INV	3	[57.28, 6.16, 6.02]	7036.38	[91.04, 29.53, 4.76]	337.65	223.24	6921.96
LNM-SO	INV	4	[72.97, 4.68, 8.01]	4583.64	[100.00, 30.00, 5.90]	172.41	57.99	4469.23
LNM-SO	INV	5	[70.49, 23.06, 1.54]	2357.24	[100.00, 30.00, 4.25]	148.83	34.42	2242.83
LNM-SO	INV	6	[72.71, 4.13, 6.07]	5346.89	[88.61, 30.00, 4.56]	690.13	575.72	5232.48
LNM-SO	INV	7	[79.85, 23.88, 2.09]	2417.72	[90.84, 25.69, 4.17]	540.71	426.29	2303.31
LNM-SO	INV	8	[30.56, 7.14, 7.26]	7374.40	[100.00, 30.00, 4.25]	148.83	34.42	7259.99
LNM-SO	INV	9	[93.69, 29.84, 3.57]	1214.89	[96.48, 29.90, 4.77]	271.15	156.74	1100.48
LNM-SO	INV	10	[62.77, 4.76, 1.10]	6146.86	[77.55, 24.00, 4.58]	1131.05	1016.63	6032.45
LNM-SO	INV	11	[62.67, 2.94, 6.07]	5585.78	[95.53, 29.89, 5.14]	242.03	127.61	5471.37
LNM-SO	INV	12	[75.42, 0.86, 6.80]	5275.37	[68.58, 30.00, 2.40]	706.01	591.59	5160.96
LNM-SO	INV	13	[66.67, 25.11, 6.88]	2662.44	[95.30, 26.71, 3.41]	395.54	281.13	2548.03
LNM-SO	INV	14	[87.78, 11.54, 8.47]	3227.48	[95.27, 29.40, 3.70]	185.15	70.73	3113.07
LNM-SO	INV	15	[79.24, 15.69, 5.88]	2512.39	[98.00, 29.12, 4.72]	234.25	119.84	2397.98
LNM-SO	INV	16	[86.77, 6.17, 7.08]	3982.85	[94.56, 30.00, 6.42]	549.91	435.50	3868.43
LNM-SO	INV	17	[84.79, 15.35, 1.89]	1884.68	[96.59, 30.00, 4.39]	493.62	379.21	1770.26
LNM-SO	INV	18	[56.41, 8.92, 4.36]	4020.68	[88.36, 25.01, 4.01]	526.24	411.83	3906.27
LNM-SO	INV	19	[49.18, 18.98, 8.60]	4956.10	[97.51, 28.93, 4.27]	228.35	113.93	4841.69
LNM-SO	INV	20	[42.75, 3.57, 4.12]	7021.67	[99.66, 29.88, 4.53]	143.98	29.56	6907.25

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
LNM-SO	LOG	1	[6, 1.32, 4.35, 5.18, 10.96]	356.75	[13, 4.46, 5.00, 10.00, 12.00]	278.90	67.65	145.50
LNM-SO	LOG	2	[7, 1.56, 3.24, 6.70, 11.41]	383.21	[13, 4.52, 5.00, 10.00, 12.00]	278.90	67.65	171.96
LNM-SO	LOG	3	[2, 4.86, 3.03, 8.32, 8.05]	410.94	[13, 3.85, 5.00, 10.00, 12.00]	278.90	67.66	199.70
LNM-SO	LOG	4	[6, 4.12, 3.35, 7.65, 8.52]	390.51	[13, 4.44, 5.00, 10.00, 12.00]	278.90	67.65	179.26
LNM-SO	LOG	5	[7, 2.09, 2.04, 6.18, 11.08]	511.02	[13, 4.33, 5.00, 10.00, 12.00]	**278.90**	67.65	299.78
LNM-SO	LOG	6	[5, 4.54, 2.04, 9.58, 11.87]	474.80	[15, 5.00, 5.00, 10.00, 12.00]	278.91	67.66	263.55
LNM-SO	LOG	7	[12, 3.67, 4.39, 5.39, 10.35]	355.66	[13, 4.52, 5.00, 10.00, 12.00]	278.90	67.65	144.41
LNM-SO	LOG	8	[10, 4.37, 2.77, 7.43, 10.78]	412.89	[13, 4.09, 5.00, 10.00, 12.00]	**278.90**	67.65	201.64
LNM-SO	LOG	9	[2, 2.23, 3.39, 7.96, 8.30]	386.93	[13, 4.58, 5.00, 10.00, 12.00]	278.90	67.65	175.68
LNM-SO	LOG	10	[6, 2.81, 3.89, 9.79, 9.24]	339.21	[13, 5.00, 5.00, 10.00, 12.00]	278.90	67.65	127.96
LNM-SO	LOG	11	[3, 4.75, 4.63, 9.79, 8.80]	316.03	[13, 5.00, 5.00, 10.00, 12.00]	**278.90**	67.65	104.79
LNM-SO	LOG	12	[13, 4.87, 4.05, 8.24, 10.63]	331.39	[13, 4.36, 5.00, 10.00, 12.00]	278.90	67.65	120.14
LNM-SO	LOG	13	[7, 4.66, 3.14, 5.97, 8.72]	422.08	[13, 4.34, 5.00, 10.00, 12.00]	278.90	67.65	210.84
LNM-SO	LOG	14	[8, 3.05, 3.51, 6.78, 7.81]	399.65	[13, 4.37, 5.00, 10.00, 12.00]	**278.90**	67.65	188.41
LNM-SO	LOG	15	[5, 1.59, 2.11, 7.01, 11.10]	490.69	[13, 4.40, 5.00, 10.00, 12.00]	**278.90**	67.65	279.44
LNM-SO	LOG	16	[13, 1.31, 2.77, 7.95, 8.17]	432.13	[13, 4.07, 5.00, 10.00, 12.00]	**278.90**	67.65	220.89
LNM-SO	LOG	17	[13, 3.43, 2.81, 6.65, 8.84]	436.46	[13, 4.56, 5.00, 10.00, 12.00]	278.90	67.65	225.22
LNM-SO	LOG	18	[6, 1.22, 2.78, 6.40, 7.19]	462.56	[13, 4.68, 5.00, 10.00, 12.00]	**278.90**	67.65	251.32
LNM-SO	LOG	19	[13, 3.91, 2.73, 5.34, 8.55]	466.62	[13, 4.47, 5.00, 10.00, 12.00]	278.90	67.65	255.37
LNM-SO	LOG	20	[3, 2.40, 3.06, 9.87, 7.79]	404.00	[13, 4.27, 5.00, 10.00, 12.00]	**278.90**	67.65	192.76
LNM-SO	**PERT**	**1**	**[9.10, 6.75, 7.35]**	**0.1561**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0237**
LNM-SO	**PERT**	**2**	**[8.17, 3.12, 3.18]**	**0.1410**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0087**
LNM-SO	**PERT**	**3**	**[11.59, 7.94, 5.53]**	**0.1616**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0292**
LNM-SO	**PERT**	**4**	**[8.80, 4.82, 5.53]**	**0.1479**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0155**
LNM-SO	**PERT**	**5**	**[10.43, 3.66, 3.12]**	**0.1460**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0137**
LNM-SO	**PERT**	**6**	**[6.43, 6.97, 7.49]**	**0.1544**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0221**
LNM-SO	**PERT**	**7**	**[8.04, 2.93, 4.75]**	**0.1425**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0101**
LNM-SO	**PERT**	**8**	**[8.52, 2.96, 1.56]**	**0.1399**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0075**
LNM-SO	**PERT**	**9**	**[6.98, 4.85, 5.54]**	**0.1457**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0134**
LNM-SO	**PERT**	**10**	**[6.92, 4.15, 5.27]**	**0.1439**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0115**

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
LNM-SO	**PERT**	**11**	**[5.32, 5.30, 3.10]**	**0.1421**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0098**
LNM-SO	**PERT**	**12**	**[5.65, 6.30, 2.47]**	**0.2252**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0928**
LNM-SO	**PERT**	**13**	**[7.13, 1.56, 1.42]**	**0.1358**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0035**
LNM-SO	**PERT**	**14**	**[10.11, 7.27, 3.93]**	**0.1546**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0222**
LNM-SO	**PERT**	**15**	**[11.85, 1.32, 5.83]**	**0.1503**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0179**
LNM-SO	**PERT**	**16**	**[9.40, 3.47, 7.83]**	**0.1511**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0188**
LNM-SO	**PERT**	**17**	**[8.53, 5.59, 6.98]**	**0.1517**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0194**
LNM-SO	**PERT**	**18**	**[8.32, 1.07, 4.10]**	**0.1398**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0074**
LNM-SO	**PERT**	**19**	**[7.91, 4.87, 7.41]**	**0.1504**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0180**
LNM-SO	**PERT**	**20**	**[9.34, 3.86, 2.87]**	**0.1438**	**[5.00, 1.00, 1.00]**	**0.1324**	**0.0000**	**0.0114**
LNM-SO	PROD	1	[6.60, 10.93, 11.27, 6.12, 2.58, 4.20]	47.28	[5.02, 10.00, 5.00, 3.00, 1.03, 2.03]	17.50	0.29	30.07
LNM-SO	PROD	2	[5.35, 18.70, 12.99, 6.86, 1.76, 2.69]	60.13	[5.00, 10.00, 5.00, 3.00, 1.02, 2.04]	17.61	0.40	42.92
LNM-SO	**PROD**	**3**	**[8.11, 18.48, 13.67, 3.88, 3.24, 2.28]**	**57.92**	**[5.00, 10.00, 5.00, 3.00, 1.00, 2.18]**	**17.21**	**0.00**	**40.71**
LNM-SO	PROD	4	[8.15, 11.72, 5.83, 3.50, 1.13, 4.43]	33.99	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	16.78
LNM-SO	PROD	5	[7.21, 14.65, 5.97, 5.60, 2.29, 3.18]	49.46	[5.00, 10.00, 5.02, 3.00, 1.00, 2.00]	17.86	0.65	32.24
LNM-SO	PROD	6	[5.52, 14.88, 7.47, 3.69, 1.31, 5.28]	44.11	[5.01, 10.00, 5.00, 3.00, 1.00, 2.00]	17.39	0.17	26.90
LNM-SO	**PROD**	**7**	**[9.59, 19.15, 14.64, 3.03, 3.20, 4.52]**	**61.10**	**[5.00, 10.00, 5.00, 3.00, 1.00, 2.18]**	**17.21**	**0.00**	**43.88**
LNM-SO	PROD	8	[9.15, 19.94, 5.17, 4.05, 3.06, 4.92]	58.19	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	40.98
LNM-SO	PROD	9	[9.27, 17.48, 9.03, 6.62, 1.89, 4.52]	58.39	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	41.18
LNM-SO	PROD	10	[6.03, 12.10, 10.68, 4.59, 3.19, 4.57]	47.04	[5.00, 10.00, 5.04, 3.02, 1.00, 2.00]	17.47	0.26	29.82
LNM-SO	PROD	11	[7.17, 16.40, 14.97, 6.80, 3.53, 5.75]	62.99	[5.01, 10.00, 5.00, 3.00, 1.00, 2.00]	17.39	0.17	45.78
LNM-SO	PROD	12	[5.41, 18.11, 14.07, 4.80, 3.42, 5.88]	61.69	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	44.48
LNM-SO	PROD	13	[5.11, 16.89, 14.21, 4.47, 3.20, 3.59]	56.77	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	39.56
LNM-SO	PROD	14	[6.29, 12.37, 10.51, 4.90, 3.35, 2.17]	46.63	[5.00, 10.00, 5.00, 3.00, 1.00, 2.06]	17.41	0.19	29.42
LNM-SO	PROD	15	[9.10, 11.93, 7.01, 5.92, 1.37, 4.35]	44.55	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	27.33
LNM-SO	PROD	16	[7.22, 16.39, 6.45, 3.51, 1.57, 3.49]	46.41	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	29.19
LNM-SO	PROD	17	[5.91, 11.28, 13.27, 6.82, 2.00, 2.11]	47.45	[5.00, 10.00, 5.04, 3.00, 1.03, 2.08]	17.71	0.50	30.24
LNM-SO	PROD	18	[5.85, 15.81, 14.29, 5.02, 3.50, 3.74]	57.24	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	40.02
LNM-SO	PROD	19	[8.65, 14.18, 10.27, 3.77, 1.45, 4.30]	46.28	[5.00, 10.00, 5.00, 3.00, 1.00, 2.00]	17.86	0.65	29.07
LNM-SO	PROD	20	[7.01, 19.80, 12.58, 4.30, 2.18, 2.30]	59.07	[5.18, 10.00, 5.00, 3.00, 1.01, 2.17]	17.64	0.42	41.85

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
LNM-SO	RELI	1	[8.16, 1.63, 5.03, 4.46]	14730.00	[2.00, 1.09, 10.00, 12.00]	54.01	51.47	14727.46
LNM-SO	RELI	2	[8.96, 2.88, 6.10, 8.84]	18168.16	[2.00, 1.00, 7.91, 9.48]	68.39	65.85	18165.62
LNM-SO	RELI	3	[5.09, 2.95, 2.86, 11.88]	13660.24	[2.00, 1.00, 10.00, 11.72]	15.75	13.21	13657.70
LNM-SO	RELI	4	[8.49, 4.52, 9.33, 4.60]	22835.04	[2.00, 1.00, 7.91, 10.09]	67.12	64.58	22832.50
LNM-SO	RELI	5	[2.80, 2.29, 1.63, 4.55]	10563.76	[2.00, 1.00, 8.53, 10.7]	137.78	135.24	10561.22
LNM-SO	RELI	6	[6.95, 2.89, 4.86, 9.63]	14471.94	[2.00, 1.00, 8.18, 9.78]	84.45	81.91	14469.40
LNM-SO	RELI	7	[3.84, 3.56, 3.26, 11.77]	11334.02	[2.01, 1.16, 9.44, 11.70]	32.56	30.02	11331.48
LNM-SO	RELI	8	[9.08, 1.22, 6.86, 5.17]	13570.83	[2.00, 1.00, 10.00, 9.23]	73.70	71.16	13568.29
LNM-SO	RELI	9	[13.14, 3.54, 4.81, 6.26]	32849.10	[2.00, 1.00, 9.25, 11.15]	30.88	28.34	32846.56
LNM-SO	RELI	10	[12.54, 4.84, 9.92, 4.88]	33499.59	[2.01, 1.13, 9.57, 10.70]	3.15	0.61	33497.05
LNM-SO	RELI	11	[3.63, 1.48, 9.54, 4.68]	3008.98	[2.00, 1.00, 10.00, 9.66]	38.88	36.34	3006.44
LNM-SO	RELI	12	[7.32, 4.99, 2.79, 2.38]	44039.00	[2.00, 1.21, 10.00, 11.98]	50.22	47.68	44036.46
LNM-SO	RELI	13	[7.93, 2.18, 1.84, 4.07]	29330.84	[2.14, 1.00, 9.03, 11.01]	34.82	32.28	29328.30
LNM-SO	RELI	14	[5.39, 4.17, 9.93, 5.91]	13405.15	[2.00, 1.00, 10.00, 11.75]	23.97	21.43	13402.61
LNM-SO	RELI	15	[6.86, 3.14, 8.20, 3.42]	16074.05	[2.00, 1.20, 10.00, 10.51]	27.55	25.01	16071.51
LNM-SO	RELI	16	[8.07, 4.25, 9.65, 6.34]	19843.36	[2.00, 1.00, 10.00, 10.32]	31.28	28.74	19840.82
LNM-SO	RELI	17	[14.39, 1.25, 4.24, 3.71]	30644.72	[2.08, 1.16, 8.99, 10.88]	69.16	66.62	30642.18
LNM-SO	RELI	18	[3.73, 1.36, 5.21, 10.55]	2876.39	[2.00, 1.00, 10.00, 11.66]	19.31	16.77	2873.85
LNM-SO	RELI	19	[9.06, 4.45, 8.95, 3.48]	26155.84	[2.00, 1.00, 10.00, 11.27]	25.74	23.20	26153.30
LNM-SO	RELI	20	[4.34, 1.05, 3.77, 10.74]	4250.57	[2.00, 1.00, 10.00, 11.74]	15.75	13.21	4248.03
TS-SO	INV	1	[86.72, 17.35, 8.77]	2408.68	[92.72, 24.35, 3.77]	521.68	407.27	2294.27
TS-SO	INV	2	[48.17, 0.51, 6.19]	12516.10	[63.17, 28.51, 3.19]	522.60	408.19	12401.68
TS-SO	INV	3	[73.39, 3.42, 1.68]	5913.89	[92.39, 28.42, 4.68]	480.64	366.22	5799.47
TS-SO	INV	4	[95.78, 19.46, 3.20]	2375.05	[94.78, 25.46, 4.20]	428.10	313.69	2260.64
TS-SO	INV	5	[33.28, 13.85, 5.09]	5659.14	[51.28, 29.85, 4.09]	1308.63	1194.21	5544.72
TS-SO	INV	6	[78.60, 24.99, 4.66]	1559.19	[87.60, 29.99, 3.66]	451.70	337.29	1444.78
TS-SO	INV	7	[97.71, 2.31, 6.86]	4132.60	[96.71, 26.31, 3.86]	436.71	322.29	4018.19
TS-SO	INV	8	[91.98, 18.92, 3.58]	3523.85	[95.98, 25.92, 2.58]	347.08	232.67	3409.44
TS-SO	INV	9	[74.51, 12.42, 4.78]	2547.77	[78.51, 27.42, 4.78]	470.09	355.68	2433.36
TS-SO	INV	10	[33.42, 20.60, 8.96]	7275.79	[42.42, 29.60, 2.96]	1602.96	1488.55	7161.37

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
TS-SO	INV	11	[89.74, 28.87, 5.31]	1969.29	[96.74, 29.87, 5.31]	296.62	182.21	1854.88
TS-SO	INV	12	[64.57, 28.60, 8.44]	2907.54	[82.57, 28.60, 5.44]	516.88	402.46	2793.13
TS-SO	INV	13	[93.47, 25.16, 8.04]	1686.18	[90.47, 29.16, 5.04]	303.77	189.35	1571.77
TS-SO	INV	14	[38.97, 1.70, 6.64]	8952.61	[47.97, 29.70, 4.64]	1239.72	1125.31	8838.20
TS-SO	INV	15	[37.37, 26.65, 6.19]	3301.83	[52.37, 27.65, 2.19]	1040.54	926.12	3187.41
TS-SO	INV	16	[75.14, 23.45, 7.86]	2499.01	[79.14, 28.45, 3.86]	516.27	401.86	2384.60
TS-SO	INV	17	[74.39, 26.45, 3.86]	2123.57	[77.39, 22.45, 3.86]	634.93	520.51	2009.16
TS-SO	INV	18	[79.95, 20.36, 2.85]	2353.56	[84.95, 26.36, 2.85]	448.41	334.00	2239.14
TS-SO	INV	19	[55.65, 13.89, 8.96]	7898.51	[59.65, 27.89, 2.96]	1150.85	1036.44	7784.09
TS-SO	INV	20	[47.79, 13.78, 9.74]	4911.45	[60.79, 28.78, 3.74]	1025.57	911.16	4797.03
TS-SO	LOG	1	[6, 2.08, 4.84, 7.54, 10.40]	312.71	[14, 4.08, 4.84, 9.54, 11.40]	290.64	79.39	101.47
TS-SO	LOG	2	[12, 4.85, 4.67, 6.66, 11.97]	314.77	[14, 4.85, 4.67, 9.66, 11.97]	289.84	78.59	103.53
TS-SO	LOG	3	[7, 2.69, 3.95, 5.47, 9.35]	379.21	[9, 4.69, 4.95, 9.47, 11.35]	288.55	77.31	167.96
TS-SO	LOG	4	[6, 3.10, 4.70, 7.96, 7.27]	345.31	[14, 4.10, 4.70, 9.96, 11.27]	292.23	80.98	134.07
TS-SO	LOG	5	[13, 1.22, 2.34, 9.65, 9.16]	454.64	[15, 4.22, 4.34, 9.65, 11.16]	308.03	96.78	243.40
TS-SO	LOG	6	[2, 2.49, 4.95, 9.05, 8.16]	318.70	[15, 4.49, 4.95, 9.05, 11.16]	292.45	81.21	107.46
TS-SO	LOG	7	[12, 2.29, 4.14, 5.62, 8.26]	380.16	[15, 4.29, 4.14, 9.62, 11.26]	313.73	102.49	168.92
TS-SO	LOG	8	[13, 4.13, 4.07, 5.31, 8.45]	387.03	[15, 4.13, 4.07, 9.31, 11.45]	316.93	105.69	175.78
TS-SO	LOG	9	[7, 4.98, 3.50, 6.73, 10.86]	370.55	[15, 4.98, 4.50, 9.73, 11.86]	294.77	83.52	159.30
TS-SO	LOG	10	[15, 2.59, 2.16, 8.50, 9.97]	477.10	[15, 4.59, 4.16, 9.50, 11.97]	308.97	97.73	265.86
TS-SO	LOG	11	[6, 4.86, 4.15, 5.02, 10.97]	367.47	[15, 4.86, 4.15, 9.02, 11.97]	311.64	100.40	156.22
TS-SO	LOG	12	[5, 2.07, 3.75, 8.49, 8.15]	365.64	[15, 4.07, 4.75, 9.49, 11.15]	294.84	83.60	154.40
TS-SO	LOG	13	[2, 2.04, 3.38, 5.46, 7.51]	428.94	[14, 4.04, 4.38, 9.46, 11.51]	304.51	93.26	217.69
TS-SO	LOG	14	[2, 3.86, 2.26, 7.10, 11.36]	466.43	[15, 4.86, 4.26, 9.10, 11.36]	312.28	101.03	255.19
TS-SO	LOG	15	[2, 4.69, 4.61, 5.98, 7.23]	371.14	[15, 4.69, 4.61, 9.98, 11.23]	294.28	83.03	159.89
TS-SO	LOG	16	[9, 4.54, 4.92, 6.08, 11.84]	316.16	[15, 4.54, 4.92, 9.08, 11.84]	287.11	75.86	104.91
TS-SO	LOG	17	[8, 4.71, 4.16, 7.39, 8.39]	353.96	[12, 4.71, 4.16, 9.39, 11.39]	313.71	102.47	142.71
TS-SO	LOG	18	[7, 3.02, 4.50, 6.39, 8.11]	356.76	[14, 4.02, 4.50, 9.39, 11.11]	302.63	91.39	145.52
TS-SO	LOG	19	[2, 2.46, 3.17, 8.66, 11.09]	372.05	[14, 4.46, 4.17, 9.66, 11.09]	313.97	102.72	160.81
TS-SO	LOG	20	[3, 3.43, 2.39, 9.07, 10.36]	442.12	[12, 4.43, 4.39, 9.07, 11.36]	307.51	96.26	230.87

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
TS-SO	PERT	1	[5.74, 4.76, 6.06]	0.1452	[5.74, 1.76, 1.06]	0.1341	0.0018	0.0129
TS-SO	PERT	2	[10.07, 4.01, 5.04]	0.1480	[5.07, 1.01, 1.04]	0.1325	0.0001	0.0156
TS-SO	PERT	3	[7.00, 5.49, 6.72]	0.1493	[5.00, 1.49, 1.72]	0.1336	0.0012	0.0169
TS-SO	PERT	4	[10.68, 3.11, 6.65]	0.1505	[5.68, 1.11, 1.65]	0.1337	0.0014	0.0181
TS-SO	PERT	5	[9.57, 1.15, 2.31]	0.1402	[5.57, 1.15, 1.31]	0.1334	0.0010	0.0079
TS-SO	PERT	6	[5.11, 3.23, 7.11]	0.1440	[5.11, 1.23, 1.11]	0.1328	0.0005	0.0116
TS-SO	PERT	7	[6.27, 3.15, 6.35]	0.1432	[5.27, 1.15, 1.35]	0.1331	0.0007	0.0109
TS-SO	PERT	8	[7.98, 2.28, 3.81]	0.1403	[5.98, 1.28, 1.81]	0.1344	0.0020	0.0079
TS-SO	PERT	9	[6.70, 1.58, 5.06]	0.1394	[5.70, 1.58, 1.06]	0.1338	0.0015	0.0070
TS-SO	PERT	10	[5.90, 5.34, 5.04]	0.1450	[5.90, 1.34, 1.04]	0.1337	0.0014	0.0127
TS-SO	PERT	11	[8.66, 1.82, 4.80]	0.1421	[5.66, 1.82, 1.80]	0.1348	0.0024	0.0097
TS-SO	PERT	12	[5.21, 1.66, 7.53]	0.1432	[5.21, 1.66, 1.53]	0.1338	0.0015	0.0108
TS-SO	PERT	13	[5.92, 3.60, 3.90]	0.1399	[5.92, 1.60, 1.90]	0.1348	0.0025	0.0075
TS-SO	PERT	14	[5.07, 2.53, 5.08]	0.1391	[5.07, 1.53, 1.08]	0.1331	0.0008	0.0067
TS-SO	PERT	15	[9.00, 7.32, 2.94]	0.1523	[5.00, 1.32, 1.94]	0.1336	0.0012	0.0199
TS-SO	PERT	16	[8.42, 6.98, 5.18]	0.1524	[5.42, 1.98, 1.18]	0.1342	0.0018	0.0201
TS-SO	PERT	17	[5.61, 4.46, 6.43]	0.1452	[5.61, 1.46, 1.43]	0.1339	0.0015	0.0128
TS-SO	PERT	18	[11.48, 1.66, 1.86]	0.1453	[5.48, 1.66, 1.86]	0.1344	0.0020	0.0129
TS-SO	PERT	19	[10.06, 6.09, 3.58]	0.1506	[5.06, 1.09, 1.58]	0.1330	0.0007	0.0182
TS-SO	PERT	20	[7.83, 6.34, 6.53]	0.1518	[5.83, 1.34, 1.53]	0.1341	0.0017	0.0195
TS-SO	PROD	1	[8.62, 14.59, 10.48, 3.93, 1.96, 5.52]	50.33	[5.62, 10.59, 5.48, 3.93, 1.96, 2.52]	28.70	11.49	33.1208
TS-SO	PROD	2	[7.20, 19.55, 8.95, 5.25, 3.20, 2.88]	58.74	[5.20, 10.55, 5.95, 3.25, 1.20, 2.88]	24.50	7.28	41.53
TS-SO	PROD	3	[7.75, 19.52, 12.72, 4.11, 1.48, 3.39]	57.06	[5.75, 10.52, 5.72, 3.11, 1.48, 2.39]	24.53	7.32	39.85
TS-SO	PROD	4	[6.86, 16.00, 6.37, 4.87, 3.92, 3.60]	53.41	[5.86, 10.00, 5.37, 3.87, 1.92, 2.60]	27.48	10.27	36.19
TS-SO	PROD	5	[8.25, 11.68, 6.74, 5.95, 3.02, 3.23]	46.19	[5.25, 10.68, 5.74, 3.95, 1.02, 2.23]	26.36	9.15	28.98
TS-SO	PROD	6	[6.08, 11.63, 13.61, 3.68, 3.38, 2.54]	43.78	[5.08, 10.63, 5.61, 3.68, 1.38, 2.54]	25.88	8.67	26.57
TS-SO	PROD	7	[7.16, 13.39, 5.44, 4.68, 1.00, 5.71]	41.80	[5.16, 10.39, 5.44, 3.68, 1.00, 3.71]	24.98	7.77	24.59
TS-SO	PROD	8	[8.64, 17.41, 9.21, 6.43, 1.52, 2.35]	56.13	[5.64, 10.41, 5.21, 3.43, 1.52, 2.35]	24.37	7.15	38.91
TS-SO	PROD	9	[8.17, 15.45, 13.79, 4.98, 3.61, 3.64]	57.51	[5.17, 10.45, 5.79, 3.98, 1.61, 0.64]	27.80	10.59	40.30
TS-SO	PROD	10	[8.37, 15.96, 11.11, 5.35, 1.09, 3.42]	52.72	[5.37, 10.96, 5.11, 3.35, 1.09, 2.42]	24.50	7.28	35.51

Method	Domain	Run	Initial Solution	Init Z	BSF	BSF_Z	BSF_Zdif	Init Zdif
TS-SO	PROD	11	[6.06, 15.74, 12.26, 3.80, 2.48, 3.67]	50.87	[5.06, 10.74, 5.26, 3.80, 1.48, 2.67]	27.14	9.92	33.66
TS-SO	PROD	12	[5.61, 11.32, 10.72, 4.75, 1.90, 3.12]	39.69	[5.61, 10.32, 5.72, 3.75, 1.90, 2.12]	27.51	10.29	22.48
TS-SO	PROD	13	[8.83, 17.51, 6.34, 3.40, 3.71, 5.78]	55.33	[5.83, 10.51, 5.34, 3.40, 1.71, 2.78]	25.98	8.76	38.12
TS-SO	PROD	14	[7.20, 12.53, 5.72, 3.64, 1.35, 4.30]	35.75	[5.20, 10.53, 5.72, 3.64, 1.35, 2.30]	25.52	8.31	18.54
TS-SO	PROD	15	[5.19, 19.94, 7.99, 6.54, 1.62, 5.84]	60.01	[5.19, 10.94, 5.99, 3.54, 1.62, 2.84]	28.34	11.13	42.80
TS-SO	PROD	16	[5.70, 19.95, 10.67, 4.00, 2.87, 4.87]	58.73	[5.70, 10.95, 5.67, 3.00, 1.87, 2.87]	27.66	10.45	41.52
TS-SO	PROD	17	[9.58, 14.11, 6.45, 4.89, 2.50, 5.67]	50.41	[5.58, 10.11, 5.45, 3.89, 1.50, 2.67]	26.02	8.81	33.20
TS-SO	PROD	18	[5.54, 10.16, 5.77, 5.12, 3.71, 4.05]	40.19	[5.54, 10.16, 5.77, 3.12, 1.71, 3.05]	24.04	6.83	22.98
TS-SO	PROD	19	[6.65, 12.75, 8.30, 4.19, 3.79, 3.98]	46.15	[5.65, 10.75, 5.30, 3.19, 1.79, 2.98]	26.77	9.56	28.93
TS-SO	PROD	20	[5.42, 15.14, 7.24, 6.42, 3.31, 5.91]	55.53	[5.42, 10.14, 5.24, 3.42, 1.31, 2.91]	23.55	6.34	38.32
TS-SO	RELI	1	[3.64, 3.80, 5.92, 11.13]	9570.51	[2.64, 1.80, 9.92, 8.13]	1536.71	1534.17	9567.97
TS-SO	RELI	2	[4.29, 3.94, 5.16, 11.32]	11656.55	[2.29, 1.94, 9.16, 8.32]	1330.08	1327.54	11654.01
TS-SO	RELI	3	[5.33, 4.87, 4.29, 11.64]	18131.28	[2.33, 1.87, 8.29, 8.64]	1286.31	1283.77	18128.74
TS-SO	RELI	4	[5.07, 4.21, 4.35, 3.87]	17806.98	[2.07, 1.21, 9.35, 9.87]	85.82	83.28	17804.44
TS-SO	RELI	5	[3.10, 2.27, 9.17, 6.99]	3507.21	[2.10, 1.27, 7.17, 10.99]	112.54	110.00	3504.67
TS-SO	RELI	6	[2.41, 4.28, 5.99, 6.48]	8337.71	[2.41, 1.28, 8.99, 9.48]	187.77	185.23	8335.17
TS-SO	RELI	7	[7.82, 2.66, 1.02, 2.01]	69591.39	[2.82, 1.66, 9.02, 7.01]	1664.63	1662.09	69588.85
TS-SO	RELI	8	[7.02, 1.82, 4.62, 4.67]	12347.32	[2.02, 1.82, 7.62, 8.67]	654.79	652.25	12344.78
TS-SO	RELI	9	[3.62, 4.91, 3.32, 3.85]	18852.50	[2.62, 1.91, 9.32, 7.85]	1862.33	1859.79	18849.96
TS-SO	RELI	10	[10.54, 3.60, 5.95, 9.26]	24195.69	[2.54, 1.60, 8.95, 8.26]	883.76	881.23	24193.15
TS-SO	RELI	11	[6.01, 3.55, 2.14, 2.65]	32232.39	[2.01, 1.55, 7.14, 10.65]	55.97	53.43	32229.85
TS-SO	RELI	12	[8.21, 2.56, 2.25, 2.78]	32485.83	[2.21, 1.56, 7.25, 7.78]	327.28	324.74	32483.29
TS-SO	RELI	13	[3.92, 1.29, 4.50, 11.37]	3356.82	[2.92, 1.29, 8.50, 10.37]	945.94	943.40	3354.28
TS-SO	RELI	14	[5.84, 3.68, 9.12, 9.15]	13245.77	[2.84, 1.68, 9.12, 8.15]	1608.27	1605.73	13243.23
TS-SO	RELI	15	[12.52, 3.13, 6.94, 5.85]	27944.34	[2.52, 1.13, 8.94, 8.85]	187.89	185.35	27941.80
TS-SO	RELI	16	[10.21, 4.10, 2.29, 11.56]	35079.54	[2.21, 1.10, 8.29, 11.56]	126.64	124.10	35077.00
TS-SO	RELI	17	[8.95, 2.62, 7.49, 11.14]	16899.65	[2.95, 1.62, 8.49, 8.14]	1718.93	1716.39	16897.11
TS-SO	RELI	18	[9.59, 3.83, 8.57, 3.12]	25926.39	[2.59, 1.83, 7.57, 7.12]	1666.63	1664.09	25923.85
TS-SO	RELI	19	[9.04, 4.60, 8.99, 3.11]	27777.06	[2.04, 1.60, 6.99, 8.11]	233.39	230.85	27774.52
TS-SO	**RELI**	**20**	**[8.44, 3.08, 2.61, 10.28]**	**23843.71**	**[2.44, 1.08, 9.61, 6.28]**	**2.54**	**0.00**	**23841.17**

APPENDIX F

COMPLETE DATA FOR SOLUTIONS SIMULATED FOR THE RUN CORRESPONDING TO THE BEST SOLUTION FOUND FOR EACH METHOD AND DOMAIN

GA-SO Method

Note that the GA-SO method was programmed as a maximization problem. Thus, values presented below should be multiplied by negative one to allow for comparison with the other two methods.

Inventory Domain: GA-SO Run Resulting in the BSF (Run 4)

Calls	CPU Time	Q	R	TBREV	LT	TBD	Z	Goal1	Goal2	Goal3	TS	TO	TR	AvgInv	Runtime	LS	BO	BWT	CWT
1	7	67.33	30	5.8	0	0	-1282.95	10681.26	-0.1420	-0.2250	590	8	19	43.42	107.9	136	36	0.1420	0.0000
2	13	100	12	8.2	0	0	-3142.89	8360.05	-0.1048	-0.2250	519	5	14	34.64	107.9	173	47	0.1048	0.1877
3	19	95.33	20	2.8	0	0	-2364.26	9500.81	-0.1908	-0.2250	572	6	39	44.46	107.9	145	41	0.1908	0.0199
4	25	81.33	30	7.6	0	0	-1124.34	10839.19	-0.1285	-0.2250	580	7	15	48.22	107.9	126	32	0.1285	0.0000
5	31	72	24	9.4	0	0	-3609.32	8038.87	-0.2041	-0.2658	540	7	12	25.79	107.9	208	58	0.2041	0.2658
6	38	53.33	0	9.4	0	0	-15381.24	-6457.48	-0.0500	-0.2250	125	1	12	5.75	107.9	396	125	0.0088	0.0319
7	44	67.33	16	1	0	0	-4571.88	6915.02	-0.2172	-0.2250	596	8	108	34.70	107.9	192	49	0.2172	0.0000
8	50	72	2	7.6	0	0	-4955.12	6360.63	-0.1776	-0.3242	492	7	15	21.71	107.9	224	65	0.1776	0.3242
9	57	58	2	8.2	0	0	-7802.11	3163.44	-0.2687	-0.4304	398	7	14	15.19	107.9	263	75	0.2687	0.4304
10	63	72	8	4.6	0	0	-3576.73	8083.04	-0.2063	-0.2250	515	7	24	27.71	107.9	164	43	0.2063	0.1977
11	69	100	22	9.4	0	0	-1967.64	9700.75	-0.0758	-0.2250	580	6	12	44.61	107.9	182	48	0.0758	0.1537
12	75	86	28	2.8	0	0	-1159.75	10762.86	-0.1144	-0.2250	593	7	39	51.30	107.9	111	30	0.1144	0.0134
13	81	44	14	6.4	0	0	-6274.01	4989.60	-0.2657	-0.3613	463	10	17	12.51	107.9	249	59	0.2657	0.3613
14	87	30	4	4	0	0	-8468.81	2330.45	-0.2552	-0.3271	409	12	27	10.53	107.9	281	86	0.2552	0.3271
15	94	100	30	2.2	0	0	-1330.77	10695.15	-0.1717	-0.2250	600	6	50	57.57	107.9	108	32	0.1717	0.0190
16	100	39.33	0	3.4	0	0	-16580.70	-7896.84	-0.0500	-0.2250	111	2	32	4.26	107.9	422	117	0.0226	0.0471
17	106	53.33	28	4.6	0	0	-2458.84	9258.67	-0.1372	-0.2250	581	10	24	37.84	107.9	181	41	0.1372	0.0000
18	112	30	28	2.2	0	0	-5622.86	6124.73	-0.4134	-0.2250	567	18	50	23.80	107.9	250	64	0.4134	0.1354
19	119	81.33	6	8.8	0	0	-4586.78	6643.46	-0.1115	-0.2250	479	6	13	26.73	107.9	202	59	0.1115	0.0466
20	125	90.67	8	5.8	0	0	-2212.59	9429.37	-0.0852	-0.2250	569	6	19	36.22	107.9	173	45	0.0852	0.0620
21	131	53.33	4	1	0	0	-6904.80	4120.56	-0.2193	-0.2250	522	9	108	21.88	107.9	231	61	0.2193	0.0000
22	137	48.67	12	2.8	0	0	-4439.73	7040.72	-0.2035	-0.2250	544	10	39	21.50	107.9	217	54	0.2035	0.0309
23	143	76.67	8	8.8	0	0	-5426.37	5728.60	-0.1501	-0.2250	456	6	13	26.41	107.9	214	69	0.1501	0.0312
24	150	76.67	22	5.2	0	0	-2399.53	9318.81	-0.1326	-0.2250	561	7	21	37.51	107.9	163	45	0.1326	0.0000
25	156	53.33	0	5.8	0	0	-15322.70	-6387.24	-0.0500	-0.2250	125	1	19	5.19	107.9	388	113	0.0074	0.0370
26	162	72	20	3.4	0	0	-1862.32	9962.98	-0.1324	-0.2250	586	8	32	39.44	107.9	144	43	0.1324	0.0346
27	168	48.67	18	2.8	0	0	-4010.29	7810.45	-0.3095	-0.2250	566	11	39	22.13	107.9	204	53	0.3095	0.0066
28	174	58	16	7.6	0	0	-5403.57	5816.47	-0.1752	-0.2764	469	8	15	19.76	107.9	223	59	0.1752	0.2764
29	180	58	2	6.4	0	0	-5773.27	5553.49	-0.2502	-0.4010	490	8	17	15.08	107.9	243	92	0.2502	0.4010
30	186	67.33	26	2.8	0	0	-2815.22	9067.41	-0.2357	-0.2250	597	9	39	40.75	107.9	187	48	0.2357	0.0000
31	193	30	30	2.8	0	0	-4384.67	7614.40	-0.4150	-0.2250	583	18	39	24.57	107.9	214	60	0.4150	0.0929
32	199	48.67	16	2.2	0	0	-3454.05	8339.46	-0.2518	-0.2250	558	11	50	24.43	107.9	156	43	0.2518	0.0000
33	205	86	30	2.2	0	0	-1157.74	10816.63	-0.1358	-0.2250	614	7	50	54.14	107.9	117	32	0.1358	0.0000
34	211	100	28	2.8	0	0	-792.38	11347.22	-0.1742	-0.2250	594	6	39	52.77	107.9	87	27	0.1742	0.0982
35	218	72	0	1	0	0	-5911.94	5087.35	-0.1257	-0.2250	528	7	108	29.31	107.9	198	55	0.1257	0.1343

Calls	CPU Time	Q	R	TBREV	LT	TBD	Z	Goal1	Goal2	Goal3	TS	TO	TR	AvgInv	Runtime	LS	BO	BWT	CWT
36	224	67.33	24	4.6	0	0	-2756.63	8827.40	-0.1064	-0.2250	587	8	24	40.78	107.9	209	53	0.1064	0.0000
37	230	67.33	30	8.2	0	0	-3170.85	8456.82	-0.1591	-0.2250	560	8	14	30.53	107.9	209	53	0.1591	0.0000
38	236	90.67	28	4.6	0	0	-774.02	11189.49	-0.0993	-0.2250	580	6	24	47.52	107.9	101	25	0.0993	0.0000
39	243	34.67	20	3.4	0	0	-4689.96	7197.65	-0.3940	-0.2250	549	15	32	17.03	107.9	217	47	0.3940	0.0517
40	249	30	12	2.2	0	0	-5681.02	6124.77	-0.4424	-0.2700	516	16	50	12.11	107.9	196	55	0.4424	0.2700
41	255	53.33	6	1	0	0	-5214.31	5954.03	-0.1380	-0.2250	541	9	108	25.41	107.9	170	47	0.1380	0.0351
42	261	44	14	5.8	0	0	-6376.08	4872.71	-0.2682	-0.2858	490	10	19	16.90	107.9	271	88	0.2682	0.2858
43	268	67.33	30	6.4	0	0	-4524.81	7508.63	-0.4410	-0.2250	562	8	17	26.34	107.9	251	60	0.4410	0.1833
44	274	76.67	8	5.8	0	0	-4026.28	7397.43	-0.1454	-0.2250	500	6	19	29.13	107.9	186	50	0.1454	0.0185
45	280	90.67	22	5.2	0	0	-1497.51	10425.71	-0.1428	-0.2250	575	6	21	44.70	107.9	132	36	0.1428	0.0000
46	286	48.67	16	1	0	0	-4205.76	7353.17	-0.2167	-0.2250	581	11	108	27.41	107.9	149	39	0.2167	0.0000
47	293	67.33	30	7	0	0	-2728.14	8946.31	-0.1417	-0.2250	552	8	16	31.72	107.9	172	52	0.1417	0.0000
48	299	95.33	20	8.2	0	0	-3977.34	7464.79	-0.1490	-0.2250	540	6	14	35.25	107.9	238	54	0.1490	0.0505
49	305	100	14	2.8	0	0	-2073.02	9568.78	-0.0735	-0.2250	580	6	39	49.17	107.9	152	40	0.0735	0.0340
50	311	100	22	2.2	0	0	-1326.22	10749.34	-0.1920	-0.2250	590	6	50	47.87	107.9	96	26	0.1920	0.0685
51	317	30	30	2.2	0	0	-3778.37	8535.64	-0.4957	-0.2250	587	19	50	21.70	107.9	162	44	0.4957	0.1685
52	323	72	20	4	0	0	-1613.84	10339.63	-0.1651	-0.2250	581	8	27	35.62	107.9	130	36	0.1651	0.1084
53	330	48.67	24	2.2	0	0	-3315.29	8567.65	-0.2775	-0.2250	582	12	50	28.62	107.9	169	49	0.2775	0.0000
54	336	95.33	28	4.6	0	0	-1059.36	10973.33	-0.1519	-0.2250	599	6	24	50.20	107.9	126	44	0.1519	0.0000
55	342	90.67	20	8.2	0	0	-3063.61	8529.11	-0.1356	-0.2250	545	6	14	36.81	107.9	195	46	0.1356	0.0133
56	348	100	30	2.8	0	0	-1449.65	10447.37	-0.1279	-0.2250	593	6	39	64.01	107.9	128	32	0.1279	0.0292
57	354	86	14	2.8	0	0	-3188.30	8534.76	-0.2003	-0.2250	572	6	39	35.31	107.9	192	44	0.2003	0.1259
58	361	90.67	22	4.6	0	0	-1922.90	9902.52	-0.1375	-0.2250	578	6	24	40.51	107.9	158	35	0.1375	0.0154
59	367	86	28	4.6	0	0	-1618.50	10329.48	-0.1632	-0.2250	590	7	24	47.55	107.9	145	43	0.1632	0.0273
60	373	90.67	30	2.2	0	0	-1676.55	10433.83	-0.2357	-0.2250	603	7	50	49.06	107.9	124	29	0.2357	0.0166
61	379	48.67	30	2.2	0	0	-2162.82	10015.89	-0.3047	-0.2250	602	12	50	32.68	107.9	128	36	0.3047	0.0000
62	385	86	16	2.2	0	0	-1769.52	10123.05	-0.1527	-0.2250	602	7	50	39.27	107.9	135	36	0.1527	0.1008
63	392	86	28	1.6	0	0	-1788.54	10283.59	-0.2291	-0.2250	595	7	68	49.61	107.9	95	28	0.2291	0.0807
64	398	67.33	30	2.2	0	0	-1910.69	10049.66	-0.1927	-0.2250	594	9	50	47.13	107.9	123	37	0.1927	0.0044
65	404	90.67	20	4	0	0	-2110.23	9692.12	-0.1435	-0.2250	592	6	27	41.43	107.9	175	48	0.1435	0.0025
66	410	86	20	8.2	0	0	-4163.13	7468.89	-0.2434	-0.3167	538	6	14	25.74	107.9	235	55	0.2434	0.3167
67	417	90.67	28	2.2	0	0	-1690.74	10256.95	-0.1691	-0.2250	598	7	50	53.40	107.9	125	32	0.1691	0.0000
68	423	90.67	28	2.8	0	0	-2023.15	9888.54	-0.1818	-0.2250	600	7	39	49.72	107.9	162	33	0.1818	0.0000
69	430	100	30	1	0	0	-2197.15	9521.10	-0.1157	-0.2250	596	6	108	66.95	107.9	80	21	0.1157	0.0000
70	436	90.67	8	7	0	0	-3789.43	7823.01	-0.2043	-0.2250	515	6	16	27.79	107.9	189	48	0.2043	0.0346

Calls	CPU Time	Q	R	TBREV	LT	TBD	Z	Goal1	Goal2	Goal3	TS	TO	TR	AvgInv	Runtime	LS	BO	BWT	CWT
71	442	86	20	4	0	0	-2370.01	9409.43	-0.1556	-0.2250	583	6	27	36.09	107.9	171	62	0.1556	0.1087
72	448	48.67	28	2.8	0	0	-2249.01	9648.46	-0.1947	-0.2250	589	12	39	38.32	107.9	147	35	0.1947	0.0000
73	454	100	26	2.2	0	0	-2009.23	10230.93	-0.3175	-0.2250	592	6	50	44.19	107.9	117	41	0.3175	0.1817
74	461	86	28	2.2	0	0	-1703.26	10018.97	-0.0762	-0.2250	597	7	50	60.31	107.9	134	35	0.0762	0.0000
75	467	86	30	4.6	0	0	-2258.63	9616.28	-0.1861	-0.2250	587	7	24	43.38	107.9	179	39	0.1861	0.0812
76	473	86	14	2.2	0	0	-2659.75	8915.34	-0.0946	-0.2250	603	7	50	45.55	107.9	187	55	0.0946	0.0202
77	479	95.33	28	2.2	0	0	-1552.13	10518.81	-0.2089	-0.2250	602	6	50	49.10	107.9	115	41	0.2089	0.0204
78	486	90.67	28	1.6	0	0	-2704.96	9248.93	-0.2562	-0.2250	598	7	68	48.81	107.9	147	35	0.2562	0.0000
79	492	90.67	20	5.2	0	0	-2334.29	9395.89	-0.1321	-0.2250	575	6	21	44.27	107.9	179	45	0.1321	0.0000
80	498	100	28	2.2	0	0	-1098.34	10741.27	-0.0747	-0.2250	596	6	50	66.57	107.9	102	29	0.0747	0.0000
81	504	95.33	30	2.2	0	0	-1122.13	11093.85	-0.2335	-0.2250	597	6	50	49.01	107.9	88	25	0.2335	0.0000
82	510	86	30	7	0	0	-1297.51	10747.54	-0.1769	-0.2250	589	7	16	40.44	107.9	139	35	0.1769	0.0617
83	517	76.67	28	2.2	0	0	-2463.60	9446.40	-0.2178	-0.2250	593	8	50	43.11	107.9	157	32	0.2178	0.0000
84	523	90.67	22	2.2	0	0	-2107.15	9515.58	-0.0684	-0.2250	588	6	50	56.57	107.9	149	38	0.0684	0.0000
85	530	100	24	2.2	0	0	-2199.68	9611.67	-0.1547	-0.2250	593	6	50	54.06	107.9	147	45	0.1547	0.0203
86	536	100	28	4.6	0	0	-2697.99	8935.69	-0.1222	-0.2250	592	6	24	56.32	107.9	214	54	0.1222	0.0786
87	542	86	22	2.2	0	0	-1857.56	10171.97	-0.2171	-0.2250	596	7	50	41.78	107.9	121	44	0.2171	0.0460
88	549	95.33	28	5.2	0	0	-1307.56	10587.17	-0.1151	-0.2250	590	6	21	52.90	107.9	140	41	0.1151	0.0000
89	555	95.33	30	7	0	0	-1758.67	10118.79	-0.1455	-0.2250	576	6	16	49.13	107.9	153	42	0.1455	0.0557
90	561	86	30	2.8	0	0	-1378.75	10809.19	-0.2432	-0.2250	598	7	39	48.23	107.9	116	28	0.2432	0.0308
91	567	90.67	30	7	0	0	-1551.32	10597.54	-0.2413	-0.2250	605	7	16	40.47	107.9	162	44	0.2413	0.0217
92	574	95.33	14	2.2	0	0	-3464.50	8318.28	-0.2482	-0.2250	591	6	50	38.74	107.9	206	52	0.2482	0.0119
93	580	86	30	10	0	0	-3327.76	8178.29	-0.1215	-0.2250	559	6	11	38.71	107.9	230	55	0.1215	0.1540
94	586	86	28	1	0	0	-2892.79	8776.10	-0.1531	-0.2250	610	7	108	55.38	107.9	127	32	0.1531	0.0015
95	593	90.67	28	5.2	0	0	-2831.25	8752.74	-0.1126	-0.2250	585	6	21	51.58	107.9	216	60	0.1126	0.0020
96	599	86	28	9.4	0	0	-3080.46	8633.69	-0.1876	-0.2250	542	6	12	37.80	107.9	189	46	0.1876	0.1049
97	605	100	30	4.6	0	0	-114.41	11940.95	-0.0826	-0.2250	605	6	24	55.73	107.9	92	30	0.0826	0.0104
98	612	100	30	3.4	0	0	-1091.76	11005.73	-0.1816	-0.2250	592	6	32	56.23	107.9	110	31	0.1816	0.0609
99	618	95.33	20	4.6	0	0	-1769.15	10040.70	-0.1182	-0.2250	580	6	24	44.72	107.9	153	36	0.1182	0.0000
100	624	90.67	30	4.6	0	0	-798.97	11165.55	-0.1018	-0.2250	590	6	24	56.64	107.9	112	30	0.1018	0.0000
101	631	81.33	30	2.2	0	0	-1308.30	10971.00	-0.2754	-0.2250	602	7	50	44.03	107.9	96	29	0.2754	0.0000
102	637	100	14	4.6	0	0	-2229.13	9439.29	-0.0976	-0.2250	566	6	24	47.99	107.9	162	43	0.0976	0.0748
103	643	95.33	30	4.6	0	0	-1122.84	10770.91	-0.0993	-0.2250	589	6	24	55.81	107.9	128	35	0.0993	0.0000
104	649	100	30	7	0	0	-1183.00	10708.56	-0.1034	-0.2250	576	6	16	49.68	107.9	126	37	0.1034	0.0817
105	657	62.67	30	4.6	0	0	-2027.69	9949.34	-0.2094	-0.2250	585	9	24	36.29	107.9	155	39	0.2094	0.0405
106	663	100	20	2.2	0	0	-1417.05	10613.22	-0.1807	-0.2250	587	6	50	50.47	107.9	98	28	0.1807	0.1045
107	669	81.33	30	4.6	0	0	-1777.46	10250.08	-0.2096	-0.2250	588	7	24	46.15	107.9	150	36	0.2096	0.0000
108	675	53.33	30	4.6	0	0	-2833.77	9378.28	-0.3745	-0.2250	600	11	24	27.16	107.9	187	60	0.3745	0.0000
109	682	100	22	4.6	0	0	-1082.38	10724.18	-0.0596	-0.2250	591	6	24	48.23	107.9	132	37	0.0596	0.0560
110	688	95.33	28	7	0	0	-1114.81	10763.03	-0.0920	-0.2250	577	6	16	50.90	107.9	123	40	0.0920	0.0557

Calls	CPU Time	Q	R	TBREV	LT	TBD	Z	Goal1	Goal2	Goal3	TS	TO	TR	AvgInv	Runtime	LS	BO	BWT	CWT
111	694	90.67	30	9.4	0	0	-1958.57	9764.44	-0.0978	-0.2250	551	6	12	47.64	107.9	146	41	0.0978	0.0000
112	702	95.33	14	4.6	0	0	-2673.84	8851.15	-0.0749	-0.2250	546	6	24	43.68	107.9	167	42	0.0749	0.0125
113	708	100	30	9.4	0	0	-1940.43	9866.85	-0.1314	-0.2250	584	6	12	53.64	107.9	180	45	0.1314	0.0572
114	716	100	30	10	0	0	-2602.08	9095.91	-0.1410	-0.2250	564	5	11	55.82	107.9	196	48	0.1410	0.0476
115	723	100	26	4.6	0	0	-1155.63	10752.68	-0.1081	-0.2250	577	6	24	51.71	107.9	115	33	0.1081	0.0220
116	729	100	30	5.2	0	0	-1807.14	9971.35	-0.1083	-0.2250	581	6	21	54.79	107.9	160	40	0.1083	0.0000
117	736	81.33	30	5.2	0	0	-1115.37	10890.52	-0.1454	-0.2250	592	7	21	45.14	107.9	124	43	0.1454	0.0000
118	743	95.33	26	4.6	0	0	-1802.43	10215.81	-0.2078	-0.2250	583	6	24	44.47	107.9	150	32	0.2078	0.0000
119	753	100	14	5.2	0	0	-3849.33	7657.05	-0.1651	-0.2250	559	5	21	41.60	107.9	245	52	0.1651	0.1424
120	760	44	30	4.6	0	0	-3315.09	8818.21	-0.3818	-0.2250	592	13	24	27.31	107.9	199	62	0.3818	0.1035
121	769	58	30	4.6	0	0	-2051.67	10103.19	-0.2855	-0.2250	581	10	24	27.36	107.9	139	41	0.2855	0.1663
122	776	100	30	8.2	0	0	-3223.90	8337.47	-0.1359	-0.2250	564	6	14	52.19	107.9	220	62	0.1359	0.0195
123	789	100	22	3.4	0	0	-2498.87	9218.32	-0.1404	-0.2250	572	6	32	50.24	107.9	169	42	0.1404	0.0566
124	796	95.33	30	9.4	0	0	-2948.74	8711.35	-0.1541	-0.2250	568	6	12	43.21	107.9	214	53	0.1541	0.0659
125	804	100	28	5.2	0	0	-2449.94	9257.36	-0.1322	-0.2250	574	6	21	52.47	107.9	182	50	0.1322	0.0464
126	811	100	28	9.4	0	0	-2159.22	9665.26	-0.1568	-0.2250	589	6	12	45.74	107.9	193	52	0.1568	0.1653
127	819	100	26	3.4	0	0	-1264.92	10757.62	-0.1648	-0.2250	588	6	32	51.85	107.9	115	36	0.1648	0.0465
128	825	100	26	5.2	0	0	-1845.27	9936.15	-0.1127	-0.2250	578	6	21	55.25	107.9	154	48	0.1127	0.0056
129	838	62.67	26	4.6	0	0	-2696.73	9599.13	-0.3980	-0.2250	574	9	24	25.19	107.9	158	41	0.3980	0.0100
130	852	100	20	4.6	0	0	-1699.64	10013.23	-0.0720	-0.2250	571	6	24	50.44	107.9	139	44	0.0720	0.0560
131	859	81.33	26	4.6	0	0	-1703.35	10240.86	-0.1687	-0.2250	582	7	24	44.35	107.9	141	40	0.1687	0.0045

Logistics Domain: GA-SO Run Resulting in the BSF (Run 1)

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
1	15	3	2.87	8.2	9.33	9.67	-213.22	-1279.57	0.0212	21.31	17.50	10.74	7.50	8.47	8.81	0.3724	0.4223	0.4500	2.12	1.58
2	29	12	4.47	2.8	8	10	-216.56	-1293.51	0.0732	22.33	17.72	10.72	2.70	7.36	9.10	0.1350	0.3649	0.4650	7.31	3.77
3	46	3	1.27	7	5	10.67	-215.86	-1290.36	0.0175	21.56	18.24	10.65	6.49	4.75	9.60	0.3229	0.2357	0.4905	1.75	1.05
4	61	15	1	1	6.67	9.33	-217.93	-1299.11	0.0693	22.69	17.95	10.78	0.98	6.23	8.54	0.0491	0.3103	0.4351	6.93	4.65
5	76	8	3.13	9.4	5.33	10.33	-214.15	-1283.51	0.0448	21.08	18.17	10.68	8.48	5.04	9.33	0.4209	0.2504	0.4768	4.48	2.34
6	89	14	2.6	4	8.67	9.67	-215.69	-1289.90	0.0650	22.12	17.61	10.74	3.82	7.92	8.81	0.1912	0.3941	0.4500	6.50	4.04
7	103	5	3.13	10	6.33	10	-213.21	-1279.60	0.0361	20.92	18.00	10.72	8.95	5.93	9.10	0.4435	0.2947	0.4650	3.61	2.21
8	118	12	4.47	2.8	9.33	7.33	-215.62	-1289.63	0.0637	22.33	17.50	10.97	2.70	8.47	6.84	0.1350	0.4223	0.3509	6.37	3.22
9	134	12	5	7	5.33	8.33	-215.42	-1288.83	0.0702	21.56	18.17	10.87	6.49	5.04	7.71	0.3229	0.2504	0.3964	7.02	2.96
10	150	14	1.8	5.2	8	7.67	-215.19	-1287.86	0.0608	21.89	17.72	10.94	4.90	7.36	7.14	0.2447	0.3649	0.3658	6.08	3.91
11	166	3	3.93	3.4	5	7	-216.99	-1295.06	0.0231	22.21	18.24	10.99	3.26	4.75	6.55	0.1628	0.2357	0.3357	2.31	1.62
12	181	13	4.47	4	8.33	9.67	-215.84	-1290.55	0.0727	22.12	17.67	10.74	3.82	7.65	8.81	0.1912	0.3800	0.4500	7.27	3.64
13	197	13	4.2	6.4	5.67	11.67	-216.00	-1291.19	0.0671	21.69	18.13	10.55	5.97	5.35	10.33	0.2970	0.2658	0.5228	6.71	3.82
14	212	6	4.2	8.2	7	11.67	-214.19	-1283.67	0.0451	21.31	17.88	10.55	7.50	6.51	10.33	0.3724	0.3231	0.5228	4.51	2.04
15	227	2	3.4	8.8	9	9.67	-212.92	-1278.35	0.0209	21.18	17.55	10.74	7.99	8.20	8.81	0.3961	0.4082	0.4500	2.09	1.19
16	243	2	3.93	8.2	5.67	11	-214.90	-1286.45	0.0199	21.31	18.13	10.62	7.50	5.35	9.87	0.3724	0.2658	0.5045	1.99	1.60
17	260	10	4.2	1	6	8.33	-217.99	-1299.35	0.0628	22.69	18.05	10.87	0.98	5.63	7.71	0.0491	0.2796	0.3964	6.28	2.89
18	278	10	2.33	1.6	7.67	9.67	-217.22	-1296.19	0.0694	22.57	17.78	10.74	1.56	7.08	8.81	0.0783	0.3508	0.4500	6.94	2.93
19	293	7	2.6	10	9.67	9.67	-211.78	-1273.79	0.0449	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	4.49	2.57
20	310	8	5	8.2	5.67	9	-214.70	-1285.75	0.0438	21.31	18.13	10.81	7.50	5.35	8.25	0.3724	0.2658	0.4199	4.38	2.70
21	323	3	3.4	1	8.67	9.33	-217.15	-1295.70	0.0238	22.69	17.61	10.78	0.98	7.92	8.54	0.0491	0.3941	0.4351	2.38	1.53
22	341	6	5	7	5.33	7.33	-215.28	-1288.21	0.0585	21.56	18.17	10.97	6.49	5.04	6.84	0.3229	0.2504	0.3509	5.85	3.60
23	357	10	5	4	8	8	-215.83	-1290.62	0.0887	22.12	17.72	10.91	3.82	7.36	7.43	0.1912	0.3649	0.3811	8.87	4.54
24	370	4	4.47	6.4	7	8.33	-215.23	-1287.86	0.0303	21.69	17.88	10.87	5.97	6.51	7.71	0.2970	0.3231	0.3964	3.03	2.13
25	384	13	3.4	5.2	6.67	7.33	-215.66	-1289.77	0.0647	21.89	17.95	10.97	4.90	6.23	6.84	0.2447	0.3103	0.3509	6.47	3.39
26	399	6	3.93	4	7	8.67	-216.29	-1292.28	0.0443	22.12	17.88	10.84	3.82	6.51	7.99	0.1912	0.3231	0.4101	4.43	2.42
27	414	8	3.67	7.6	9	10	-213.88	-1282.41	0.0472	21.44	17.55	10.72	7.01	8.20	9.10	0.3485	0.4082	0.4650	4.72	2.58
28	431	13	2.6	7	7	7	-214.48	-1284.97	0.0642	21.56	17.88	10.99	6.49	6.51	6.55	0.3229	0.3231	0.3357	6.42	3.33
29	445	5	5	7.6	8.33	9	-214.01	-1282.92	0.0407	21.44	17.67	10.81	7.01	7.65	8.25	0.3485	0.3800	0.4199	4.07	2.86
30	461	11	4.73	5.2	9.33	7	-214.45	-1284.87	0.0743	21.89	17.50	10.99	4.90	8.47	6.55	0.2447	0.4223	0.3357	7.43	3.62
31	475	3	2.87	8.2	9	11	-213.50	-1280.71	0.0199	21.31	17.55	10.62	7.50	8.20	9.87	0.3724	0.4082	0.5045	1.99	1.30
32	490	2	3.93	8.2	6	9.67	-214.51	-1284.89	0.0239	21.31	18.05	10.74	7.50	5.63	8.81	0.3724	0.2796	0.4500	2.39	1.70
33	505	3	2.87	5.8	9.33	9.67	-214.84	-1286.22	0.0230	21.82	17.50	10.74	5.44	8.47	8.81	0.2705	0.4223	0.4500	2.30	1.47
34	525	10	3.13	9.4	5.33	10.33	-214.13	-1283.51	0.0612	21.08	18.17	10.68	8.48	5.04	9.33	0.4209	0.2504	0.4768	6.12	3.00
35	542	8	5	4	8	8	-215.87	-1290.62	0.0563	22.12	17.72	10.91	3.82	7.36	7.43	0.1912	0.3649	0.3811	5.63	2.86
36	558	5	5	6.4	7.67	9	-215.10	-1287.35	0.0352	21.69	17.78	10.81	5.97	7.08	8.25	0.2970	0.3508	0.4199	3.52	2.40
37	576	13	3.4	4	8.67	9.67	-215.68	-1289.90	0.0694	22.12	17.61	10.74	3.82	7.92	8.81	0.1912	0.3941	0.4500	6.94	3.51
38	589	7	2.6	5.2	6.67	7.33	-215.67	-1289.77	0.0501	21.89	17.95	10.97	4.90	6.23	6.84	0.2447	0.3103	0.3509	5.01	2.94
39	607	5	3.13	10	7	9.67	-212.80	-1277.90	0.0332	20.92	17.88	10.74	8.95	6.51	8.81	0.4435	0.3231	0.4500	3.32	1.81
40	620	7	2.6	10	9	10	-212.14	-1275.29	0.0517	20.92	17.55	10.72	8.95	8.20	9.10	0.4435	0.4082	0.4650	5.17	2.79
41	634	8	3.93	8.2	5.67	11	-214.87	-1286.45	0.0473	21.31	18.13	10.62	7.50	5.35	9.87	0.3724	0.2658	0.5045	4.73	2.41
42	652	2	2.6	10	9.67	9.67	-211.81	-1273.79	0.0195	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	1.95	1.36
43	666	10	4.73	4	8.67	9.67	-215.69	-1289.90	0.0658	22.12	17.61	10.74	3.82	7.92	8.81	0.1912	0.3941	0.4500	6.58	3.22
44	679	4	4.47	6.4	7	9	-215.31	-1288.21	0.0367	21.69	17.88	10.81	5.97	6.51	8.25	0.2970	0.3231	0.4199	3.67	2.10
45	695	3	2.6	7.6	9	10	-213.91	-1282.41	0.0233	21.44	17.55	10.72	7.01	8.20	9.10	0.3485	0.4082	0.4650	2.33	1.60
46	712	8	3.93	8.2	9	11	-213.45	-1280.71	0.0606	21.31	17.55	10.62	7.50	8.20	9.87	0.3724	0.4082	0.5045	6.06	3.96
47	729	3	2.87	8.2	9.67	9.67	-213.12	-1279.18	0.0228	21.31	17.45	10.74	7.50	8.75	8.81	0.3724	0.4338	0.4500	2.28	1.26
48	743	7	2.6	10	9.33	9.67	-211.87	-1274.18	0.0457	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	4.57	2.78
49	757	13	2.6	5.8	7	7	-215.33	-1288.46	0.0683	21.82	17.88	10.99	5.44	6.51	6.55	0.2705	0.3231	0.3357	6.83	3.62
50	773	5	3.67	10	7	9.67	-212.79	-1277.90	0.0346	20.92	17.88	10.74	8.95	6.51	8.81	0.4435	0.3231	0.4500	3.46	1.90

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
51	787	8	4.47	6.4	7	9	-215.30	-1288.21	0.0466	21.69	17.88	10.81	5.97	6.51	8.25	0.2970	0.3231	0.4199	4.66	2.33
52	802	3	2.07	10	9.33	9.67	-211.90	-1274.18	0.0257	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	2.57	1.87
53	820	8	2.6	7	7	7	-214.50	-1284.97	0.0511	21.56	17.88	10.99	6.49	6.51	6.55	0.3229	0.3231	0.3357	5.11	2.38
54	837	12	2.6	10	9	10	-212.12	-1275.29	0.0652	20.92	17.55	10.72	8.95	8.20	9.10	0.4435	0.4082	0.4650	6.52	3.09
55	852	7	2.6	7.6	9	10	-213.89	-1282.41	0.0417	21.44	17.55	10.72	7.01	8.20	9.10	0.3485	0.4082	0.4650	4.17	1.94
56	868	7	2.87	8.2	9.67	9.67	-213.10	-1279.18	0.0421	21.31	17.45	10.74	7.50	8.75	8.81	0.3724	0.4338	0.4500	4.21	2.16
57	886	3	2.6	7.6	9	10.67	-213.85	-1282.18	0.0239	21.44	17.55	10.65	7.01	8.20	9.60	0.3485	0.4082	0.4905	2.39	1.44
58	901	3	3.93	8.8	9	9.67	-212.91	-1278.35	0.0272	21.18	17.55	10.74	7.99	8.20	8.81	0.3961	0.4082	0.4500	2.72	1.80
59	915	2	3.4	8.8	9	10	-213.07	-1278.96	0.0214	21.18	17.55	10.72	7.99	8.20	9.10	0.3961	0.4082	0.4650	2.14	1.31
60	931	3	2.6	7.6	9	11	-213.92	-1282.43	0.0181	21.44	17.55	10.62	7.01	8.20	9.87	0.3485	0.4082	0.5045	1.81	1.26
61	945	3	2.87	8.2	9	10	-213.49	-1280.68	0.0194	21.31	17.55	10.72	7.50	8.20	9.10	0.3724	0.4082	0.4650	1.94	1.42
62	961	3	2.6	10	9	10	-212.18	-1275.29	0.0194	20.92	17.55	10.72	8.95	8.20	9.10	0.4435	0.4082	0.4650	1.94	1.45
63	974	5	2.6	10	9.67	9.67	-211.80	-1273.79	0.0269	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	2.69	2.16
64	988	7	2.6	10	9.67	7	-211.41	-1272.31	0.0483	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	4.83	2.61
65	1003	15	3.67	6.4	9	10	-214.65	-1285.68	0.0699	21.69	17.55	10.72	5.97	8.20	9.10	0.2970	0.4082	0.4650	6.99	4.86
66	1021	3	2.6	8.2	9	11	-213.50	-1280.71	0.0191	21.31	17.55	10.62	7.50	8.20	9.87	0.3724	0.4082	0.5045	1.91	1.25
67	1035	7	2.6	6.4	9	10	-214.68	-1285.68	0.0434	21.69	17.55	10.72	5.97	8.20	9.10	0.2970	0.4082	0.4650	4.34	2.14
68	1050	15	3.67	10	9.67	7	-211.39	-1272.31	0.0689	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.89	5.14
69	1065	4	3.67	7.6	9	10	-213.90	-1282.41	0.0324	21.44	17.55	10.72	7.01	8.20	9.10	0.3485	0.4082	0.4650	3.24	1.77
70	1084	7	2.6	10	9	9.67	-211.99	-1274.68	0.0465	20.92	17.55	10.74	8.95	8.20	8.81	0.4435	0.4082	0.4500	4.65	2.36
71	1099	7	2.6	10	9.67	10	-211.93	-1274.40	0.0433	20.92	17.45	10.72	8.95	8.75	9.10	0.4435	0.4338	0.4650	4.33	2.30
72	1117	3	2.87	8.8	9	9.67	-212.92	-1278.35	0.0229	21.18	17.55	10.74	7.99	8.20	8.81	0.3961	0.4082	0.4500	2.29	1.58
73	1132	2	3.4	8.2	9.33	9.67	-213.22	-1279.57	0.0211	21.31	17.50	10.74	7.50	8.47	8.81	0.3724	0.4223	0.4500	2.11	1.42
74	1150	3	2.07	10	9.67	9.67	-211.80	-1273.79	0.0239	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	2.39	1.60
75	1165	7	2.6	8.8	9.33	9.67	-212.76	-1277.85	0.0496	21.18	17.50	10.74	7.99	8.47	8.81	0.3961	0.4223	0.4500	4.96	2.27
76	1179	3	2.6	7.6	7.67	10.67	-214.43	-1284.54	0.0205	21.44	17.78	10.65	7.01	7.08	9.60	0.3485	0.3508	0.4905	2.05	1.30
77	1194	2	2.6	10	9.33	9.67	-211.90	-1274.18	0.0182	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	1.82	1.37
78	1210	7	2.6	10	10	9.67	-211.65	-1273.23	0.0435	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	4.35	2.30
79	1225	2	2.6	10	9.33	11	-212.06	-1274.81	0.0211	20.92	17.50	10.62	8.95	8.47	9.87	0.4435	0.4223	0.5045	2.11	1.47
80	1239	2	3.4	8.2	9	11	-213.50	-1280.71	0.0209	21.31	17.55	10.62	7.50	8.20	9.87	0.3724	0.4082	0.5045	2.09	1.27
81	1254	3	2.6	9.4	9.33	9.67	-212.53	-1276.77	0.0230	21.08	17.50	10.74	8.48	8.47	8.81	0.4209	0.4223	0.4500	2.30	1.52
82	1271	7	2.6	10	8	9.67	-212.42	-1276.40	0.0416	20.92	17.72	10.74	8.95	7.36	8.81	0.4435	0.3649	0.4500	4.16	2.55
83	1287	7	1.53	10	9.33	9.67	-211.88	-1274.18	0.0398	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	3.98	2.31
84	1301	2	4.73	10	9.33	9.67	-211.89	-1274.18	0.0283	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	2.83	1.71
85	1318	2	2.07	10	10	7	-211.32	-1271.75	0.0133	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.33	1.13
86	1334	4	4.73	7.6	9.33	9.67	-213.61	-1281.29	0.0417	21.44	17.50	10.74	7.01	8.47	8.81	0.3485	0.4223	0.4500	4.17	2.10
87	1351	7	2.6	8.8	9	10	-213.04	-1278.96	0.0492	21.18	17.55	10.72	7.99	8.20	9.10	0.3961	0.4082	0.4650	4.92	2.25
88	1368	2	2.6	10	9.67	7	-211.45	-1272.31	0.0153	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	1.53	1.22
89	1384	12	2.07	10	10	7	-211.26	-1271.75	0.0616	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.16	3.00
90	1401	2	2.6	10	9	10	-212.18	-1275.29	0.0158	20.92	17.55	10.72	8.95	8.20	9.10	0.4435	0.4082	0.4650	1.58	1.34
91	1417	5	4.73	10	9.33	11	-212.04	-1274.81	0.0364	20.92	17.50	10.62	8.95	8.47	9.87	0.4435	0.4223	0.5045	3.64	2.17
92	1431	2	4.73	10	8	9.67	-212.44	-1276.40	0.0258	20.92	17.72	10.74	8.95	7.36	8.81	0.4435	0.3649	0.4500	2.58	1.45
93	1449	7	1.53	10	9.33	11	-212.04	-1274.81	0.0384	20.92	17.50	10.62	8.95	8.47	9.87	0.4435	0.4223	0.5045	3.84	1.98
94	1465	2	2.6	9.4	9.67	9.67	-212.45	-1276.38	0.0127	21.08	17.45	10.74	8.48	8.75	8.81	0.4209	0.4338	0.4500	1.27	0.95
95	1479	2	2.6	10	10	9.67	-211.67	-1273.23	0.0206	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.06	1.29
96	1497	7	2.6	10	9.33	11	-212.03	-1274.81	0.0429	20.92	17.50	10.62	8.95	8.47	9.87	0.4435	0.4223	0.5045	4.29	2.07
97	1513	7	1.53	10	9.67	9.67	-211.78	-1273.79	0.0393	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	3.93	2.06
98	1527	2	2.6	9.4	9.33	11	-212.69	-1277.41	0.0185	21.08	17.50	10.62	8.48	8.47	9.87	0.4209	0.4223	0.5045	1.85	1.30
99	1541	7	1.53	10	10	9.67	-211.65	-1273.23	0.0386	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	3.86	1.95
100	1559	3	1.53	10	7.67	9.67	-212.61	-1277.04	0.0164	20.92	17.78	10.74	8.95	7.08	8.81	0.4435	0.3508	0.4500	1.64	1.13

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
101	1573	2	2.07	10	9.33	9.67	-211.91	-1274.18	0.0157	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	1.57	1.28
102	1591	7	2.6	10	10	7	-211.28	-1271.75	0.0475	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.75	2.24
103	1607	2	2.6	10	8	9.67	-212.45	-1276.40	0.0168	20.92	17.72	10.74	8.95	7.36	8.81	0.4435	0.3649	0.4500	1.68	1.29
104	1623	12	2.07	10	9.33	10	-212.00	-1274.79	0.0663	20.92	17.50	10.72	8.95	8.47	9.10	0.4435	0.4223	0.4650	6.63	3.11
105	1641	4	2.07	10	10	7	-211.31	-1271.75	0.0243	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.43	1.54
106	1659	2	2.6	10	9	9.67	-212.03	-1274.68	0.0168	20.92	17.55	10.74	8.95	8.20	8.81	0.4435	0.4082	0.4500	1.68	1.37
107	1675	7	1.53	10	9.33	7	-211.52	-1272.70	0.0389	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	3.89	2.11
108	1692	4	2.07	10	10	11	-211.82	-1273.87	0.0243	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	2.43	1.53
109	1708	4	2.07	10	9.33	9.67	-211.90	-1274.18	0.0203	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	2.03	1.37
110	1724	3	2.6	10	9.33	9.67	-211.89	-1274.18	0.0294	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	2.94	2.04
111	1738	7	2.07	10	9.33	9.67	-211.88	-1274.18	0.0427	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	4.27	2.57
112	1755	7	1.53	10	10	7	-211.29	-1271.75	0.0406	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.06	1.98
113	1769	7	2.6	10	9.33	7	-211.51	-1272.70	0.0447	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	4.47	2.63
114	1787	7	2.6	10	10	10	-211.80	-1273.84	0.0371	20.92	17.40	10.72	8.95	9.02	9.10	0.4435	0.4478	0.4650	3.71	1.88
115	1801	7	1	10	9.33	11	-212.04	-1274.81	0.0362	20.92	17.50	10.62	8.95	8.47	9.87	0.4435	0.4223	0.5045	3.62	1.92
116	1816	4	2.07	10	9.67	9.67	-211.81	-1273.79	0.0218	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	2.18	1.36
117	1834	3	2.6	10	9.67	9.67	-211.80	-1273.79	0.0271	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	2.71	1.67
118	1849	2	2.6	10	10	7	-211.32	-1271.75	0.0155	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.55	1.22
119	1868	7	2.6	10	9.33	10.33	-211.92	-1274.37	0.0435	20.92	17.50	10.68	8.95	8.47	9.33	0.4435	0.4223	0.4768	4.35	2.31
120	1884	8	2.07	10	10	11	-211.80	-1273.87	0.0458	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	4.58	2.29
121	1899	3	1.53	10	10	9.67	-211.68	-1273.23	0.0178	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	1.78	1.40
122	1915	12	2.07	10	9.67	10.33	-211.80	-1273.98	0.0657	20.92	17.45	10.68	8.95	8.75	9.33	0.4435	0.4338	0.4768	6.57	3.14
123	1932	12	2.07	10	10	11	-211.77	-1273.87	0.0649	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	6.49	3.29
124	1948	8	2.07	10	10	7	-211.28	-1271.75	0.0456	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.56	2.30
125	1965	3	1.53	10	10	7	-211.31	-1271.75	0.0173	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.73	1.32
126	1981	4	2.33	10	9.67	11	-211.96	-1274.42	0.0238	20.92	17.45	10.62	8.95	8.75	9.87	0.4435	0.4338	0.5045	2.38	1.70
127	1997	5	1.53	10	10	9.67	-211.67	-1273.23	0.0227	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.27	1.73
128	2015	2	1.53	10	10	9.67	-211.68	-1273.23	0.0144	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	1.44	1.07
129	2029	2	1.53	10	10	7	-211.32	-1271.75	0.0111	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.11	0.99
130	2045	12	2.07	10	10	8.33	-211.52	-1272.80	0.0622	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.22	2.97
131	2061	7	2.6	10	6.67	9.67	-212.97	-1278.65	0.0387	20.92	17.95	10.74	8.95	6.23	8.81	0.4435	0.3103	0.4500	3.87	2.23
132	2075	4	2.6	10	10	7	-211.30	-1271.75	0.0264	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.64	1.61
133	2089	12	2.07	5.2	10	7	-214.23	-1283.92	0.0613	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	6.13	3.13
134	2105	3	1	8.8	10	7	-212.21	-1275.43	0.0143	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	1.43	1.10
135	2119	3	1.53	10	10	7.67	-211.50	-1272.53	0.0181	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	1.81	1.31
136	2137	14	2.6	10	10	9.67	-211.62	-1273.23	0.0626	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.26	4.14
137	2153	6	2.07	10	10	7	-211.30	-1271.75	0.0313	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.13	2.09
138	2171	3	2.07	10	10	7	-211.31	-1271.75	0.0195	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.95	1.45
139	2187	13	2.07	10	10	11	-211.78	-1273.87	0.0625	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	6.25	3.46
140	2209	4	2.6	5.2	10	7	-214.27	-1283.92	0.0312	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	3.12	1.58
141	2223	3	1.53	10	8.67	7	-211.80	-1273.75	0.0168	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	1.68	1.23
142	2241	7	1.53	10	10	11	-211.80	-1273.87	0.0400	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	4.00	1.97
143	2258	4	2.33	10	9.67	9.67	-211.80	-1273.79	0.0226	20.92	17.45	10.74	8.95	8.75	8.81	0.4435	0.4338	0.4500	2.26	1.41
144	2276	4	2.07	5.2	10	7	-214.27	-1283.92	0.0289	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	2.89	1.48
145	2293	4	2.07	10	10	9.67	-211.67	-1273.23	0.0200	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.00	1.35
146	2309	2	2.6	10	10	11	-211.83	-1273.87	0.0183	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	1.83	1.37
147	2325	7	2.6	9.4	10	7	-211.91	-1274.35	0.0480	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	4.80	2.42
148	2340	7	2.6	10	8.67	7	-211.77	-1273.75	0.0488	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	4.88	2.52
149	2354	5	1.53	10	10	7	-211.31	-1271.75	0.0244	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.44	2.19
150	2368	12	2.07	10	10	9.67	-211.62	-1273.23	0.0652	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.52	3.14

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
151	2384	8	2.07	10	10	9.67	-211.64	-1273.23	0.0459	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	4.59	2.24
152	2402	7	1.53	10	8.67	7	-211.77	-1273.75	0.0411	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	4.11	2.13
153	2418	3	2.6	9.4	10	7	-211.94	-1274.35	0.0229	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	2.29	1.61
154	2434	3	2.07	10	9.33	7	-211.54	-1272.70	0.0202	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	2.02	1.54
155	2448	7	1.53	10	6.67	7	-212.62	-1277.17	0.0347	20.92	17.95	10.99	8.95	6.23	6.55	0.4435	0.3103	0.3357	3.47	2.07
156	2464	3	1.53	10	6	7	-212.85	-1278.02	0.0157	20.92	18.05	10.99	8.95	5.63	6.55	0.4435	0.2796	0.3357	1.57	1.16
157	2482	7	2.6	7	10	7	-213.37	-1280.30	0.0470	21.56	17.40	10.99	6.49	9.02	6.55	0.3229	0.4478	0.3357	4.70	2.38
158	2498	7	1.53	10	10	7.33	-211.39	-1272.18	0.0406	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	4.06	1.98
159	2514	3	1.53	7.6	10	7	-213.05	-1278.87	0.0151	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	1.51	1.28
160	2532	12	2.07	10	8.67	7	-211.74	-1273.75	0.0709	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	7.09	3.24
161	2548	13	2.6	10	10	7	-211.26	-1271.75	0.0662	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.62	3.74
162	2566	7	2.07	10	10	11	-211.80	-1273.87	0.0427	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	4.27	2.19
163	2584	7	1.53	10	10	7.67	-211.48	-1272.53	0.0414	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	4.14	2.11
164	2599	13	2.07	5.2	10	9.67	-214.60	-1285.40	0.0584	21.89	17.40	10.74	4.90	9.02	8.81	0.2447	0.4478	0.4500	5.84	3.33
165	2615	13	2.07	10	10	9.67	-211.63	-1273.23	0.0597	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	5.97	3.44
166	2629	6	2.07	10	9.33	7	-211.53	-1272.70	0.0324	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	3.24	2.16
167	2644	3	1.53	10	10	11	-211.83	-1273.87	0.0158	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	1.58	1.11
168	2658	13	1	10	10	7.67	-211.46	-1272.53	0.0586	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.86	3.31
169	2674	7	2.87	10	10	9.67	-211.64	-1273.23	0.0447	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	4.47	2.38
170	2690	14	2.6	10	10	7	-211.27	-1271.75	0.0575	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.75	4.07
171	2707	5	2.07	10	10	7	-211.30	-1271.75	0.0276	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.76	2.35
172	2723	13	2.07	10	10	7	-211.26	-1271.75	0.0643	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.43	3.65
173	2738	5	2.07	5.2	10	7	-214.27	-1283.92	0.0312	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	3.12	2.12
174	2754	7	1	10	9.67	7	-211.43	-1272.31	0.0373	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	3.73	1.96
175	2775	10	2.6	10	10	7	-211.27	-1271.75	0.0544	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.44	3.69
176	2793	2	2.6	5.2	10	7	-214.29	-1283.92	0.0166	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	1.66	1.28
177	2808	7	2.07	10	10	7	-211.28	-1271.75	0.0441	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.41	2.08
178	2823	2	1.53	5.2	10	7	-214.29	-1283.92	0.0121	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	1.21	1.07
179	2841	7	1.53	10	8	7	-212.07	-1274.92	0.0351	20.92	17.72	10.99	8.95	7.36	6.55	0.4435	0.3649	0.3357	3.51	2.00
180	2857	13	2.07	10	10	7.67	-211.45	-1272.53	0.0601	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	6.01	3.42
181	2875	13	1	10	10	7	-211.26	-1271.75	0.0605	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.05	3.43
182	2891	10	1.53	10	10	10.33	-211.68	-1273.43	0.0515	20.92	17.40	10.68	8.95	9.02	9.33	0.4435	0.4478	0.4768	5.15	2.85
183	2905	5	2.07	10	8.67	7	-211.79	-1273.75	0.0270	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	2.70	2.34
184	2920	7	3.13	10	10	7.33	-211.38	-1272.18	0.0509	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	5.09	2.46
185	2936	13	1.53	10	9.67	7	-211.40	-1272.31	0.0635	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.35	3.44
186	2954	12	2.07	5.2	8.67	7	-214.72	-1285.92	0.0626	21.89	17.61	10.99	4.90	7.92	6.55	0.2447	0.3941	0.3357	6.26	3.10
187	2968	8	2.6	10	10	7	-211.28	-1271.75	0.0475	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.75	2.37
188	2984	12	1.53	10	10	7.33	-211.37	-1272.18	0.0613	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.13	2.94
189	3000	5	2.07	10	8	7	-212.07	-1274.92	0.0322	20.92	17.72	10.99	8.95	7.36	6.55	0.4435	0.3649	0.3357	3.22	2.44
190	3015	13	1	10	10	9.67	-211.63	-1273.23	0.0578	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	5.78	3.35
191	3031	5	2.07	10	8.33	7	-211.95	-1274.40	0.0303	20.92	17.67	10.99	8.95	7.65	6.55	0.4435	0.3800	0.3357	3.03	2.23
192	3047	7	1	10	10	7	-211.29	-1271.75	0.0372	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.72	1.93
193	3065	3	4.2	10	10	7.67	-211.49	-1272.53	0.0305	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	3.05	1.98
194	3084	7	1	10	9.67	7.67	-211.62	-1273.08	0.0373	20.92	17.45	10.94	8.95	8.75	7.14	0.4435	0.4338	0.3658	3.73	1.92
195	3099	6	1	10	10	7.67	-211.49	-1272.53	0.0277	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	2.77	1.88
196	3117	14	2.07	10	10	7	-211.27	-1271.75	0.0563	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.63	3.99
197	3137	7	2.6	10	10	7.33	-211.38	-1272.18	0.0475	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	4.75	2.24
198	3154	7	3.13	10	10	7	-211.28	-1271.75	0.0509	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.09	2.45
199	3169	12	1	10	10	7	-211.26	-1271.75	0.0597	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.97	2.94
200	3183	13	2.07	10	8.67	7	-211.76	-1273.75	0.0570	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.70	3.18

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
201	3198	13	1.53	10	10	7	-211.26	-1271.75	0.0624	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.24	3.55
202	3213	14	2.07	10	10	7.67	-211.46	-1272.53	0.0566	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.66	3.99
203	3229	14	2.6	10	10	8.33	-211.52	-1272.80	0.0587	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.87	3.99
204	3245	14	2.6	10	10	7.67	-211.46	-1272.53	0.0578	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.78	4.07
205	3261	3	4.2	10	10	8.33	-211.56	-1272.80	0.0286	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	2.86	1.82
206	3278	6	1	10	8.67	7	-211.79	-1273.75	0.0317	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	3.17	2.08
207	3294	15	1.53	10	10	7	-211.26	-1271.75	0.0598	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.98	4.81
208	3312	12	2.6	10	10	8.33	-211.52	-1272.80	0.0634	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.34	3.00
209	3329	7	2.07	10	10	8.33	-211.54	-1272.80	0.0431	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	4.31	2.18
210	3347	14	2.07	10	9.67	7.67	-211.59	-1273.08	0.0558	20.92	17.45	10.94	8.95	8.75	7.14	0.4435	0.4338	0.3658	5.58	3.91
211	3364	3	3.13	10	10	7	-211.31	-1271.75	0.0239	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.39	1.69
212	3382	13	2.07	10	10	7.33	-211.37	-1272.18	0.0603	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.03	3.41
213	3401	13	2.07	10	10	9	-211.61	-1273.16	0.0578	20.92	17.40	10.81	8.95	9.02	8.25	0.4435	0.4478	0.4199	5.78	3.31
214	3415	12	2.6	5.2	10	7.33	-214.34	-1284.35	0.0620	21.89	17.40	10.97	4.90	9.02	6.84	0.2447	0.4478	0.3509	6.20	3.17
215	3430	13	1.53	7.6	10	7	-213.00	-1278.87	0.0620	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	6.20	3.56
216	3446	14	2.6	10	10	7.33	-211.37	-1272.18	0.0575	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	5.75	4.08
217	3461	3	1	10	10	7	-211.32	-1271.75	0.0151	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	1.51	1.20
218	3477	8	1	10	10	7	-211.29	-1271.75	0.0417	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.17	2.18
219	3491	7	2.6	10	10	9	-211.62	-1273.16	0.0479	20.92	17.40	10.81	8.95	9.02	8.25	0.4435	0.4478	0.4199	4.79	2.49
220	3510	13	1.53	10	10	7.33	-211.37	-1272.18	0.0595	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	5.95	3.36
221	3527	15	1	10	10	7.67	-211.46	-1272.53	0.0577	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.77	4.73
222	3542	15	1	10	10	7	-211.27	-1271.75	0.0574	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.74	4.73
223	3558	3	1	10	10	9.67	-211.68	-1273.23	0.0155	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	1.55	1.19
224	3574	13	2.6	8.8	10	7	-212.16	-1275.43	0.0609	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.09	3.36
225	3592	7	3.67	10	10	7	-211.27	-1271.75	0.0544	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.44	2.61
226	3607	3	2.07	10	7.33	7	-212.33	-1275.92	0.0185	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	1.85	1.27
227	3623	14	2.6	10	8.67	7.33	-211.86	-1274.18	0.0550	20.92	17.61	10.97	8.95	7.92	6.84	0.4435	0.3941	0.3509	5.50	3.86
228	3641	3	3.13	10	10	9	-211.64	-1273.16	0.0289	20.92	17.40	10.81	8.95	9.02	8.25	0.4435	0.4478	0.4199	2.89	1.90
229	3655	13	2.07	10	9.33	7	-211.49	-1272.70	0.0626	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.26	3.36
230	3670	13	2.6	10	10	8.33	-211.52	-1272.80	0.0581	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.81	3.29
231	3690	14	1	10	10	7	-211.27	-1271.75	0.0539	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.39	3.80
232	3709	7	2.07	10	9.33	7	-211.52	-1272.70	0.0418	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	4.18	2.37
233	3725	14	3.13	10	10	7	-211.27	-1271.75	0.0587	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.87	4.15
234	3739	14	2.07	7.6	10	7	-213.00	-1278.87	0.0616	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	6.16	3.89
235	3754	7	3.13	10	10	7.67	-211.46	-1272.53	0.0525	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.25	2.61
236	3770	7	3.13	10	8.67	7	-211.76	-1273.75	0.0526	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.26	2.72
237	3787	13	1	10	10	8.33	-211.52	-1272.80	0.0573	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.73	3.26
238	3804	3	3.13	10	10	9.67	-211.67	-1273.23	0.0246	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.46	1.74
239	3820	3	3.67	10	10	7	-211.30	-1271.75	0.0261	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.61	1.81
240	3837	15	2.07	10	10	7	-211.26	-1271.75	0.0622	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.22	4.88
241	3853	7	4.2	10	10	7	-211.27	-1271.75	0.0578	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.78	2.71
242	3872	13	1	10	7.33	7	-212.29	-1275.92	0.0551	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	5.51	3.07
243	3889	14	1.53	10	10	7	-211.27	-1271.75	0.0551	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.51	3.90
244	3906	14	4.2	10	10	7	-211.26	-1271.75	0.0611	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.11	4.30
245	3923	15	1.27	10	10	7	-211.27	-1271.75	0.0586	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.86	4.77
246	3939	13	2.6	9.4	10	7	-211.90	-1274.35	0.0604	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.04	3.33
247	3954	12	2.6	9.4	10	7	-211.89	-1274.35	0.0634	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.34	3.04
248	3968	3	3.13	10	8.67	7	-211.80	-1273.75	0.0235	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	2.35	1.42
249	3985	5	3.13	10	9.67	7	-211.43	-1272.31	0.0321	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	3.21	2.03
250	4002	3	2.6	10	10	7	-211.31	-1271.75	0.0217	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.17	1.59

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
251	4017	12	3.13	10	10	7	-211.26	-1271.75	0.0635	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.35	3.07
252	4033	15	1	10	10	9.67	-211.62	-1273.23	0.0621	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.21	4.82
253	4050	7	3.93	10	10	7	-211.27	-1271.75	0.0561	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.61	2.67
254	4069	11	2.6	10	10	7	-211.27	-1271.75	0.0560	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.60	3.13
255	4086	12	1.53	10	10	7	-211.26	-1271.75	0.0606	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.06	2.97
256	4104	7	4.2	10	10	9.67	-211.64	-1273.23	0.0500	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	5.00	2.63
257	4120	14	1	7.6	10	7	-213.00	-1278.87	0.0581	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	5.81	3.78
258	4136	15	1.27	10	8.67	7	-211.76	-1273.75	0.0576	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.76	4.79
259	4155	14	4.2	10	10	8.33	-211.52	-1272.80	0.0625	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.25	4.13
260	4172	15	1.8	10	10	7	-211.26	-1271.75	0.0610	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.10	4.84
261	4190	15	1.27	7.6	10	7	-213.00	-1278.87	0.0625	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	6.25	4.82
262	4207	7	4.47	10	10	7	-211.27	-1271.75	0.0596	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.96	2.75
263	4221	15	1.27	5.2	10	7	-214.23	-1283.92	0.0629	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	6.29	4.85
264	4238	15	1.27	10	10	7.67	-211.46	-1272.53	0.0587	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.87	4.77
265	4258	14	1.27	10	10	7.67	-211.46	-1272.53	0.0548	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.48	3.85
266	4273	15	4.73	10	10	8.33	-211.51	-1272.80	0.0673	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.73	4.97
267	4288	7	3.93	10	7.33	7	-212.30	-1275.92	0.0422	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	4.22	2.40
268	4306	7	4.47	10	8.67	7	-211.75	-1273.75	0.0601	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	6.01	3.06
269	4322	14	1.53	10	10	7.67	-211.46	-1272.53	0.0554	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.54	3.90
270	4339	10	3.93	10	10	7	-211.27	-1271.75	0.0596	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.96	4.17
271	4354	13	2.07	8.8	9.33	7	-212.39	-1276.37	0.0612	21.18	17.50	10.99	7.99	8.47	6.55	0.3961	0.4223	0.3357	6.12	3.34
272	4371	7	1.8	5.2	10	7	-214.26	-1283.92	0.0369	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	3.69	1.88
273	4387	8	1.53	10	10	7	-211.28	-1271.75	0.0436	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.36	2.24
274	4404	13	2.6	5.2	10	7	-214.21	-1283.92	0.0785	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	7.85	4.17
275	4420	7	1.53	10	9.33	7.33	-211.62	-1273.13	0.0387	20.92	17.50	10.97	8.95	8.47	6.84	0.4435	0.4223	0.3509	3.87	2.12
276	4442	14	1.53	8.8	10	7	-212.17	-1275.43	0.0546	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	5.46	3.76
277	4464	7	3.93	9.4	10	7	-211.90	-1274.35	0.0566	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.66	2.59
278	4480	12	1.53	10	7.33	7	-212.27	-1275.92	0.0704	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	7.04	3.19
279	4498	7	1.53	5.2	10	7	-214.26	-1283.92	0.0362	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	3.62	1.86
280	4514	7	1.53	8.8	10	11	-212.70	-1277.54	0.0399	21.18	17.40	10.62	7.99	9.02	9.87	0.3961	0.4478	0.5045	3.99	2.06
281	4529	12	2.07	10	9.67	7	-211.40	-1272.31	0.0622	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.22	3.01
282	4546	8	1.27	10	10	7	-211.28	-1271.75	0.0427	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.27	2.21
283	4564	14	3.93	10	10	7	-211.26	-1271.75	0.0605	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.05	4.27
284	4579	7	1.53	7.6	10	9.67	-213.39	-1280.35	0.0398	21.44	17.40	10.74	7.01	9.02	8.81	0.3485	0.4478	0.4500	3.98	1.89
285	4594	5	3.13	10	10	7	-211.30	-1271.75	0.0337	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.37	2.77
286	4612	14	3.67	10	10	7	-211.26	-1271.75	0.0599	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.99	4.23
287	4627	7	1	9.4	10	7	-211.92	-1274.35	0.0377	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	3.77	2.08
288	4648	7	4.73	10	10	7	-211.26	-1271.75	0.0612	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.12	2.78
289	4664	7	1.53	9.4	10	7	-211.92	-1274.35	0.0411	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	4.11	2.22
290	4681	5	2.07	10	10	8.33	-211.56	-1272.80	0.0227	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	2.27	1.77
291	4696	9	1.27	10	10	7	-211.28	-1271.75	0.0456	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.56	2.74
292	4714	12	2.6	10	10	9.67	-211.62	-1273.23	0.0673	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.73	3.21
293	4731	7	1	10	10	7.67	-211.48	-1272.53	0.0377	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	3.77	1.95
294	4755	14	3.93	10	9.67	7	-211.40	-1272.31	0.0599	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	5.99	4.22
295	4775	7	1.53	8.8	10	9.67	-212.54	-1276.91	0.0442	21.18	17.40	10.74	7.99	9.02	8.81	0.3961	0.4478	0.4500	4.42	1.99
296	4792	7	3.67	10	10	9.67	-211.64	-1273.23	0.0483	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	4.83	2.57
297	4809	8	2.07	9.4	10	7	-211.91	-1274.35	0.0538	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.38	2.48
298	4828	14	1	10	10	7.67	-211.46	-1272.53	0.0542	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	5.42	3.80
299	4846	6	1	10	10	7.33	-211.41	-1272.18	0.0275	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	2.75	1.89
300	4863	8	1.53	8.8	10	7	-212.17	-1275.43	0.0480	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	4.80	2.29

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
301	4878	5	2.07	10	10	9.67	-211.67	-1273.23	0.0244	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.44	1.76
302	4893	5	4.2	10	10	7	-211.29	-1271.75	0.0399	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.99	3.21
303	4911	15	1.27	9.4	10	7	-211.90	-1274.35	0.0573	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.73	4.70
304	4928	7	3.67	10	10	7.33	-211.38	-1272.18	0.0544	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	5.44	2.61
305	4943	6	1	10	10	7	-211.30	-1271.75	0.0278	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.78	1.96
306	4961	7	3.67	5.2	10	7	-214.26	-1283.92	0.0414	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	4.14	1.97
307	4978	15	3.13	10	10	7	-211.26	-1271.75	0.0669	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.69	5.00
308	4994	5	3.13	10	10	9.67	-211.66	-1273.23	0.0278	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	2.78	1.78
309	5015	5	2.07	10	10	7.33	-211.41	-1272.18	0.0287	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	2.87	2.25
310	5030	14	3.4	10	10	7	-211.27	-1271.75	0.0593	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.93	4.19
311	5049	7	4.73	10	10	8.33	-211.52	-1272.80	0.0588	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.88	3.20
312	5066	14	3.13	10	10	9.67	-211.62	-1273.23	0.0644	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.44	4.20
313	5084	14	3.4	10	8.67	7	-211.76	-1273.75	0.0567	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.67	3.94
314	5101	5	1	10	10	7	-211.31	-1271.75	0.0214	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.14	1.98
315	5118	5	3.13	5.2	7.33	7	-215.28	-1288.09	0.0348	21.89	17.83	10.99	4.90	6.78	6.55	0.2447	0.3363	0.3357	3.48	2.23
316	5135	5	3.13	10	9.33	7	-211.53	-1272.70	0.0278	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	2.78	2.28
317	5153	5	4.2	10	10	9.67	-211.66	-1273.23	0.0311	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	3.11	1.95
318	5170	8	3.13	10	10	7	-211.28	-1271.75	0.0494	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.94	2.42
319	5186	7	3.13	10	10	9.67	-211.64	-1273.23	0.0459	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	4.59	2.45
320	5204	14	3.4	10	9.33	7	-211.50	-1272.70	0.0576	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	5.76	3.89
321	5223	5	3.13	8.8	10	7	-212.20	-1275.43	0.0303	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	3.03	2.53
322	5239	7	3.13	10	7.33	7	-212.31	-1275.92	0.0398	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	3.98	2.25
323	5256	7	4.73	10	7.33	7	-212.30	-1275.92	0.0446	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	4.46	2.53
324	5273	11	3.13	10	10	7	-211.27	-1271.75	0.0573	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.73	3.18
325	5290	14	3.13	10	9.33	9.67	-211.86	-1274.18	0.0607	20.92	17.50	10.74	8.95	8.47	8.81	0.4435	0.4223	0.4500	6.06	4.09
326	5305	14	4.73	10	10	7	-211.26	-1271.75	0.0623	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.23	4.37
327	5322	5	3.67	10	10	7	-211.29	-1271.75	0.0368	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.68	2.99
328	5339	15	3.13	10	9.33	7	-211.50	-1272.70	0.0578	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	5.78	4.78
329	5356	8	3.67	10	10	7	-211.27	-1271.75	0.0513	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.13	2.45
330	5376	7	3.13	10	10	10.33	-211.69	-1273.43	0.0456	20.92	17.40	10.68	8.95	9.02	9.33	0.4435	0.4478	0.4768	4.56	2.80
331	5398	7	3.67	10	9.33	7	-211.51	-1272.70	0.0505	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	5.05	3.13
332	5415	14	3.13	10	10	8.33	-211.52	-1272.80	0.0599	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.99	4.03
333	5432	7	3.67	8.8	10	7	-212.18	-1275.43	0.0413	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	4.13	2.18
334	5448	11	3.13	9.4	10	7	-211.90	-1274.35	0.0556	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.56	2.97
335	5467	14	1.27	10	10	7	-211.27	-1271.75	0.0545	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.45	3.85
336	5483	10	3.13	10	10	7	-211.27	-1271.75	0.0565	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.65	3.89
337	5501	14	3.13	10	8.67	11	-212.26	-1275.87	0.0640	20.92	17.61	10.62	8.95	7.92	9.87	0.4435	0.3941	0.5045	6.40	4.36
338	5521	13	3.13	10	10	7	-211.26	-1271.75	0.0680	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.80	3.86
339	5537	12	3.4	10	10	7	-211.26	-1271.75	0.0640	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.40	3.09
340	5554	7	3.67	10	10	8.33	-211.53	-1272.80	0.0525	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.25	2.73
341	5575	11	1	10	10	7	-211.27	-1271.75	0.0523	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.23	2.95
342	5590	7	3.67	7.6	10	7	-213.02	-1278.87	0.0422	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	4.22	1.93
343	5610	12	3.13	10	8.67	7	-211.73	-1273.75	0.0768	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	7.68	3.64
344	5627	5	1	5.2	10	7	-214.28	-1283.92	0.0240	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	2.40	1.78
345	5644	13	3.13	10	9.33	7	-211.49	-1272.70	0.0659	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.59	3.40
346	5659	3	1.53	10	10	7.33	-211.42	-1272.18	0.0178	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	1.78	1.33
347	5674	14	3.4	5.2	10	7	-214.22	-1283.92	0.0682	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	6.82	4.10
348	5693	12	1	7.6	10	7	-213.00	-1278.87	0.0574	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	5.74	3.02
349	5709	7	1.8	8.8	10	7	-212.19	-1275.43	0.0366	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	3.66	1.90
350	5726	14	3.13	9.4	10	7	-211.90	-1274.35	0.0581	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.81	3.96

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
351	5744	6	3.13	10	10	7	-211.29	-1271.75	0.0348	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.48	2.17
352	5763	10	1.53	10	10	7	-211.28	-1271.75	0.0503	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.03	3.25
353	5780	5	1	10	9.33	7	-211.54	-1272.70	0.0202	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	2.02	1.79
354	5795	14	3.4	10	10	9.67	-211.62	-1273.23	0.0653	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.53	4.23
355	5812	5	4.2	10	9.33	7	-211.53	-1272.70	0.0316	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	3.16	2.49
356	5830	14	3.4	9.4	10	7	-211.90	-1274.35	0.0586	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.86	3.98
357	5847	5	4.73	10	10	7	-211.28	-1271.75	0.0429	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.29	3.50
358	5864	14	1.53	10	10	7.33	-211.38	-1272.18	0.0551	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	5.51	3.90
359	5881	4	3.13	10	10	7	-211.30	-1271.75	0.0285	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.85	1.69
360	5898	10	3.4	10	10	7	-211.27	-1271.75	0.0575	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.75	3.98
361	5916	13	3.67	10	10	7	-211.25	-1271.75	0.0699	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.99	3.98
362	5934	3	1	10	9.67	7	-211.45	-1272.31	0.0153	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	1.53	1.21
363	5955	15	1	10	7.33	7	-212.28	-1275.92	0.0657	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	6.57	5.05
364	5971	10	3.4	10	9.67	7	-211.40	-1272.31	0.0595	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	5.95	2.99
365	5988	10	3.4	9.4	10	7	-211.90	-1274.35	0.0581	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.81	3.37
366	6008	5	4.73	10	9.67	7	-211.42	-1272.31	0.0404	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	4.04	1.94
367	6023	15	4.2	10	10	7	-211.25	-1271.75	0.0716	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.16	5.10
368	6041	5	3.13	5.2	10	7	-214.26	-1283.92	0.0383	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	3.83	2.49
369	6056	5	4.2	10	8.67	7	-211.78	-1273.75	0.0365	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	3.65	2.65
370	6071	4	3.13	9.4	10	7	-211.93	-1274.35	0.0335	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	3.35	1.95
371	6088	15	3.13	10	9.67	7	-211.39	-1272.31	0.0666	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.66	5.07
372	6105	13	4.2	10	8.67	7	-211.75	-1273.75	0.0588	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.87	3.22
373	6121	12	4.2	10	10	7	-211.26	-1271.75	0.0654	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.54	3.13
374	6145	15	4.2	10	8.67	7	-211.75	-1273.75	0.0632	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	6.32	4.96
375	6161	8	4.2	10	10	7	-211.27	-1271.75	0.0532	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.32	2.47
376	6177	10	3.4	10	9.33	7	-211.49	-1272.70	0.0601	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.01	2.98
377	6195	5	4.73	10	10	7.67	-211.48	-1272.53	0.0374	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	3.74	2.74
378	6213	7	3.4	10	10	7	-211.27	-1271.75	0.0527	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.27	2.54
379	6230	10	4.2	10	10	7	-211.26	-1271.75	0.0606	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.06	4.27
380	6247	7	3.13	7.6	10	7	-213.02	-1278.87	0.0410	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	4.10	1.91
381	6265	3	1	10	7.33	7	-212.34	-1275.92	0.0138	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	1.38	1.04
382	6282	5	3.13	7.6	10	7	-213.04	-1278.87	0.0251	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	2.51	1.91
383	6301	8	3.4	10	10	7	-211.28	-1271.75	0.0503	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.03	2.44
384	6321	10	3.4	7.6	10	7	-213.00	-1278.87	0.0627	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	6.27	3.32
385	6339	13	4.2	10	10	7	-211.25	-1271.75	0.0718	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.18	4.08
386	6356	10	4.2	9.4	9.33	7	-212.14	-1275.29	0.0521	21.08	17.50	10.99	8.48	8.47	6.55	0.4209	0.4223	0.3357	5.21	3.13
387	6373	7	1	10	7.33	7.33	-212.42	-1276.35	0.0330	20.92	17.83	10.97	8.95	6.78	6.84	0.4435	0.3363	0.3509	3.30	1.95
388	6390	8	3.67	10	10	8.33	-211.54	-1272.80	0.0430	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	4.30	2.26
389	6407	6	4.73	10	10	7	-211.29	-1271.75	0.0400	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.00	2.33
390	6422	5	4.73	10	10	9.67	-211.66	-1273.23	0.0328	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	3.28	2.07
391	6438	5	3.4	10	10	7	-211.29	-1271.75	0.0353	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.53	2.88
392	6454	13	4.2	10	9.67	7	-211.38	-1272.31	0.0754	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	7.54	3.78
393	6471	7	3.4	9.4	10	7	-211.91	-1274.35	0.0532	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.32	2.47
394	6489	8	3.13	10	8.67	7	-211.75	-1273.75	0.0625	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	6.25	2.71
395	6509	5	3.67	8.8	10	7	-212.19	-1275.43	0.0327	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	3.27	2.57
396	6526	8	3.13	10	10	8.33	-211.54	-1272.80	0.0421	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	4.21	2.22
397	6543	5	4.47	10	10	7	-211.29	-1271.75	0.0414	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.14	3.35
398	6560	9	3.67	10	10	7	-211.28	-1271.75	0.0490	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.90	2.83
399	6577	3	3.4	10	10	7	-211.31	-1271.75	0.0250	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	2.50	1.75
400	6595	9	3.67	9.4	10	7	-211.91	-1274.35	0.0525	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.25	3.12

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
401	6611	5	4.2	10	10	8.33	-211.56	-1272.80	0.0291	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	2.91	2.09
402	6627	5	4.2	8.8	10	7	-212.19	-1275.43	0.0350	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	3.50	2.60
403	6647	9	4.2	10	10	7	-211.28	-1271.75	0.0497	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.97	2.85
404	6666	5	4.47	10	10	8.33	-211.56	-1272.80	0.0299	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	2.99	2.12
405	6685	7	1	10	10	7.33	-211.40	-1272.18	0.0371	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	3.71	1.93
406	6706	8	4.73	10	10	7	-211.27	-1271.75	0.0551	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.51	2.50
407	6723	13	4.47	10	10	7	-211.25	-1271.75	0.0727	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.27	4.13
408	6739	7	3.13	2.8	10	7	-215.30	-1288.26	0.0561	22.33	17.40	10.99	2.70	9.02	6.55	0.1350	0.4478	0.3357	5.61	3.32
409	6756	8	1	10	9.67	7	-211.42	-1272.31	0.0423	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	4.23	2.19
410	6772	6	1	10	9.67	7	-211.44	-1272.31	0.0283	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	2.83	1.94
411	6790	5	1	10	9.67	7	-211.45	-1272.31	0.0211	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	2.11	1.80
412	6809	5	4.2	10	7.33	7	-212.32	-1275.92	0.0298	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	2.98	1.99
413	6827	12	4.73	10	10	7	-211.26	-1271.75	0.0663	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.63	3.15
414	6843	5	4.2	10	9.67	7	-211.43	-1272.31	0.0376	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	3.76	1.98
415	6862	7	1	10	10	9.67	-211.65	-1273.23	0.0362	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	3.62	1.80
416	6881	5	3.4	10	10	8.33	-211.56	-1272.80	0.0267	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	2.67	1.97
417	6898	10	1	10	10	7	-211.28	-1271.75	0.0483	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.83	3.06
418	6914	10	3.4	10	10	7.67	-211.47	-1272.53	0.0469	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	4.69	2.89
419	6932	5	2.6	10	10	7	-211.30	-1271.75	0.0306	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.06	2.55
420	6949	13	2.33	10	10	7	-211.26	-1271.75	0.0652	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.52	3.69
421	6965	13	4.47	10	10	8.33	-211.52	-1272.80	0.0591	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	5.91	3.31
422	6982	15	4.2	10	10	7.33	-211.36	-1272.18	0.0718	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	7.18	5.10
423	6998	15	4.2	7.6	10	7	-212.97	-1278.87	0.0854	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	8.54	5.33
424	7014	8	3.4	5.2	10	7	-214.23	-1283.92	0.0674	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	6.74	3.59
425	7031	15	4.2	10	10	7.67	-211.44	-1272.53	0.0696	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	6.96	5.02
426	7047	5	4.73	10	10	7.33	-211.39	-1272.18	0.0454	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	4.54	2.73
427	7064	8	3.4	10	9.33	7	-211.51	-1272.70	0.0484	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	4.84	2.53
428	7088	15	4.2	10	10	8.33	-211.51	-1272.80	0.0659	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.59	4.95
429	7105	13	2.33	9.4	10	7	-211.90	-1274.35	0.0601	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.01	3.31
430	7123	13	2.33	10	10	11	-211.78	-1273.87	0.0634	20.92	17.40	10.62	8.95	9.02	9.87	0.4435	0.4478	0.5045	6.34	3.48
431	7142	14	1	8.8	10	7.33	-212.27	-1275.86	0.0545	21.18	17.40	10.97	7.99	9.02	6.84	0.3961	0.4478	0.3509	5.45	3.73
432	7159	9	4.47	10	10	7	-211.28	-1271.75	0.0501	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.01	2.86
433	7175	8	4.47	10	10	7	-211.27	-1271.75	0.0542	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.42	2.48
434	7193	5	4.2	7.6	10	7	-213.04	-1278.87	0.0277	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	2.77	2.06
435	7212	8	1.27	10	10	7.67	-211.48	-1272.53	0.0401	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	4.01	2.09
436	7230	8	3.13	9.4	10	7	-211.90	-1274.35	0.0617	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.17	2.73
437	7247	8	4.2	10	10	8.33	-211.54	-1272.80	0.0438	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	4.38	2.29
438	7268	8	3.4	10	10	8.33	-211.54	-1272.80	0.0426	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	4.26	2.24
439	7291	8	3.4	8.8	10	7	-212.16	-1275.43	0.0609	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.09	2.53
440	7310	8	3.4	10	10	7.67	-211.47	-1272.53	0.0441	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	4.41	2.23
441	7328	8	3.93	10	10	7	-211.27	-1271.75	0.0522	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.22	2.46
442	7344	8	3.13	10	10	7.33	-211.39	-1272.18	0.0461	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	4.61	2.39
443	7364	12	3.4	10	10	9.67	-211.61	-1273.23	0.0704	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	7.04	3.29
444	7382	13	4.47	7.6	10	8.33	-213.25	-1279.92	0.0664	21.44	17.40	10.87	7.01	9.02	7.71	0.3485	0.4478	0.3964	6.64	3.43
445	7399	13	4.47	8.8	10	7.33	-212.27	-1275.86	0.0596	21.18	17.40	10.97	7.99	9.02	6.84	0.3961	0.4478	0.3509	5.96	3.41
446	7415	8	3.4	10	7.33	7	-212.30	-1275.92	0.0484	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	4.84	2.37
447	7434	12	5	10	10	7	-211.26	-1271.75	0.0668	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.68	3.16
448	7456	10	3.4	10	10	9.67	-211.63	-1273.23	0.0579	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	5.79	3.20
449	7480	13	4.47	10	7.33	7	-212.28	-1275.92	0.0635	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	6.35	3.50
450	7495	8	3.4	7.6	10	7	-213.00	-1278.87	0.0569	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	5.69	3.13

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
451	7514	6	5	10	10	7	-211.29	-1271.75	0.0409	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.09	2.36
452	7533	12	4.2	10	9.67	7	-211.39	-1272.31	0.0660	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.60	3.13
453	7552	12	3.4	9.4	10	7	-211.89	-1274.35	0.0651	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.51	3.06
454	7569	13	5	10	10	7	-211.25	-1271.75	0.0746	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.46	4.24
455	7586	6	4.47	10	10	7	-211.29	-1271.75	0.0392	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	3.92	2.30
456	7611	13	4.47	5.2	10	7	-214.19	-1283.92	0.0937	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	9.37	4.72
457	7627	13	5	10	8.67	7	-211.75	-1273.75	0.0594	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.94	3.25
458	7650	13	4.47	10	10	7.33	-211.36	-1272.18	0.0638	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.38	3.61
459	7666	15	5	10	10	7	-211.25	-1271.75	0.0751	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.51	5.15
460	7682	13	4.47	10	10	9.67	-211.62	-1273.23	0.0638	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.38	3.59
461	7698	12	3.4	10	10	8.33	-211.51	-1272.80	0.0652	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.52	3.09
462	7715	6	4.47	10	10	7.67	-211.48	-1272.53	0.0377	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	3.77	2.08
463	7732	13	4.47	8.8	10	7	-212.16	-1275.43	0.0644	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.44	3.43
464	7753	13	4.47	10	8.67	7	-211.75	-1273.75	0.0590	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	5.90	3.22
465	7771	13	4.47	10	10	7.67	-211.45	-1272.53	0.0636	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	6.36	3.61
466	7793	14	5	10	10	7	-211.26	-1271.75	0.0629	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.29	4.40
467	7809	15	4.47	10	9.33	7	-211.50	-1272.70	0.0594	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	5.94	4.89
468	7828	13	4.47	7.6	10	7	-212.98	-1278.87	0.0749	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	7.49	4.50
469	7845	12	5	10	9.33	7	-211.49	-1272.70	0.0680	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.80	3.15
470	7863	15	5	5.2	10	7	-214.20	-1283.92	0.0860	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	8.60	5.87
471	7881	12	4.47	10	10	7	-211.26	-1271.75	0.0659	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.59	3.14
472	7900	13	4.47	9.4	10	8.33	-212.14	-1275.40	0.0718	21.08	17.40	10.87	8.48	9.02	7.71	0.4209	0.4478	0.3964	7.18	4.03
473	7918	13	4.47	10	6.67	7	-212.56	-1277.17	0.0823	20.92	17.95	10.99	8.95	6.23	6.55	0.4435	0.3103	0.3357	8.23	4.61
474	7936	12	2.87	10	10	7	-211.26	-1271.75	0.0630	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.30	3.06
475	7956	14	4.47	10	10	7	-211.26	-1271.75	0.0617	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.17	4.34
476	7971	15	4.47	8.8	9.33	7	-212.36	-1276.37	0.0885	21.18	17.50	10.99	7.99	8.47	6.55	0.3961	0.4223	0.3357	8.85	5.28
477	7987	13	4.47	10	9.67	7	-211.38	-1272.31	0.0766	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	7.66	3.82
478	8003	13	4.2	5.2	10	7	-214.20	-1283.92	0.0915	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	9.15	4.66
479	8024	12	4.47	10	9.67	7	-211.39	-1272.31	0.0665	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.65	3.14
480	8043	15	4.47	10	10	7	-211.25	-1271.75	0.0728	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.28	5.12
481	8062	15	5	10	8.67	7	-211.75	-1273.75	0.0648	20.92	17.61	10.99	8.95	7.92	6.55	0.4435	0.3941	0.3357	6.48	4.99
482	8079	13	5	10	10	9.67	-211.62	-1273.23	0.0647	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	6.47	3.62
483	8100	13	5	10	9.67	7	-211.38	-1272.31	0.0790	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	7.90	3.92
484	8117	15	2.87	10	10	7	-211.26	-1271.75	0.0657	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.57	4.97
485	8137	13	5	10	9.33	7	-211.48	-1272.70	0.0717	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	7.17	3.53
486	8154	13	4.47	10	9.33	7	-211.48	-1272.70	0.0701	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	7.01	3.46
487	8172	9	4.47	10	9.67	7	-211.41	-1272.31	0.0545	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	5.45	3.09
488	8196	15	5	10	7.33	7	-212.24	-1275.92	0.0970	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	9.70	5.33
489	8218	12	3.93	10	10	7	-211.26	-1271.75	0.0649	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.49	3.11
490	8235	12	5	5.2	10	7	-214.23	-1283.92	0.0659	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	6.59	3.24
491	8252	15	5	9.4	10	7	-211.89	-1274.35	0.0703	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	7.03	5.00
492	8273	15	3.93	10	10	7	-211.25	-1271.75	0.0704	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.04	5.08
493	8289	9	4.47	10	8.67	8.33	-212.01	-1274.80	0.0545	20.92	17.61	10.87	8.95	7.92	7.71	0.4435	0.3941	0.3964	5.45	3.10
494	8308	11	4.47	10	10	7	-211.26	-1271.75	0.0604	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.04	3.31
495	8326	15	5	10	10	9.67	-211.61	-1273.23	0.0767	20.92	17.40	10.74	8.95	9.02	8.81	0.4435	0.4478	0.4500	7.67	5.40
496	8344	9	2.33	10	10	7	-211.28	-1271.75	0.0471	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	4.71	2.78
497	8360	9	4.47	9.4	10	7	-211.90	-1274.35	0.0546	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	5.46	3.28
498	8377	13	5	10	10	7.33	-211.36	-1272.18	0.0646	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.46	3.65
499	8409	15	5	10	10	7.33	-211.35	-1272.18	0.0754	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	7.54	5.15
500	8430	12	3.93	9.4	10	7	-211.89	-1274.35	0.0662	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.62	3.08

Calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	Goal1	Goal2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
501	8449	11	5	10	10	7.33	-211.37	-1272.18	0.0615	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.15	3.38
502	8467	15	4.73	10	10	7	-211.25	-1271.75	0.0739	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.39	5.14
503	8489	13	4.2	10	10	7.33	-211.37	-1272.18	0.0634	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.34	3.59
504	8516	13	4.47	5.2	10	7.67	-214.41	-1284.70	0.0717	21.89	17.40	10.94	4.90	9.02	7.14	0.2447	0.4478	0.3658	7.17	3.71
505	8534	15	5	10	9.67	7	-211.38	-1272.31	0.0745	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	7.45	5.26
506	8555	13	3.4	10	10	7	-211.25	-1271.75	0.0690	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.90	3.92
507	8579	13	5	8.8	10	7	-212.15	-1275.43	0.0653	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.53	3.44
508	8601	13	3.4	5.2	10	7	-214.20	-1283.92	0.0850	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	8.50	4.47
509	8620	15	4.47	10	10	7.67	-211.44	-1272.53	0.0706	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	7.06	5.04
510	8642	15	4.47	5.2	10	7	-214.21	-1283.92	0.0842	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	8.42	5.83
511	8663	15	3.4	10	10	7	-211.26	-1271.75	0.0681	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.81	5.03
512	8681	15	4.73	10	9.33	7	-211.50	-1272.70	0.0597	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	5.97	4.92
513	8700	13	5	7.6	10	7	-212.98	-1278.87	0.0772	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	7.72	4.65
514	8720	15	4.47	10	9.67	7	-211.39	-1272.31	0.0723	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	7.23	5.22
515	8740	15	5	10	10	8.33	-211.51	-1272.80	0.0680	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.80	4.98
516	8759	13	4.73	10	9.33	7	-211.48	-1272.70	0.0709	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	7.09	3.48
517	8782	13	5	5.2	10	7	-214.19	-1283.92	0.0980	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	9.80	4.92
518	8800	13	3.93	10	10	7	-211.25	-1271.75	0.0708	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.08	4.03
519	8819	15	4.47	7.6	10	7	-212.97	-1278.87	0.0876	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	8.76	5.35
520	8837	15	4.73	7.6	10	7.33	-213.07	-1279.30	0.0898	21.44	17.40	10.97	7.01	9.02	6.84	0.3485	0.4478	0.3509	8.98	5.35
521	8859	13	4.73	9.4	9.33	7	-212.12	-1275.29	0.0695	21.08	17.50	10.99	8.48	8.47	6.55	0.4209	0.4223	0.3357	6.95	3.55
522	8875	8	5	10	10	7	-211.27	-1271.75	0.0561	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	5.61	2.54
523	8895	13	2.87	10	10	7	-211.26	-1271.75	0.0671	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.71	3.80
524	8913	13	4.2	10	9.33	7	-211.48	-1272.70	0.0692	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.92	3.44
525	8934	13	4.2	8.8	10	7	-212.16	-1275.43	0.0639	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.39	3.42
526	8965	15	4.47	10	10	8.33	-211.51	-1272.80	0.0666	20.92	17.40	10.87	8.95	9.02	7.71	0.4435	0.4478	0.3964	6.66	4.96
527	8994	13	4.73	10	10	7	-211.25	-1271.75	0.0736	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	7.36	4.19
528	9012	15	4.47	10	7.33	7	-212.24	-1275.92	0.0929	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	9.29	5.35
529	9029	15	4.47	9.4	10	7	-211.89	-1274.35	0.0685	21.08	17.40	10.99	8.48	9.02	6.55	0.4209	0.4478	0.3357	6.85	4.98
530	9053	15	5	10	9.33	7	-211.49	-1272.70	0.0601	20.92	17.50	10.99	8.95	8.47	6.55	0.4435	0.4223	0.3357	6.01	4.94
531	9075	15	3.93	10	7.33	7	-212.25	-1275.92	0.0886	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	8.86	5.39
532	9098	15	5	4	10	7	-214.84	-1286.42	0.0603	22.12	17.40	10.99	3.82	9.02	6.55	0.1912	0.4478	0.3357	6.03	4.47
533	9121	15	5	5.2	7.33	7	-215.23	-1288.09	0.0760	21.89	17.83	10.99	4.90	6.78	6.55	0.2447	0.3363	0.3357	7.60	5.11
534	9162	15	5	8.8	10	7	-212.15	-1275.43	0.0675	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.75	4.93
535	9181	13	2.87	10	10	7.33	-211.37	-1272.18	0.0615	20.92	17.40	10.97	8.95	9.02	6.84	0.4435	0.4478	0.3509	6.15	3.48
536	9199	13	2.87	10	10	7.67	-211.45	-1272.53	0.0613	20.92	17.40	10.94	8.95	9.02	7.14	0.4435	0.4478	0.3658	6.13	3.49
537	9217	15	5	7.6	10	7	-212.96	-1278.87	0.0917	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	9.17	5.38
538	9239	14	3.93	5.2	9.67	7	-214.36	-1284.48	0.0639	21.89	17.45	10.99	4.90	8.75	6.55	0.2447	0.4338	0.3357	6.39	3.85
539	9263	11	5	10	10	7	-211.26	-1271.75	0.0616	20.92	17.40	10.99	8.95	9.02	6.55	0.4435	0.4478	0.3357	6.16	3.36
540	9287	13	2.33	10	7.33	7	-212.28	-1275.92	0.0583	20.92	17.83	10.99	8.95	6.78	6.55	0.4435	0.3363	0.3357	5.83	3.24
541	9303	13	2.87	5.2	10	7	-214.21	-1283.92	0.0807	21.89	17.40	10.99	4.90	9.02	6.55	0.2447	0.4478	0.3357	8.07	4.29
542	9325	15	4.47	8.8	10	7	-212.15	-1275.43	0.0660	21.18	17.40	10.99	7.99	9.02	6.55	0.3961	0.4478	0.3357	6.60	4.90
543	9352	11	5	10	9.67	7	-211.40	-1272.31	0.0606	20.92	17.45	10.99	8.95	8.75	6.55	0.4435	0.4338	0.3357	6.06	3.09
544	9372	14	4.2	7.6	10	7	-212.99	-1278.87	0.0685	21.44	17.40	10.99	7.01	9.02	6.55	0.3485	0.4478	0.3357	6.85	4.17

PERT Domain: GA-SO Run Resulting in the BSF (Run 1)

Calls	CPU Time	LOW3	LOW4	LOW6	Z	Goal1	Goal2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
1	2	7.8	7.53	6.6	-0.1559	-21.76	-20.85	0.6875	0.1825	0.1300	0.2775	0.4100	0.1975	0.3950	0.2775	0.5250	20.85	0.1009
2	8	7.33	6.13	5.67	-0.1493	-21.76	-20.73	0.6550	0.2025	0.1425	0.2250	0.4300	0.1900	0.4425	0.2250	0.5850	20.73	0.1046
3	10	9.2	1	2.87	-0.1399	-21.76	-20.56	0.5850	0.2100	0.2050	0.1300	0.4550	0.1525	0.5125	0.1300	0.7175	20.56	0.1103
4	12	6.4	7.07	1	-0.4459	-21.83	-20.70	0.7025	0.1700	0.1275	0.3075	0.3950	0.1250	0.4400	0.3075	0.5675	20.70	0.1034
5	14	6.87	5.2	7.53	-0.1503	-21.76	-20.75	0.6475	0.2125	0.1400	0.2025	0.4450	0.2325	0.4250	0.2025	0.5650	20.75	0.1049
6	16	6.87	5.2	4.27	-0.1446	-21.76	-20.65	0.6450	0.2050	0.1500	0.2100	0.4350	0.1700	0.4700	0.2100	0.6200	20.65	0.1069
7	17	8.27	3.8	1.47	-0.1407	-21.76	-20.58	0.6125	0.2025	0.1850	0.1775	0.4350	0.1400	0.4975	0.1775	0.6825	20.58	0.1088
8	19	10.6	5.67	7.07	-0.1555	-21.76	-20.85	0.6050	0.1925	0.2025	0.1875	0.4175	0.2050	0.4050	0.1875	0.6075	20.85	0.1024
9	21	11.07	4.27	7.53	-0.1548	-21.76	-20.83	0.5850	0.2000	0.2150	0.1575	0.4275	0.2150	0.4125	0.1575	0.6275	20.83	0.1034
10	23	11.07	4.27	5.67	-0.1515	-21.76	-20.77	0.5850	0.1900	0.2250	0.1625	0.4225	0.1750	0.4375	0.1625	0.6625	20.77	0.1043
11	25	10.6	6.13	4.27	-0.1524	-21.76	-20.79	0.6050	0.1825	0.2125	0.2100	0.3950	0.1500	0.4275	0.2100	0.6400	20.79	0.1029
12	27	9.2	1.47	7.07	-0.1466	-21.76	-20.68	0.5925	0.2200	0.1875	0.1250	0.4675	0.2275	0.4600	0.1250	0.6475	20.68	0.1080
13	29	8.27	1.93	6.6	-0.1445	-21.76	-20.65	0.6050	0.2175	0.1775	0.1375	0.4675	0.2225	0.4625	0.1375	0.6400	20.65	0.1088
14	31	5.93	1.93	3.8	-0.1373	-21.76	-20.52	0.6350	0.2125	0.1525	0.1600	0.4750	0.1700	0.5175	0.1600	0.6700	20.52	0.1115
15	33	10.13	4.27	5.2	-0.1488	-21.76	-20.72	0.5975	0.1950	0.2075	0.1675	0.4300	0.1750	0.4500	0.1675	0.6575	20.72	0.1055
16	35	5	1	6.6	-0.1399	-21.76	-20.56	0.6225	0.2325	0.1450	0.1400	0.4825	0.2350	0.4800	0.1400	0.6250	20.56	0.1110
17	37	5.93	3.8	5.2	-0.1421	-21.76	-20.60	0.6475	0.2100	0.1425	0.1850	0.4625	0.1875	0.4850	0.1850	0.6275	20.60	0.1088
18	39	7.33	1.93	1.93	-0.1370	-21.76	-20.51	0.6200	0.2050	0.1750	0.1575	0.4625	0.1475	0.5200	0.1575	0.6950	20.51	0.1113
19	41	6.87	5.67	6.13	-0.1484	-21.76	-20.72	0.6500	0.2050	0.1450	0.2125	0.4375	0.2000	0.4425	0.2125	0.5875	20.72	0.1053
20	42	10.6	3.33	7.07	-0.1514	-21.76	-20.77	0.5875	0.2025	0.2100	0.1425	0.4450	0.2125	0.4350	0.1425	0.6450	20.77	0.1052
21	44	5.47	5.2	3.33	-0.1422	-21.76	-20.61	0.6575	0.2050	0.1375	0.2225	0.4350	0.1525	0.4875	0.2225	0.6250	20.61	0.1076
22	46	11.07	6.13	7.53	-0.1584	-21.76	-20.90	0.6025	0.1900	0.2075	0.1900	0.4125	0.2100	0.3925	0.1900	0.6000	20.90	0.1010
23	48	8.73	7.07	4.73	-0.1526	-21.76	-20.79	0.6700	0.1775	0.1525	0.2750	0.3950	0.1550	0.4175	0.2750	0.5700	20.79	0.1022
24	50	5.47	1.47	1.47	-0.1338	-21.76	-20.45	0.6300	0.2125	0.1575	0.1575	0.4725	0.1500	0.5350	0.1575	0.6925	20.45	0.1127
25	52	7.33	3.8	5.2	-0.1436	-21.76	-20.63	0.6325	0.2100	0.1575	0.1800	0.4525	0.1850	0.4775	0.1800	0.6350	20.63	0.1082
26	54	10.13	1.93	3.33	-0.1433	-21.76	-20.63	0.5800	0.1950	0.2250	0.1400	0.4400	0.1525	0.4825	0.1400	0.7075	20.63	0.1084
27	56	9.2	7.53	1	-0.1521	-21.76	-20.78	0.6750	0.1650	0.1600	0.3000	0.3750	0.1200	0.4200	0.3000	0.5800	20.78	0.1013
28	58	9.2	8	1	-0.1541	-21.76	-20.82	0.6850	0.1600	0.1550	0.3125	0.3725	0.1200	0.4125	0.3125	0.5675	20.82	0.1000
29	60	11.53	7.07	6.6	-0.1600	-21.76	-20.93	0.6100	0.1750	0.2150	0.2225	0.3875	0.1875	0.3750	0.2225	0.5900	20.93	0.0996
30	62	10.6	6.6	2.87	-0.1525	-21.76	-20.79	0.6150	0.1750	0.2100	0.2300	0.3850	0.1350	0.4250	0.2300	0.6350	20.79	0.1023
31	64	8.73	7.07	7.53	-0.1570	-21.76	-20.87	0.6625	0.1925	0.1450	0.2450	0.4175	0.2225	0.3875	0.2450	0.5325	20.87	0.1012
32	66	11.07	4.27	4.73	-0.1503	-21.76	-20.75	0.5825	0.1900	0.2275	0.1625	0.4200	0.1650	0.4450	0.1625	0.6725	20.75	0.1046
33	68	7.33	3.33	4.73	-0.1421	-21.76	-20.60	0.6275	0.2100	0.1625	0.1675	0.4600	0.1775	0.4925	0.1675	0.6550	20.60	0.1089
34	70	8.73	7.53	5.2	-0.1549	-21.76	-20.83	0.6750	0.1750	0.1500	0.2825	0.3925	0.1600	0.4075	0.2825	0.5575	20.83	0.1009
35	71	5.47	1.47	7.07	-0.1420	-21.76	-20.60	0.6225	0.2325	0.1450	0.1400	0.4825	0.2400	0.4750	0.1400	0.6200	20.60	0.1100
36	73	9.2	1.47	1.47	-0.1391	-21.76	-20.55	0.5875	0.2075	0.2050	0.1350	0.4525	0.1450	0.5150	0.1350	0.7200	20.55	0.1103
37	75	8.73	6.13	4.27	-0.1491	-21.76	-20.73	0.6375	0.1950	0.1675	0.2250	0.4125	0.1575	0.4500	0.2250	0.6175	20.73	0.1045
38	77	10.6	7.07	4.73	-0.1556	-21.76	-20.85	0.6275	0.1700	0.2025	0.2500	0.3775	0.1475	0.4000	0.2500	0.6025	20.85	0.1010
39	79	12	1.93	6.6	-0.1526	-21.76	-20.79	0.5550	0.1925	0.2525	0.1200	0.4350	0.2000	0.4275	0.1200	0.6800	20.79	0.1048
40	81	9.2	7.53	6.6	-0.1574	-21.76	-20.88	0.6725	0.1775	0.1500	0.2675	0.4050	0.1950	0.3875	0.2675	0.5375	20.88	0.1004

Calls	CPU Time	LOW3	LOW4	LOW6	Z	Goal1	Goal2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
41	83	7.8	7.53	1	-0.1504	-21.76	-20.75	0.6975	0.1675	0.1350	0.3125	0.3850	0.1225	0.4300	0.3125	0.5650	20.75	0.1018
42	85	10.6	6.13	6.6	-0.1556	-21.76	-20.85	0.6075	0.1900	0.2025	0.1925	0.4150	0.2000	0.4050	0.1925	0.6075	20.85	0.1021
43	87	7.8	7.53	4.27	-0.1529	-21.76	-20.80	0.6900	0.1775	0.1325	0.2975	0.3925	0.1500	0.4200	0.2975	0.5525	20.80	0.1014
44	89	11.07	1.93	3.33	-0.1457	-21.76	-20.67	0.5675	0.1925	0.2400	0.1350	0.4325	0.1525	0.4725	0.1350	0.7125	20.67	0.1072
45	91	10.13	4.27	7.53	-0.1528	-21.76	-20.80	0.6000	0.2050	0.1950	0.1625	0.4375	0.2250	0.4175	0.1625	0.6125	20.80	0.1043
46	93	5.47	1.47	5.2	-0.1382	-21.76	-20.53	0.6325	0.2175	0.1500	0.1550	0.4775	0.1875	0.5075	0.1550	0.6575	20.53	0.1115
47	95	5	2.87	1.47	-0.1353	-21.76	-20.48	0.6425	0.2075	0.1500	0.1800	0.4625	0.1475	0.5225	0.1800	0.6725	20.48	0.1114
48	97	5.47	1.47	6.6	-0.1408	-21.76	-20.58	0.6225	0.2325	0.1450	0.1400	0.4825	0.2350	0.4800	0.1400	0.6250	20.58	0.1105
49	99	9.2	7.07	5.2	-0.1537	-21.76	-20.81	0.6650	0.1775	0.1575	0.2700	0.3950	0.1625	0.4100	0.2700	0.5675	20.81	0.1020
50	100	5.47	1	1	-0.1328	-21.76	-20.44	0.6275	0.2125	0.1600	0.1525	0.4750	0.1450	0.5425	0.1525	0.7025	20.44	0.1132
51	102	7.8	7.53	5.2	-0.1539	-21.76	-20.82	0.6900	0.1800	0.1300	0.2925	0.3975	0.1625	0.4150	0.2925	0.5450	20.82	0.1012
52	104	8.73	7.53	6.6	-0.1568	-21.76	-20.87	0.6725	0.1800	0.1475	0.2675	0.4050	0.1975	0.3875	0.2675	0.5350	20.87	0.1006
53	106	9.67	4.27	7.53	-0.1520	-21.76	-20.78	0.6075	0.2100	0.1825	0.1675	0.4400	0.2275	0.4225	0.1675	0.6050	20.78	0.1047
54	108	5.47	1.47	4.27	-0.1368	-21.76	-20.51	0.6325	0.2175	0.1500	0.1550	0.4775	0.1775	0.5175	0.1550	0.6675	20.51	0.1119
55	110	5	6.13	7.07	-0.1500	-21.76	-20.75	0.6725	0.2075	0.1200	0.2300	0.4425	0.2225	0.4275	0.2300	0.5475	20.75	0.1043
56	112	6.87	5.2	1.47	-0.1419	-21.76	-20.60	0.6450	0.1975	0.1575	0.2225	0.4225	0.1375	0.4825	0.2225	0.6400	20.60	0.1075
57	114	8.73	6.6	6.6	-0.1537	-21.76	-20.81	0.6525	0.1950	0.1525	0.2275	0.4250	0.2100	0.4100	0.2275	0.5625	20.81	0.1029
58	116	5.47	1.47	7.53	-0.1432	-21.76	-20.62	0.6200	0.2350	0.1450	0.1350	0.4850	0.2550	0.4650	0.1350	0.6100	20.62	0.1095
59	118	5	1.47	1.47	-0.1333	-21.76	-20.45	0.6325	0.2125	0.1550	0.1575	0.4750	0.1525	0.5350	0.1575	0.6900	20.45	0.1129
60	120	5.47	1	6.6	-0.1403	-21.76	-20.57	0.6225	0.2325	0.1450	0.1400	0.4825	0.2350	0.4800	0.1400	0.6250	20.57	0.1109
61	122	5.47	1.47	1	-0.1334	-21.76	-20.45	0.6300	0.2100	0.1600	0.1575	0.4725	0.1450	0.5375	0.1575	0.6975	20.45	0.1128
62	124	5	7.07	4.73	-0.1493	-21.76	-20.73	0.7000	0.1825	0.1175	0.2950	0.4050	0.1575	0.4300	0.2950	0.5475	20.73	0.1031
63	126	8.73	1.93	6.6	-0.1453	-21.76	-20.66	0.6025	0.2150	0.1825	0.1350	0.4675	0.2225	0.4600	0.1350	0.6425	20.66	0.1084
64	128	5	1	1	-0.1324	-21.76	-20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.43	0.1134
65	130	5	1	7.07	-0.1410	-21.76	-20.58	0.6200	0.2350	0.1450	0.1350	0.4850	0.2450	0.4750	0.1350	0.6200	20.58	0.1105
66	132	5	1.47	1	-0.1329	-21.76	-20.44	0.6300	0.2125	0.1575	0.1575	0.4725	0.1475	0.5375	0.1575	0.6950	20.44	0.1130
67	134	8.73	1	6.6	-0.1443	-21.76	-20.64	0.5925	0.2200	0.1875	0.1250	0.4675	0.2225	0.4650	0.1250	0.6525	20.64	0.1090
68	136	5	1	4.73	-0.1365	-21.76	-20.50	0.6300	0.2200	0.1500	0.1500	0.4800	0.1800	0.5200	0.1500	0.6700	20.50	0.1123
69	138	7.33	1.93	1.47	-0.1366	-21.76	-20.50	0.6200	0.2050	0.1750	0.1575	0.4625	0.1475	0.5200	0.1575	0.6950	20.50	0.1113
70	140	5.47	1.47	1.93	-0.1342	-21.76	-20.46	0.6325	0.2125	0.1550	0.1575	0.4750	0.1525	0.5350	0.1575	0.6900	20.46	0.1126
71	142	7.33	1.47	1.47	-0.1361	-21.76	-20.49	0.6100	0.2100	0.1800	0.1475	0.4625	0.1475	0.5250	0.1475	0.7050	20.49	0.1117
72	144	5	1.93	1.93	-0.1344	-21.76	-20.46	0.6375	0.2100	0.1525	0.1625	0.4750	0.1550	0.5300	0.1625	0.6825	20.46	0.1123
73	146	5	1	5.2	-0.1372	-21.76	-20.51	0.6275	0.2250	0.1475	0.1475	0.4800	0.1925	0.5125	0.1475	0.6600	20.51	0.1121
74	147	5	1	1.47	-0.1328	-21.76	-20.44	0.6300	0.2150	0.1550	0.1525	0.4775	0.1525	0.5400	0.1525	0.6950	20.44	0.1133
75	149	6.87	5.2	1	-0.1416	-21.76	-20.59	0.6450	0.1975	0.1575	0.2225	0.4225	0.1375	0.4825	0.2225	0.6400	20.59	0.1076
76	151	5	1.47	4.27	-0.1364	-21.76	-20.50	0.6325	0.2175	0.1500	0.1550	0.4775	0.1775	0.5175	0.1550	0.6675	20.50	0.1121
77	153	5.93	1	6.6	-0.1407	-21.76	-20.58	0.6225	0.2325	0.1450	0.1400	0.4825	0.2350	0.4800	0.1400	0.6250	20.58	0.1107
78	155	5.47	5.2	1	-0.1402	-21.76	-20.57	0.6600	0.1975	0.1425	0.2300	0.4300	0.1375	0.4900	0.2300	0.6325	20.57	0.1080
79	157	5.47	1.47	3.33	-0.1357	-21.76	-20.49	0.6325	0.2175	0.1500	0.1550	0.4775	0.1675	0.5275	0.1550	0.6775	20.49	0.1123
80	160	5	1	1.93	-0.1332	-21.76	-20.44	0.6300	0.2175	0.1525	0.1525	0.4775	0.1550	0.5400	0.1525	0.6925	20.44	0.1132

Calls	CPU Time	LOW3	LOW4	LOW6	Z	Goal1	Goal2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
81	162	5	2.4	1	-0.2147	-21.78	-20.46	0.6400	0.2075	0.1525	0.1750	0.4650	0.1425	0.5300	0.1750	0.6825	20.46	0.1120
82	163	5	1.47	2.87	-0.1347	-21.76	-20.47	0.6325	0.2175	0.1500	0.1550	0.4775	0.1675	0.5275	0.1550	0.6775	20.47	0.1126
83	165	7.33	1.47	1.93	-0.1365	-21.76	-20.50	0.6100	0.2100	0.1800	0.1475	0.4625	0.1475	0.5250	0.1475	0.7050	20.50	0.1117
84	167	5.47	1.47	2.87	-0.1351	-21.76	-20.48	0.6325	0.2150	0.1525	0.1550	0.4775	0.1650	0.5275	0.1550	0.6800	20.48	0.1124
85	169	5.47	1	1.93	-0.1337	-21.76	-20.45	0.6300	0.2150	0.1550	0.1525	0.4775	0.1525	0.5400	0.1525	0.6950	20.45	0.1130
86	171	7.33	1.47	1	-0.1357	-21.76	-20.49	0.6100	0.2100	0.1800	0.1475	0.4625	0.1450	0.5275	0.1475	0.7075	20.49	0.1118
87	173	5.93	1.47	2.87	-0.1356	-21.76	-20.49	0.6300	0.2150	0.1550	0.1550	0.4750	0.1650	0.5250	0.1550	0.6800	20.49	0.1123
88	175	5.93	1	1	-0.1333	-21.76	-20.45	0.6225	0.2125	0.1650	0.1500	0.4725	0.1450	0.5400	0.1500	0.7050	20.45	0.1130
89	177	5	1.47	6.6	-0.1404	-21.76	-20.57	0.6225	0.2325	0.1450	0.1400	0.4825	0.2350	0.4800	0.1400	0.6250	20.57	0.1107
90	179	5.47	1.93	1.47	-0.1344	-21.76	-20.46	0.6350	0.2075	0.1575	0.1625	0.4725	0.1500	0.5300	0.1625	0.6875	20.46	0.1122
91	181	7.33	2.87	1	-0.1376	-21.76	-20.52	0.6225	0.2050	0.1725	0.1725	0.4500	0.1400	0.5150	0.1725	0.6875	20.52	0.1105
92	183	5.93	1.47	1.47	-0.1343	-21.76	-20.46	0.6275	0.2100	0.1625	0.1575	0.4700	0.1475	0.5325	0.1575	0.6950	20.46	0.1125
93	185	5.47	1.93	1	-0.1340	-21.76	-20.46	0.6350	0.2050	0.1600	0.1650	0.4700	0.1425	0.5325	0.1650	0.6925	20.46	0.1123
94	187	5	4.73	2.4	-0.1397	-21.76	-20.56	0.6550	0.2050	0.1400	0.2150	0.4400	0.1525	0.4925	0.2150	0.6325	20.56	0.1088
95	189	5.93	1.47	1	-0.1339	-21.76	-20.46	0.6275	0.2100	0.1625	0.1575	0.4700	0.1450	0.5350	0.1575	0.6975	20.46	0.1126
96	191	5	1.47	1.93	-0.1338	-21.76	-20.45	0.6325	0.2150	0.1525	0.1575	0.4750	0.1550	0.5350	0.1575	0.6875	20.45	0.1128
97	193	8.73	1.47	1	-0.1379	-21.76	-20.53	0.5975	0.2075	0.1950	0.1425	0.4550	0.1425	0.5200	0.1425	0.7150	20.53	0.1107
98	196	5	2.87	1	-0.1349	-21.76	-20.47	0.6400	0.2075	0.1525	0.1800	0.4600	0.1425	0.5250	0.1800	0.6775	20.47	0.1115
99	198	5.47	1	1.47	-0.1332	-21.76	-20.44	0.6275	0.2150	0.1575	0.1525	0.4750	0.1500	0.5400	0.1525	0.6975	20.44	0.1131
100	200	6.87	1	1	-0.1345	-21.76	-20.47	0.6100	0.2125	0.1775	0.1450	0.4650	0.1450	0.5325	0.1450	0.7100	20.47	0.1125
101	202	5	4.73	1	-0.1386	-21.76	-20.54	0.6550	0.2000	0.1450	0.2175	0.4375	0.1400	0.4975	0.2175	0.6425	20.54	0.1090
102	204	8.73	1	1	-0.1374	-21.76	-20.52	0.5875	0.2100	0.2025	0.1300	0.4575	0.1425	0.5250	0.1300	0.7275	20.52	0.1111
103	207	8.73	2.87	1	-0.1397	-21.76	-20.56	0.6075	0.2025	0.1900	0.1650	0.4425	0.1375	0.5075	0.1650	0.6975	20.56	0.1095
104	209	5	1.93	1	-0.1336	-21.76	-20.45	0.6350	0.2075	0.1575	0.1650	0.4700	0.1450	0.5325	0.1650	0.6900	20.45	0.1125
105	211	5	1	2.87	-0.1341	-21.76	-20.46	0.6300	0.2200	0.1500	0.1500	0.4800	0.1675	0.5325	0.1500	0.6825	20.46	0.1130
106	214	5	1	5.67	-0.1380	-21.76	-20.53	0.6275	0.2275	0.1450	0.1475	0.4800	0.2050	0.5025	0.1475	0.6475	20.53	0.1118
107	217	5	2.87	1.93	-0.1357	-21.76	-20.49	0.6425	0.2100	0.1475	0.1775	0.4650	0.1525	0.5225	0.1775	0.6700	20.49	0.1114
108	222	5	4.73	1.93	-0.1393	-21.76	-20.55	0.6550	0.2050	0.1400	0.2150	0.4400	0.1500	0.4950	0.2150	0.6350	20.55	0.1088
109	225	5.93	1	2.87	-0.1351	-21.76	-20.48	0.6250	0.2175	0.1575	0.1475	0.4775	0.1650	0.5300	0.1475	0.6875	20.48	0.1127
110	230	5	2.87	2.4	-0.1362	-21.76	-20.50	0.6425	0.2100	0.1475	0.1750	0.4675	0.1575	0.5200	0.1750	0.6675	20.50	0.1113
111	233	6.87	1	1.47	-0.1349	-21.76	-20.47	0.6100	0.2125	0.1775	0.1450	0.4650	0.1475	0.5300	0.1450	0.7075	20.47	0.1124
112	241	5	5.2	1	-0.1398	-21.76	-20.56	0.6625	0.1975	0.1400	0.2325	0.4300	0.1375	0.4900	0.2325	0.6300	20.56	0.1081
113	248	5	3.33	1	-0.3080	-21.80	-20.49	0.6450	0.2075	0.1475	0.1850	0.4600	0.1425	0.5250	0.1850	0.6725	20.49	0.1110
114	250	6.87	1.47	1	-0.1350	-21.76	-20.48	0.6125	0.2100	0.1775	0.1500	0.4625	0.1450	0.5275	0.1500	0.7050	20.48	0.1121
115	255	5	2.87	2.87	-0.1366	-21.76	-20.51	0.6425	0.2125	0.1450	0.1750	0.4675	0.1650	0.5150	0.1750	0.6600	20.51	0.1112
116	257	9.2	1	1	-0.1383	-21.76	-20.53	0.5850	0.2100	0.2050	0.1300	0.4550	0.1425	0.5225	0.1300	0.7275	20.53	0.1106
117	260	5	1	2.4	-0.1336	-21.76	-20.45	0.6300	0.2175	0.1525	0.1525	0.4775	0.1575	0.5375	0.1525	0.6900	20.45	0.1131
118	262	8.73	1	1.47	-0.1378	-21.76	-20.53	0.5875	0.2100	0.2025	0.1300	0.4575	0.1450	0.5225	0.1300	0.7250	20.53	0.1110
119	264	5.93	4.73	1	-0.1394	-21.76	-20.56	0.6525	0.2000	0.1475	0.2150	0.4375	0.1400	0.4975	0.2150	0.6450	20.56	0.1087
120	267	5	1.93	2.87	-0.1353	-21.76	-20.48	0.6375	0.2125	0.1500	0.1625	0.4750	0.1650	0.5225	0.1625	0.6725	20.48	0.1122
121	269	5.93	1	1.93	-0.1341	-21.76	-20.46	0.6250	0.2150	0.1600	0.1500	0.4750	0.1525	0.5375	0.1500	0.6975	20.46	0.1128
122	271	5.47	1	4.73	-0.1369	-21.76	-20.51	0.6300	0.2200	0.1500	0.1500	0.4800	0.1800	0.5200	0.1500	0.6700	20.51	0.1121
123	274	8.73	3.33	1	-0.1404	-21.76	-20.57	0.6075	0.2025	0.1900	0.1650	0.4425	0.1375	0.5075	0.1650	0.6975	20.57	0.1090

Production Domain: GA-SO Run Resulting in the BSF (Run 1)

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
1	37	8.33	14	7	4.33	3.6	6	-51.25	2242.72	1.900	22.60	56.58	0.9523	5.7343	0.0393	1.9602	0.0103
2	49	5.67	14.67	10.33	4.6	1.8	6	-49.90	2305.32	1.875	23.23	55.42	0.9291	5.5853	0.0408	1.8888	0.0182
3	60	7	13.33	11	3.53	2.2	2.8	-43.21	2614.63	1.799	26.34	52.78	0.9225	5.6847	0.0399	1.7040	0.0311
4	73	7	17.33	11	4.07	1.8	5.47	-54.60	2091.53	1.771	21.09	55.77	0.9597	5.7742	0.0386	1.6171	0.0339
5	84	5.67	20	15	7	2.8	6	-67.33	1502.86	1.900	15.19	63.30	0.9755	5.7742	0.0386	1.9372	0.0121
6	97	6.33	14.67	14.33	5.67	1	2.27	-50.79	2263.72	1.900	22.82	55.64	0.9550	5.7732	0.0386	1.9135	0.0180
7	114	10	18.67	12.33	7	2.2	4.93	-63.74	1668.17	1.900	16.84	61.54	0.9951	5.7742	0.0386	1.8844	0.0195
8	129	6.33	11.33	10.33	7	2.6	4.93	-49.78	2311.18	1.853	23.29	56.37	0.9115	5.7612	0.0388	1.7860	0.0255
9	145	6.33	12	6.33	4.33	4	6	-46.07	2482.55	1.827	25.01	53.75	0.9031	5.7742	0.0386	1.7286	0.0276
10	156	9.67	10.67	9	3.27	2.8	5.47	-39.85	2768.51	1.835	27.87	53.23	0.8918	5.7742	0.0386	1.7460	0.0282
11	170	7.33	17.33	12.33	5.67	1	4.13	-55.95	2030.19	1.744	20.47	58.07	0.9727	5.5876	0.0429	1.6261	0.0322
12	182	5.33	18	7.67	6.47	2	2.27	-55.48	2047.86	1.895	20.66	55.34	0.9915	5.5954	0.0411	1.9254	0.0142
13	195	9.33	18	7	4.6	2.2	4.93	-55.05	2072.34	1.717	20.90	56.60	0.9759	5.7742	0.0386	1.5099	0.0379
14	208	10	16	11.67	7	1	3.33	-57.07	1976.61	1.830	19.94	58.49	0.9560	5.7742	0.0386	1.7356	0.0287
15	221	5	13.33	15	3.8	2.4	4.13	-47.32	2424.55	1.847	24.43	54.75	0.9314	5.5575	0.0424	1.8406	0.0229
16	234	5.33	19.33	13.67	5.4	2.2	3.6	-60.61	1811.81	1.900	18.29	58.60	0.9857	5.7742	0.0386	1.9138	0.0195
17	248	6.67	10.67	15	5.93	3.6	3.33	-49.90	2304.95	1.886	23.22	57.19	0.9151	5.7742	0.0386	1.8480	0.0213
18	262	9.67	19.33	7	7	3.6	5.2	-64.16	1648.65	1.900	16.65	61.26	0.9774	5.7250	0.0394	1.9588	0.0099
19	278	7.67	13.33	11	4.87	2.4	4.13	-48.90	2351.47	1.862	23.70	54.97	0.9454	5.5428	0.0427	1.8756	0.0175
20	291	8.33	17.33	7.67	5.13	2	3.6	-54.25	2106.35	1.820	21.24	55.81	0.9601	5.6768	0.0400	1.7471	0.0268
21	304	6.33	10.67	7.67	4.33	2.8	2	-36.53	2921.53	1.825	29.42	48.99	0.8944	5.4975	0.0538	1.8178	0.0260
22	317	5	20	13	6.73	4	2	-64.35	1640.25	1.880	16.57	61.04	0.9869	5.7742	0.0386	1.8358	0.0231
23	331	5.67	13.33	5	6.73	2.8	2.27	-48.09	2387.88	1.900	24.07	53.12	0.9597	5.7742	0.0386	1.9242	0.0132
24	343	6.33	16	15	7	2.6	4.13	-59.99	1840.26	1.900	18.57	61.00	0.9495	5.7742	0.0386	1.9573	0.0101
25	356	9.33	19.33	13.67	4.07	3.4	2.27	-61.19	1786.88	1.836	18.04	58.85	0.9937	5.7742	0.0386	1.7476	0.0278
26	370	8.67	10	10.33	3.8	4	3.07	-41.67	2685.98	1.791	27.05	52.91	0.8910	5.7473	0.0391	1.6666	0.0332
27	383	7.33	10.67	9	4.33	2.8	4.93	-41.11	2709.06	1.900	27.28	52.70	0.8798	5.7742	0.0386	1.9104	0.0155
28	398	7	20	13.67	5.13	1.2	3.07	-59.83	1847.82	1.900	18.65	58.25	0.9934	5.7742	0.0386	1.9096	0.0176
29	413	7.67	14	9	3.53	4	2.8	-47.72	2405.06	1.900	24.23	54.59	0.9470	5.7742	0.0386	1.8883	0.0178
30	425	9.33	20	10.33	3.8	3.8	2	-59.54	1861.65	1.887	18.79	58.04	0.9960	5.7742	0.0386	1.8496	0.0238
31	436	7	13.33	11	3.27	2.8	5.47	-46.85	2444.93	1.900	24.63	54.77	0.9485	5.7742	0.0386	1.9138	0.0158
32	449	9.67	10.67	9	3.53	2.2	2.8	-35.97	2945.39	1.900	29.65	50.75	0.8983	5.7742	0.0386	1.8809	0.0185
33	463	7.67	14	6.33	3.53	4	2.8	-46.02	2483.24	1.900	25.02	53.07	0.9445	5.7742	0.0386	1.8883	0.0178
34	474	7.33	10.67	6.33	4.33	2.8	4.93	-39.15	2799.21	1.900	28.19	51.09	0.8753	5.7742	0.0386	1.9104	0.0155
35	487	6.33	16	7.67	4.33	2.8	2	-50.04	2298.10	1.900	23.17	53.25	0.9706	5.7742	0.0386	1.9423	0.0143
36	499	6.33	10.67	15	7	2.6	4.13	-50.89	2258.84	1.900	22.76	57.82	0.9349	5.7742	0.0386	1.9586	0.0100
37	514	9.33	18	7	4.6	2.4	6	-56.03	2022.61	1.900	20.40	57.48	0.9771	5.7742	0.0386	1.9193	0.0169
38	529	6.33	12	11.67	4.33	3.8	4.93	-48.44	2375.88	1.723	23.94	55.98	0.9112	5.7742	0.0386	1.5214	0.0346
39	541	7	17.33	11	5.67	1	2.27	-53.78	2126.00	1.900	21.44	56.46	0.9704	5.7742	0.0386	1.9878	0.0055
40	554	6.33	14.67	14.33	4.07	1.8	5.47	-51.13	2250.19	1.810	22.68	56.78	0.9558	5.6341	0.0404	1.7419	0.0278
41	567	6.33	14.67	14.33	5.67	2.8	5.47	-56.42	2004.60	1.900	20.21	59.25	0.9569	5.6966	0.0398	1.9117	0.0161
42	579	9.67	10.67	9	3.27	1	2.27	-30.20	3211.99	1.858	32.33	48.27	0.8742	5.7191	0.0395	1.8089	0.0242
43	591	7.33	10.67	9	4.33	2.8	4.13	-40.13	2756.19	1.808	27.75	52.44	0.8894	5.7742	0.0386	1.6904	0.0314
44	604	6.33	10.67	7.67	4.33	2.8	2.8	-36.36	2928.57	1.853	29.48	50.36	0.8983	5.6550	0.0402	1.8219	0.0215
45	617	6.33	10.67	9	4.33	2.8	4.93	-39.64	2778.59	1.817	27.97	52.94	0.8655	5.7742	0.0386	1.7094	0.0286

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
46	630	7.33	14.67	14.33	5.67	1	2.27	-50.83	2261.85	1.897	22.80	55.60	0.9805	5.7373	0.0392	1.8808	0.0188
47	644	7	10.67	7.67	4.33	2.8	2	-36.99	2902.30	1.750	29.22	49.53	0.8807	5.4805	0.0449	1.6723	0.0304
48	659	5.67	14.67	10.33	4.6	3.4	6	-53.64	2132.70	1.900	21.50	57.15	0.9358	5.7742	0.0386	1.8994	0.0208
49	673	6.33	10.67	9	4.87	3.6	4.93	-44.93	2533.63	1.879	25.52	54.17	0.9126	5.7742	0.0386	1.8325	0.0212
50	689	7	13.33	7.67	4.33	2.8	2	-43.74	2590.40	1.804	26.10	52.35	0.9343	5.7742	0.0386	1.6838	0.0263
51	700	6.33	10.67	9	3.53	4	2.8	-39.83	2771.56	1.743	27.91	51.45	0.8906	5.7742	0.0386	1.5607	0.0348
52	713	7.67	14	6.33	4.33	2.8	4.93	-47.37	2422.46	1.848	24.41	54.02	0.9460	5.7742	0.0386	1.7705	0.0280
53	726	6.33	10.67	9	4.33	3.6	2	-39.55	2782.87	1.806	28.02	51.98	0.9109	5.6797	0.0400	1.7194	0.0303
54	738	6.33	10.67	7.67	4.33	2.8	4.93	-38.64	2824.63	1.817	28.44	52.21	0.8668	5.7742	0.0386	1.7094	0.0286
55	751	8.33	10.67	9	3.53	2.2	2.8	-35.40	2975.69	1.734	29.96	50.11	0.8736	5.6976	0.0398	1.5688	0.0329
56	764	9	14	6.33	3.53	4	2.8	-47.78	2402.78	1.873	24.22	53.37	0.9535	5.7742	0.0386	1.8206	0.0276
57	777	7	10.67	9	3.27	1	2.27	-28.04	3311.44	1.856	33.33	47.00	0.8831	5.6039	0.0406	1.8439	0.0204
58	790	9.67	10.67	7.67	3.27	2.8	2	-35.37	2976.55	1.753	29.97	49.95	0.9136	5.7095	0.0396	1.6029	0.0325
59	803	7	17.33	10.33	4.33	2.8	4.13	-55.30	2056.63	1.886	20.74	56.71	0.9643	5.7244	0.0394	1.8642	0.0201
60	817	7.33	10.67	15	5.67	1	2.27	-42.04	2666.64	1.871	26.85	53.20	0.9291	5.7742	0.0386	1.8182	0.0214
61	832	7	10.67	7.67	4.33	2.8	3.33	-37.81	2861.25	1.883	28.81	51.10	0.8955	5.7468	0.0391	1.8504	0.0211
62	847	6.67	10.67	15	5.93	3.6	2	-49.25	2334.32	1.900	23.52	56.33	0.9280	5.7742	0.0386	1.9231	0.0161
63	860	5.67	13.33	11	4.87	2.4	4.13	-47.91	2396.70	1.885	24.15	54.53	0.9343	5.7742	0.0386	1.8444	0.0215
64	873	5	14.67	10.33	4.6	3.4	6	-52.64	2179.77	1.844	21.97	56.64	0.9442	5.7368	0.0392	1.7749	0.0294
65	886	7.33	10.67	7.67	4.33	2.8	2.8	-37.24	2887.89	1.854	29.08	50.54	0.8975	5.7564	0.0389	1.7884	0.0289
66	898	6.33	10.67	6.33	4.33	2.8	2.8	-35.34	2975.47	1.853	29.96	49.56	0.8947	5.6550	0.0402	1.8219	0.0215
67	911	6.67	10.67	7.67	3.27	2.8	2	-32.36	3111.22	1.900	31.32	48.67	0.9051	5.7742	0.0386	1.8743	0.0212
68	924	9.67	10.67	15	5.93	3.6	2	-50.69	2269.00	1.872	22.86	57.73	0.9258	5.7681	0.0387	1.8218	0.0191
69	937	7	10.67	6.33	4.33	2.8	2.8	-35.92	2948.54	1.872	29.69	49.37	0.8833	5.6993	0.0399	1.8434	0.0252
70	949	6.33	10.67	7.67	4.33	2.8	3.33	-37.40	2880.61	1.867	29.00	50.86	0.8905	5.7742	0.0386	1.8094	0.0266
71	963	7.33	10.67	7	4.33	2.8	4.93	-39.73	2772.51	1.900	27.92	51.50	0.8764	5.7742	0.0386	1.9104	0.0155
72	978	7.33	10.67	7.67	4.87	2.8	4.93	-41.25	2702.64	1.900	27.22	52.88	0.9013	5.7742	0.0386	1.8854	0.0189
73	990	7.33	10.67	6.33	4.33	2.8	2.8	-36.29	2932.01	1.854	29.52	49.71	0.8967	5.7564	0.0389	1.7884	0.0289
74	1003	6.33	10.67	15	7	3	4.13	-50.97	2257.01	1.834	22.74	57.88	0.9394	5.7742	0.0386	1.7423	0.0297
75	1015	8.67	10	5	3.8	4	3.07	-37.68	2869.14	1.791	28.89	49.96	0.8897	5.7473	0.0391	1.6666	0.0332
76	1028	6.33	10.67	6.33	4.87	3.6	4.93	-42.85	2629.40	1.879	26.48	52.64	0.9051	5.7742	0.0386	1.8325	0.0212
77	1041	6.33	10.67	9	4.33	2.8	2.8	-37.52	2875.15	1.853	28.95	51.11	0.9039	5.6550	0.0402	1.8219	0.0215
78	1053	7.33	10.67	7.67	3.27	2.8	2	-33.48	3061.97	1.806	30.83	48.26	0.9013	5.6849	0.0399	1.7169	0.0288
79	1066	9.67	10.67	9	4.33	2.8	4.13	-42.09	2664.68	1.875	26.84	52.89	0.8998	5.7314	0.0393	1.8403	0.0221
80	1078	9.67	10.67	9	3.53	2.2	2.53	-35.23	2979.51	1.900	29.99	50.39	0.8871	5.7742	0.0386	1.9215	0.0189
81	1091	7.33	10.67	7.67	4.87	1	2.27	-32.62	3099.42	1.900	31.20	49.22	0.8957	5.7742	0.0386	1.8848	0.0202
82	1106	7	10.67	9	3.27	2.8	4.93	-37.32	2889.11	1.655	29.09	51.31	0.8902	5.6696	0.0401	1.4202	0.0353
83	1120	6.33	10.67	7.67	3.53	2.2	2.8	-31.73	3144.48	1.732	31.65	48.48	0.8757	5.6122	0.0406	1.5942	0.0322
84	1135	9.67	10.67	9	4.33	2.8	3.33	-41.53	2691.49	1.826	27.10	52.79	0.9184	5.7742	0.0386	1.7265	0.0305
85	1148	7.33	10.67	7.67	4.07	2.2	2.8	-35.19	2982.85	1.826	30.03	49.49	0.8731	5.7742	0.0386	1.7277	0.0268
86	1162	9.67	10.67	9	3.8	2.8	2.8	-39.89	2765.16	1.900	27.85	51.66	0.9122	5.7742	0.0386	1.8878	0.0180
87	1174	7	10.67	9	3.27	2.8	2.8	-35.44	2970.75	1.865	29.91	49.34	0.8771	5.7742	0.0386	1.8062	0.0262
88	1186	7.33	10.67	6.33	4.33	1	2.27	-29.98	3221.71	1.872	32.43	47.20	0.8982	5.7742	0.0386	1.8189	0.0226
89	1199	7.33	10.67	7.67	3.27	3.6	4.93	-39.52	2786.34	1.727	28.06	51.36	0.8951	5.6969	0.0398	1.5548	0.0328
90	1211	6.33	10.67	6.33	4.87	2.8	2	-37.55	2874.85	1.812	28.95	50.35	0.8835	5.7742	0.0386	1.7000	0.0347

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
91	1223	9	10.67	7.67	4.33	2.8	2.8	-38.95	2810.55	1.815	28.30	51.70	0.8957	5.7742	0.0386	1.7043	0.0277
92	1235	7.33	10.67	9	3.53	2.2	2.53	-34.03	3039.32	1.708	30.60	49.45	0.8751	5.4904	0.0449	1.5854	0.0330
93	1248	6.67	10.67	7.67	3.27	2.8	4.93	-35.88	2950.04	1.873	29.70	50.82	0.8824	5.7742	0.0386	1.8215	0.0216
94	1261	7.33	10.67	6.33	4.33	2.8	2	-36.90	2906.85	1.724	29.27	49.34	0.9089	5.7742	0.0386	1.5224	0.0342
95	1275	7.33	13.33	7.67	3.27	2.8	2	-41.53	2689.75	1.897	27.10	50.62	0.9311	5.7742	0.0386	1.8691	0.0196
96	1290	6.67	10.67	7.67	4.33	2.8	2.8	-37.10	2893.39	1.900	29.13	50.40	0.9024	5.7742	0.0386	1.9334	0.0125
97	1301	9.67	10.67	6.33	4.87	2.8	2	-39.84	2767.93	1.884	27.87	51.63	0.9251	5.7742	0.0386	1.8425	0.0251
98	1315	6.33	10.67	9	3.53	2.2	2.8	-33.05	3083.69	1.732	31.04	49.27	0.8817	5.6122	0.0406	1.5942	0.0322
99	1328	7.33	10.67	6.33	4.07	2.2	2.8	-34.17	3029.74	1.826	30.50	48.68	0.8720	5.7742	0.0386	1.7277	0.0268
100	1339	9.67	10.67	9	3.8	1	2.27	-32.57	3104.22	1.806	31.24	49.64	0.8993	5.7742	0.0386	1.6878	0.0321
101	1353	7	10.67	9	3.53	2.2	2.8	-33.79	3045.91	1.895	30.66	49.72	0.8744	5.7628	0.0388	1.8689	0.0192
102	1368	7.33	10.67	7	4.07	2.2	2.8	-34.65	3007.70	1.826	30.28	49.11	0.8735	5.7742	0.0386	1.7277	0.0268
103	1381	9.67	10.67	9	3.8	2.8	4.93	-40.38	2748.13	1.665	27.67	53.22	0.9049	5.7742	0.0386	1.4043	0.0374
104	1393	9.67	12	9	3.53	2.2	2.53	-39.65	2778.32	1.816	27.97	52.18	0.9250	5.7742	0.0386	1.7077	0.0297
105	1406	9	10.67	7.67	4.33	3.2	2.8	-40.69	2730.21	1.826	27.49	52.30	0.9064	5.7617	0.0388	1.7324	0.0299
106	1418	7.33	10.67	7.67	3.53	2.2	2.53	-32.65	3102.95	1.708	31.23	48.81	0.8765	5.4904	0.0449	1.5854	0.0330
107	1429	7.33	10.67	9	3.27	2.8	3.07	-35.38	2975.77	1.761	29.96	50.30	0.8963	5.6704	0.0401	1.6310	0.0318
108	1444	6.67	10.67	7.67	3	2.8	4.93	-35.53	2965.41	1.900	29.85	49.87	0.8859	5.7742	0.0386	1.9469	0.0143
109	1457	10	10.67	9	3.53	2.2	2.8	-35.66	2960.07	1.890	29.80	50.42	0.9034	5.7242	0.0394	1.8716	0.0180
110	1470	7	10.67	9	3.27	1.4	2.27	-29.48	3244.09	1.900	32.65	47.56	0.8700	5.7742	0.0386	1.9592	0.0099
111	1483	6	10.67	6.33	4.33	2.8	4.93	-38.23	2842.61	1.842	28.62	50.79	0.8729	5.6737	0.0417	1.7936	0.0268
112	1495	7.33	13.33	7.67	4.87	2.8	4.93	-47.99	2392.93	1.874	24.11	54.34	0.9266	5.7742	0.0386	1.8241	0.0236
113	1508	7	10	9	3.27	1	2.27	-25.85	3413.05	1.808	34.35	46.46	0.8595	5.6879	0.0405	1.7208	0.0265
114	1522	7.33	10.67	6.33	4.33	1	2	-29.28	3257.06	1.743	32.78	47.56	0.8948	5.7742	0.0386	1.5607	0.0351
115	1538	9	10.67	7.67	4.33	1.6	2.8	-34.86	2998.00	1.834	30.18	50.32	0.9103	5.7742	0.0386	1.7437	0.0296
116	1550	5	10.67	9	3.53	2.2	4.13	-33.68	3053.05	1.800	30.73	49.06	0.8740	5.7604	0.0388	1.6790	0.0311
117	1562	9.67	10.67	9	4.33	2.8	2.8	-40.72	2730.28	1.768	27.49	52.51	0.9195	5.7742	0.0386	1.6115	0.0328
118	1575	7.33	10.67	6.33	3.53	2.2	2.53	-31.42	3159.11	1.708	31.80	48.05	0.8768	5.4904	0.0449	1.5854	0.0330
119	1588	7.33	10.67	7.67	4.33	1	2.27	-31.14	3168.35	1.872	31.89	48.00	0.9014	5.7742	0.0386	1.8189	0.0226
120	1601	6	10.67	9	3.27	2.8	3.07	-34.29	3023.59	1.867	30.44	49.47	0.8718	5.5827	0.0448	1.8729	0.0209
121	1616	7.33	10.67	9	3.27	2.6	3.07	-34.90	2995.81	1.854	30.16	49.47	0.8865	5.7742	0.0386	1.7834	0.0259
122	1627	7	10.67	6.33	4.07	2.2	2.8	-33.07	3079.82	1.857	31.00	48.87	0.8685	5.7742	0.0386	1.7894	0.0281
123	1640	7.33	10.67	9	3.27	1	2.27	-28.72	3283.86	1.703	33.05	47.45	0.8786	5.6613	0.0402	1.5184	0.0328
124	1654	7.33	10.67	9	3.53	2.2	2.8	-34.29	3022.77	1.890	30.43	49.75	0.8921	5.7742	0.0386	1.8562	0.0250
125	1667	7	10	9	3.53	2.2	2.53	-30.56	3195.31	1.857	32.16	48.24	0.8840	5.6866	0.0399	1.8193	0.0246
126	1682	6.67	10.67	7.67	3.53	2.2	2.53	-31.84	3137.23	1.819	31.58	48.85	0.8876	5.7051	0.0397	1.7371	0.0265
127	1697	7.33	10.67	5	3	2.8	4.93	-32.94	3087.16	1.803	31.08	48.70	0.8711	5.7576	0.0389	1.6874	0.0293
128	1710	6.67	10.67	9	4.33	1	2.27	-31.20	3166.74	1.816	31.87	48.56	0.8972	5.6703	0.0401	1.7416	0.0260
129	1722	7	10.67	9	3.27	1	2.8	-28.11	3307.47	1.871	33.28	48.50	0.8720	5.6079	0.0440	1.8725	0.0190
130	1734	7.33	10.67	7	3.53	2.2	2.53	-32.00	3132.43	1.708	31.53	48.44	0.8769	5.4904	0.0449	1.5854	0.0330
131	1747	6	10.67	9	3.8	3.2	3.07	-36.80	2907.41	1.900	29.27	50.74	0.8871	5.7445	0.0391	1.9172	0.0167
132	1759	7	10	9	3	1	2.53	-24.97	3453.01	1.823	34.74	46.77	0.8814	5.6502	0.0405	1.7631	0.0269
133	1771	7.33	10.67	6.33	3.53	2.2	2.27	-31.41	3160.84	1.664	31.82	47.77	0.8706	5.6756	0.0400	1.4353	0.0349
134	1784	7	10	9	3.27	1.8	2.27	-28.47	3294.13	1.750	33.15	47.63	0.8630	5.6982	0.0398	1.6008	0.0303
135	1796	7.33	10.67	6.33	3.27	1.4	2.27	-27.67	3328.57	1.851	33.50	46.00	0.8686	5.6213	0.0442	1.8276	0.0189

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
136	1809	7	10.67	9	4.33	1	2.27	-31.67	3143.67	1.875	31.64	48.75	0.8933	5.7438	0.0391	1.8355	0.0249
137	1820	6.67	10.67	7.67	3.53	2.2	2.27	-31.94	3132.24	1.848	31.53	48.00	0.8818	5.7742	0.0386	1.7719	0.0245
138	1835	7	10.67	9	3.27	1.4	2.53	-29.52	3243.06	1.858	32.64	47.33	0.8862	5.6076	0.0417	1.8460	0.0200
139	1850	7	10	9	3.27	1.4	2.53	-27.37	3342.88	1.824	33.64	46.90	0.8668	5.7742	0.0386	1.7241	0.0271
140	1866	6.67	15.33	9	3.27	1.8	2.27	-43.97	2578.15	1.861	25.97	51.78	0.9632	5.7742	0.0386	1.7981	0.0252
141	1879	7.33	10.67	6.33	4.33	1.8	2.27	-32.95	3087.98	1.745	31.09	48.44	0.8656	5.5498	0.0436	1.6405	0.0329
142	1892	7	10	9	3	1	2.27	-23.95	3500.79	1.799	35.23	45.96	0.8680	5.6494	0.0403	1.7150	0.0261
143	1904	7	10	9	3.27	1	2.53	-26.25	3394.70	1.807	34.16	46.82	0.8559	5.7103	0.0396	1.7104	0.0286
144	1917	7	10	9	3.8	1	2.27	-27.29	3345.90	1.855	33.67	47.44	0.8715	5.7637	0.0388	1.7881	0.0303
145	1929	7	15.33	9	3	1	2.53	-42.19	2660.11	1.867	26.80	50.58	0.9478	5.7196	0.0395	1.8282	0.0227
146	1941	6.33	10.67	9	3.27	1	2.8	-28.10	3307.88	1.882	33.29	47.33	0.8877	5.7457	0.0391	1.8482	0.0186
147	1954	7.33	10.67	7.67	3.53	2.2	2.27	-32.61	3105.63	1.664	31.26	48.57	0.8716	5.6756	0.0400	1.4353	0.0349
148	1967	6	10.67	6.33	4.33	1	2	-28.13	3307.69	1.834	33.29	46.91	0.8679	5.6486	0.0412	1.7850	0.0227
149	1980	7	10.67	9	3.27	2.2	2.27	-32.40	3110.34	1.875	31.31	48.77	0.8955	5.7742	0.0386	1.8247	0.0263
150	1994	7.33	10	6.33	3.27	1.4	2.27	-24.52	3472.83	1.873	34.95	45.58	0.8517	5.5943	0.0418	1.8818	0.0197
151	2008	7.33	10.67	9	3	1	2.53	-27.12	3352.34	1.900	33.73	47.35	0.8732	5.6903	0.0399	1.9192	0.0178
152	2023	7	10	6.33	4.33	1	2.27	-26.91	3363.16	1.849	33.85	46.71	0.8760	5.7548	0.0389	1.7801	0.0264
153	2035	7.33	10.67	6.33	3.8	1.4	2.27	-28.52	3288.35	1.881	33.10	47.13	0.8746	5.7601	0.0389	1.8427	0.0218
154	2046	7.33	10.67	6.33	4.07	1.4	2.27	-30.08	3219.89	1.748	32.41	47.27	0.8937	5.5164	0.0483	1.6573	0.0257
155	2058	7.33	10	9	3	1	2.53	-26.29	3402.03	1.430	34.23	46.64	0.8766	5.6795	0.0405	0.9669	0.0419
156	2072	6.33	10.67	9	3.27	2.6	3.07	-34.15	3029.32	1.900	30.49	49.91	0.8928	5.7742	0.0386	1.8777	0.0227
157	2085	7.33	10.67	6.33	4.33	1.4	2	-30.93	3179.25	1.814	32.00	47.52	0.8928	5.7513	0.0390	1.7119	0.0284
158	2099	7	10	7.67	3.8	1	2.27	-25.78	3415.04	1.855	34.36	46.75	0.8703	5.7637	0.0388	1.7881	0.0303
159	2110	7	15.33	9	3.27	1.8	2.27	-45.52	2506.82	1.870	25.26	51.71	0.9603	5.7742	0.0386	1.8151	0.0258
160	2122	7	10.67	9	3.8	1.4	2.27	-31.34	3158.29	1.895	31.79	48.28	0.8976	5.7742	0.0386	1.8660	0.0198
161	2134	9.67	10	9	3.27	1	2.27	-28.00	3315.39	1.767	33.36	47.55	0.8864	5.7742	0.0386	1.6101	0.0302
162	2146	7.33	10	9	3.27	1	2.53	-26.71	3373.49	1.810	33.95	47.03	0.8760	5.7742	0.0386	1.6954	0.0291
163	2159	7	10	6.33	3.27	1.4	2.27	-24.74	3462.76	1.868	34.85	45.84	0.8581	5.7742	0.0386	1.8120	0.0226
164	2173	7	10.67	6.33	3.8	1.4	2.27	-28.78	3276.17	1.895	32.98	46.82	0.8955	5.7742	0.0386	1.8660	0.0198
165	2188	7.33	10	6.33	4.33	1	2.27	-27.46	3337.35	1.876	33.59	46.97	0.8558	5.7141	0.0398	1.8477	0.0230
166	2202	7	10	9	3.8	1.4	2.8	-29.22	3257.40	1.843	32.78	48.35	0.8738	5.6860	0.0399	1.7906	0.0227
167	2215	6.67	10.67	6.33	3	1	2.27	-24.58	3471.58	1.806	34.94	44.93	0.8887	5.6613	0.0402	1.7246	0.0259
168	2227	7	10.67	9	3	1	2.53	-27.76	3328.12	1.698	33.49	47.02	0.8860	5.6310	0.0445	1.5181	0.0383
169	2238	7.33	12.67	9	4.33	1	2.27	-38.71	2822.98	1.752	28.43	50.32	0.9138	5.5763	0.0456	1.6456	0.0307
170	2250	7.33	10.67	6.33	4.33	1	2.8	-30.74	3186.51	1.879	32.08	47.18	0.8893	5.7742	0.0386	1.8327	0.0236
171	2262	5.67	10.67	9	3.27	1.8	2.27	-29.56	3240.35	1.900	32.61	47.47	0.8686	5.7742	0.0386	1.9019	0.0190
172	2274	7.33	10.67	9	3.8	1.4	2.27	-31.23	3164.03	1.881	31.85	48.73	0.8850	5.7601	0.0389	1.8427	0.0218
173	2286	7.33	10.67	6.33	3.27	1	2.27	-26.17	3400.76	1.703	34.23	45.92	0.8706	5.6613	0.0402	1.5184	0.0328
174	2298	7	10	9	3	1	2	-24.52	3483.10	1.451	35.05	45.77	0.8739	5.6085	0.0410	1.0324	0.0443
175	2311	9.67	10	9	3.27	1.4	2.53	-30.62	3195.53	1.739	32.16	48.76	0.8893	5.7457	0.0391	1.5632	0.0343
176	2326	7	10	7.67	3.53	1	2.27	-25.73	3418.32	1.825	34.40	45.84	0.8587	5.6879	0.0407	1.7541	0.0278
177	2342	7.33	10	9	3	1	2.27	-25.09	3450.08	1.733	34.72	46.43	0.8535	5.6724	0.0401	1.5752	0.0347
178	2353	7	10	6.33	4.07	1	2.27	-26.09	3400.03	1.900	34.22	46.33	0.8696	5.7513	0.0390	1.9159	0.0139
179	2366	7	10.67	9	3.27	1	2.53	-28.02	3312.55	1.836	33.34	47.42	0.8790	5.7742	0.0386	1.7477	0.0263
180	2379	7	10	9	3.8	1	2.53	-27.93	3315.33	1.894	33.36	47.42	0.8618	5.7742	0.0386	1.8630	0.0227

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
181	2392	7.33	10	9	3.27	1.4	2.8	-28.29	3300.44	1.830	33.22	47.24	0.8687	5.7742	0.0386	1.7357	0.0299
182	2404	7.33	10.67	6.33	3	1	2.27	-24.74	3463.37	1.846	34.86	45.17	0.8490	5.7710	0.0387	1.7683	0.0242
183	2416	7	10	6.33	4.33	1	2	-26.10	3399.91	1.872	34.22	46.07	0.8582	5.7742	0.0386	1.8203	0.0264
184	2427	7	10	6.33	3.27	1	2.27	-23.47	3522.46	1.808	35.45	44.82	0.8499	5.6879	0.0405	1.7208	0.0265
185	2441	6.67	10.67	6.33	3.8	1	2.27	-27.33	3344.38	1.826	33.66	46.14	0.8561	5.7742	0.0386	1.7279	0.0280
186	2453	6	10.67	6.33	3	1	2.27	-23.28	3529.70	1.881	35.52	44.49	0.8497	5.7742	0.0386	1.8365	0.0189
187	2466	7.33	10	9	3.27	1	2	-24.90	3458.43	1.746	34.80	46.46	0.8525	5.6431	0.0424	1.6109	0.0362
188	2479	7.33	10	9	3.27	1.4	2.27	-26.98	3359.71	1.873	33.81	47.26	0.8642	5.5943	0.0418	1.8818	0.0197
189	2491	9.67	10	6.33	3.27	1	2.53	-26.56	3383.16	1.704	34.05	45.90	0.8610	5.5026	0.0474	1.5732	0.0353
190	2505	10	10	9	3	1	2.27	-27.05	3357.16	1.834	33.78	47.77	0.8830	5.7639	0.0388	1.7465	0.0299
191	2520	5.67	10	9	3	1.8	2.53	-26.41	3385.40	1.888	34.07	46.74	0.8751	5.7364	0.0392	1.8644	0.0224
192	2535	7	10	9	3.27	1	2	-25.35	3436.04	1.806	34.58	46.10	0.8617	5.7742	0.0386	1.6875	0.0296
193	2547	7	10	7.67	3.53	1.8	2.27	-28.23	3304.24	1.787	33.25	47.47	0.8808	5.5985	0.0451	1.7084	0.0277
194	2559	7	12.67	6.33	4.07	1	2	-34.78	3000.81	1.876	30.22	48.16	0.9260	5.7742	0.0386	1.8271	0.0243
195	2574	7.33	10	6.33	3.27	1	2	-22.32	3577.09	1.746	35.99	44.81	0.8399	5.6431	0.0424	1.6109	0.0362
196	2586	7	10	9	3	1	4.13	-27.13	3355.86	1.734	33.77	46.95	0.8612	5.6933	0.0414	1.5707	0.0337
197	2597	8.67	10.67	6.33	3	1	2.27	-26.04	3404.41	1.804	34.26	45.80	0.8535	5.7742	0.0386	1.6833	0.0320
198	2611	10	10	9	3	1.2	2.27	-28.18	3305.06	1.836	33.26	47.23	0.9056	5.6928	0.0398	1.7738	0.0212
199	2624	10	10	6.33	3.27	1	2	-26.41	3389.70	1.705	34.11	46.02	0.8630	5.6693	0.0401	1.5195	0.0318
200	2636	7	10	14.33	3.27	1	2.27	-31.36	3159.50	1.808	31.80	48.97	0.8743	5.6879	0.0405	1.7208	0.0265
201	2649	7.33	10	13	3.53	1	2.27	-31.48	3152.16	1.894	31.73	49.12	0.9038	5.7742	0.0386	1.8631	0.0242
202	2663	7.33	10.67	5	3.53	1.8	2.27	-28.91	3273.26	1.769	32.95	46.54	0.8679	5.7055	0.0401	1.6369	0.0298
203	2675	7	10	6.33	3.27	1.8	2	-25.55	3429.68	1.700	34.51	46.37	0.8684	5.7742	0.0386	1.4749	0.0385
204	2689	7	10	9	3.27	1.4	2.27	-26.96	3360.81	1.868	33.82	47.66	0.8789	5.7742	0.0386	1.8120	0.0226
205	2703	7.33	10	9	3	1	4.13	-27.15	3356.85	1.666	33.78	47.08	0.8570	5.6626	0.0413	1.4454	0.0403
206	2715	7	10	9	5.13	1	2.27	-32.28	3116.91	1.830	31.37	48.99	0.8789	5.7742	0.0386	1.7357	0.0265
207	2728	7.33	10	6.33	3.27	1	2.27	-23.35	3529.99	1.724	35.52	45.06	0.8523	5.3935	0.0522	1.6511	0.0306
208	2742	10	10	6.33	3.27	1	2.27	-25.93	3412.09	1.695	34.34	46.37	0.8727	5.7619	0.0388	1.4702	0.0376
209	2754	7.33	10	9	3	1	2	-25.46	3430.44	1.840	34.52	46.02	0.8572	5.7742	0.0386	1.7553	0.0265
210	2766	7.33	10	9	3	1.4	4.13	-29.63	3237.88	1.870	32.59	48.18	0.8853	5.7499	0.0390	1.8238	0.0245
211	2779	7	10	9	3	1.4	2.27	-26.08	3400.21	1.900	34.22	46.73	0.8701	5.7742	0.0386	1.9131	0.0179
212	2792	7.33	10	9	3.27	1	2.27	-26.05	3405.73	1.724	34.27	46.64	0.8629	5.3935	0.0522	1.6511	0.0306
213	2807	7	10	9	3	1	3.33	-25.60	3426.21	1.738	34.48	46.89	0.8525	5.7125	0.0408	1.5722	0.0349
214	2821	7.33	10.67	6.33	3	1.8	2	-27.15	3352.68	1.841	33.74	46.07	0.8761	5.4083	0.0467	1.8783	0.0212
215	2834	10	10	6.33	3.53	1	2.27	-26.91	3362.02	1.896	33.84	46.23	0.8802	5.7742	0.0386	1.8682	0.0213
216	2848	6.33	10	9	3	1	2.27	-24.70	3468.20	1.714	34.90	45.57	0.8735	5.4868	0.0459	1.5992	0.0348
217	2863	7.33	10	5	3.27	1.4	2.27	-23.50	3519.60	1.873	35.42	44.65	0.8476	5.5943	0.0418	1.8818	0.0197
218	2878	7	10	9	3	1.4	3.33	-27.10	3354.33	1.870	33.75	47.66	0.8663	5.7501	0.0390	1.8231	0.0231
219	2890	7.33	10	6.33	3.27	1.4	2	-24.80	3460.58	1.850	34.83	45.14	0.8737	5.7476	0.0391	1.7846	0.0268
220	2903	10	10	6.33	3	1	2.27	-24.59	3470.28	1.834	34.92	46.07	0.8713	5.7639	0.0388	1.7465	0.0299
221	2918	6.33	10	9	3.27	1	2	-24.53	3472.87	1.842	34.95	45.66	0.8603	5.7170	0.0395	1.7789	0.0264
222	2932	7.33	10	9	3	1.4	2.27	-27.46	3337.42	1.869	33.59	47.12	0.8561	5.7742	0.0386	1.8127	0.0237
223	2944	7.33	10	7	3.27	1	2	-22.98	3546.73	1.746	35.69	45.20	0.8400	5.6431	0.0424	1.6109	0.0362
224	2957	7.33	10.67	6.33	3.27	1	2	-25.51	3426.36	1.900	34.48	45.23	0.8738	5.7742	0.0386	1.9307	0.0137
225	2970	7.33	10	6.33	3.27	1	3.07	-24.84	3458.86	1.833	34.81	45.42	0.8666	5.6510	0.0404	1.7822	0.0232

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
226	2984	7.33	10	9	3	1	3.33	-25.93	3412.12	1.704	34.34	46.43	0.8535	5.7742	0.0386	1.4827	0.0371
227	2998	6.33	10	6.33	3.27	1	2.27	-22.34	3576.74	1.725	35.99	44.12	0.8509	5.6458	0.0403	1.5684	0.0319
228	3012	7	10	6.33	3.27	1	2	-22.33	3575.17	1.806	35.98	44.67	0.8567	5.7742	0.0386	1.6875	0.0296
229	3023	7.33	12.67	6.33	3.27	1.4	2.27	-34.18	3028.65	1.857	30.49	48.29	0.9224	5.6549	0.0402	1.8288	0.0238
230	3037	10	10	6.33	3.27	1	3.07	-27.17	3351.28	1.849	33.73	47.04	0.8761	5.7742	0.0386	1.7730	0.0246
231	3051	7.33	10	6.33	3	1	3.07	-23.82	3512.53	1.561	35.35	45.34	0.8641	5.4796	0.0449	1.2961	0.0375
232	3063	7.33	10	6.33	3	1	2.27	-22.95	3548.31	1.733	35.71	44.66	0.8361	5.6724	0.0401	1.5752	0.0347
233	3076	7.33	10	6.33	3.27	1.4	4.4	-27.58	3338.03	1.630	33.60	46.43	0.8493	5.7654	0.0388	1.3382	0.0377
234	3089	5	10	6.33	3.27	1	2	-20.55	3657.16	1.791	36.80	43.83	0.8446	5.6214	0.0424	1.7080	0.0298
235	3103	9.67	10	6.33	3.27	1	2	-25.27	3439.80	1.809	34.61	46.23	0.8807	5.7742	0.0386	1.6930	0.0288
236	3116	7.33	15.33	9	3	1	3.33	-43.09	2617.81	1.900	26.37	51.35	0.9395	5.6379	0.0437	1.9281	0.0132
237	3128	6	10	6.33	3.27	1.4	2.27	-22.72	3556.97	1.813	35.79	45.32	0.8581	5.7672	0.0387	1.7037	0.0305
238	3140	6.67	10	6.33	3.27	1	2	-22.58	3565.93	1.710	35.88	44.71	0.8419	5.3876	0.0491	1.6237	0.0333
239	3153	5	10.67	6.33	3.27	1	2	-23.04	3542.69	1.790	35.65	44.53	0.8457	5.6664	0.0401	1.6907	0.0309
240	3165	7.33	10	6.33	3.8	1	2.27	-25.79	3415.30	1.837	34.37	45.35	0.8570	5.7529	0.0390	1.7554	0.0275
241	3177	7.33	10	6.33	3	1	2	-22.61	3561.18	1.840	35.84	44.47	0.8504	5.7742	0.0386	1.7553	0.0265
242	3189	5.67	10	9	3	1	2.27	-22.66	3559.64	1.822	35.82	45.09	0.8542	5.7294	0.0393	1.7339	0.0296
243	3202	7.33	10	5	3.27	1	2.27	-22.07	3588.80	1.724	36.11	44.17	0.8500	5.3935	0.0522	1.6511	0.0306
244	3214	9.67	15.33	9	3	1	3.33	-44.33	2564.52	1.750	25.84	52.50	0.9669	5.6500	0.0403	1.6176	0.0256
245	3229	5	10	6.33	3.27	1	2.27	-20.20	3670.85	1.890	36.94	43.45	0.8478	5.6793	0.0401	1.8869	0.0201
246	3241	5.67	10	9	3.27	1.4	2.27	-25.55	3426.95	1.814	34.48	46.47	0.8670	5.6010	0.0412	1.7616	0.0265
247	3252	5	10.67	6.33	3	1	3.07	-24.10	3495.21	1.747	35.18	44.03	0.8604	5.5161	0.0487	1.6562	0.0336
248	3266	5	10.67	6.33	3.27	1	2.27	-23.37	3529.69	1.708	35.52	44.47	0.8563	5.7742	0.0386	1.4915	0.0367
249	3280	5	10	6.33	3.27	1	3.33	-22.23	3582.65	1.691	36.05	44.82	0.8507	5.7742	0.0386	1.4568	0.0353
250	3292	5	10	9	3	1	2.27	-21.75	3599.38	1.900	36.22	44.94	0.8605	5.7742	0.0386	1.8826	0.0166
251	3305	7.33	10	5	3.27	1	3.07	-23.74	3509.36	1.833	35.32	44.54	0.8621	5.6510	0.0404	1.7822	0.0232
252	3317	6.67	10	6.33	4.33	1	4.4	-29.92	3230.42	1.613	32.51	48.06	0.8761	5.7742	0.0386	1.3018	0.0400
253	3329	7.33	10	9	3	1	3.07	-26.18	3404.05	1.561	34.25	47.03	0.8767	5.4796	0.0449	1.2961	0.0375
254	3342	6.33	10	6.33	3	1	2.27	-21.80	3601.71	1.714	36.24	44.05	0.8651	5.4868	0.0459	1.5992	0.0348
255	3354	5.67	10	9	3.27	1	2.27	-24.73	3468.04	1.659	34.90	45.26	0.8632	5.6737	0.0415	1.4275	0.0361
256	3366	5.67	10	7.67	3	1	2.27	-21.52	3611.97	1.822	36.35	44.20	0.8440	5.7294	0.0393	1.7339	0.0296
257	3377	7.33	15.33	6.33	3.27	1	2	-40.60	2734.66	1.813	27.55	49.44	0.9472	5.7742	0.0386	1.7005	0.0306
258	3391	6.67	10	6.33	3.27	1	2.27	-23.68	3512.16	1.844	35.35	44.59	0.8472	5.7742	0.0386	1.7628	0.0277
259	3403	6.33	10	9	3	1	3.07	-24.59	3471.16	1.807	34.93	45.97	0.8751	5.6068	0.0417	1.7456	0.0275
260	3418	7.33	10	11.67	3.27	1	2	-27.24	3350.79	1.746	33.72	48.03	0.8727	5.6431	0.0424	1.6109	0.0362
261	3430	7	10	5	3.27	1	3.07	-22.01	3589.80	1.813	36.12	44.31	0.8474	5.6067	0.0410	1.7565	0.0283
262	3443	7.33	10	9	4.07	1	2.27	-28.36	3295.39	1.900	33.16	48.42	0.8756	5.6889	0.0399	1.9063	0.0172
263	3456	5	10.67	11.67	3.8	1	2.27	-30.88	3182.36	1.776	32.03	48.27	0.8818	5.7437	0.0391	1.6366	0.0340
264	3472	5	10	6.33	3	1	2.27	-19.11	3720.71	1.900	37.44	43.19	0.8446	5.7742	0.0386	1.8826	0.0166
265	3484	5	10	14.33	3.27	1	2.27	-27.73	3324.84	1.890	33.46	47.98	0.8899	5.6793	0.0401	1.8869	0.0201
266	3499	5	10	5	3.27	1	2.27	-18.97	3727.77	1.890	37.51	42.55	0.8436	5.6793	0.0401	1.8869	0.0201
267	3510	5	11.33	6.33	3.27	1	3.07	-26.81	3368.07	1.842	33.90	45.33	0.8680	5.7742	0.0386	1.7598	0.0270
268	3524	7	10	5	3.27	1	3.33	-23.65	3514.11	1.815	35.36	44.79	0.8513	5.6890	0.0399	1.7335	0.0304
269	3537	6.33	15.33	6.33	3.27	1	2.27	-39.40	2788.35	1.874	28.09	49.40	0.9466	5.7742	0.0386	1.8234	0.0226
270	3551	5	10	10.33	3.27	1	2.27	-24.04	3494.33	1.890	35.16	46.01	0.8720	5.6793	0.0401	1.8869	0.0201

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
271	3562	5	10	6.33	3	1	2.8	-20.31	3666.15	1.890	36.89	43.33	0.8295	5.7742	0.0386	1.8562	0.0220
272	3576	6.67	10	6.33	4.33	1	2.27	-27.06	3358.43	1.770	33.80	46.48	0.8723	5.6082	0.0412	1.6703	0.0289
273	3589	6.33	10	6.33	3.27	1	2	-22.24	3578.51	1.842	36.01	43.95	0.8490	5.7170	0.0395	1.7789	0.0264
274	3604	7.33	10	5	4.33	1	2.27	-26.24	3393.44	1.876	34.15	46.10	0.8517	5.7141	0.0398	1.8477	0.0230
275	3618	7.33	10	11.67	5.4	1	2	-35.78	2954.73	1.873	29.74	50.98	0.9026	5.7742	0.0386	1.8211	0.0240
276	3631	7.33	10	5	3.27	1	3.33	-23.75	3511.48	1.727	35.34	45.08	0.8478	5.7742	0.0386	1.5297	0.0381
277	3645	7	10	6.33	3	1	2.27	-21.39	3618.56	1.799	36.41	44.41	0.8622	5.6494	0.0403	1.7150	0.0261
278	3658	7.33	10	6.33	4.07	1	2.27	-25.84	3411.34	1.900	34.33	46.76	0.8652	5.6889	0.0399	1.9063	0.0172
279	3670	5.67	10	7.67	3.53	1	2.27	-23.87	3506.98	1.699	35.29	45.34	0.8446	5.4582	0.0479	1.5795	0.0361
280	3683	6.67	10	5	3.27	1	2.27	-22.72	3556.14	1.844	35.79	43.67	0.8421	5.7742	0.0386	1.7628	0.0277
281	3698	7.33	10	6.33	3.27	1	2.8	-23.69	3510.58	1.881	35.33	45.13	0.8699	5.7348	0.0393	1.8511	0.0210
282	3710	5	10	5	3.27	1	2	-19.18	3720.58	1.791	37.44	42.94	0.8415	5.6214	0.0424	1.7080	0.0298
283	3723	5	10	9	3	1	2	-21.94	3593.82	1.770	36.16	44.93	0.8643	5.5876	0.0409	1.6779	0.0278
284	3735	6.33	10	6.33	3.8	1	2.27	-25.42	3430.75	1.900	34.53	44.75	0.8544	5.7742	0.0386	1.9192	0.0141
285	3747	5	10	9	3	1	2.8	-22.95	3544.76	1.890	35.67	44.97	0.8394	5.7742	0.0386	1.8562	0.0220
286	3759	7	16	6.33	3.27	1	2.8	-42.94	2624.77	1.900	26.45	50.21	0.9592	5.7742	0.0386	1.8895	0.0197
287	3773	7.33	10	10.33	3	1	2.27	-26.53	3383.72	1.733	34.05	47.11	0.8559	5.6724	0.0401	1.5752	0.0347
288	3786	5	10	9	3.27	1.8	2	-26.14	3400.04	1.801	34.22	46.36	0.8587	5.7038	0.0397	1.7005	0.0290
289	3801	5	12.67	6.33	3.27	1	2.27	-30.09	3216.68	1.862	32.38	46.32	0.8780	5.7532	0.0390	1.8068	0.0252
290	3814	6.67	10	5	3.27	1.8	2.27	-24.29	3486.52	1.734	35.09	45.16	0.8554	5.5173	0.0506	1.6293	0.0342
291	3827	7.33	11.33	6.33	3.27	1	2.27	-28.22	3302.82	1.854	33.24	46.39	0.8730	5.7377	0.0392	1.7947	0.0229
292	3843	7	10	9	3	1	2.8	-25.61	3424.06	1.807	34.46	46.24	0.8625	5.6343	0.0419	1.7357	0.0294
293	3854	6.67	10	5	3.27	1	2.8	-22.24	3586.25	1.523	36.09	44.61	0.8494	5.4403	0.0480	1.2322	0.0431
294	3868	6	10	6.33	3.27	1	4.93	-25.32	3436.07	1.872	34.58	46.18	0.8497	5.7608	0.0388	1.8230	0.0258
295	3883	7	11.33	6.33	3.27	1	2	-27.36	3341.49	1.895	33.63	45.89	0.8579	5.6839	0.0399	1.8952	0.0170
296	3898	5	10	9	5.4	1	2.27	-30.30	3207.49	1.849	32.28	48.58	0.8779	5.7253	0.0405	1.7897	0.0219
297	3910	5	10	7	3	1.2	2.27	-21.02	3633.32	1.892	36.56	44.30	0.8356	5.7070	0.0396	1.8814	0.0181
298	3923	7.33	10	6.33	3	1	2.8	-22.94	3547.47	1.782	35.70	44.83	0.8427	5.7420	0.0391	1.6503	0.0315
299	3937	5.33	10	5	3.27	1	2.8	-21.21	3624.24	1.900	36.48	42.89	0.8419	5.7167	0.0395	1.8975	0.0197
300	3948	6.67	10	5	3.27	2.6	2.8	-27.92	3317.59	1.825	33.39	46.29	0.8542	5.7347	0.0393	1.7387	0.0268
301	3960	5	10	11.67	3.27	1	2.27	-25.53	3426.08	1.890	34.48	46.59	0.8742	5.6793	0.0401	1.8869	0.0201
302	3973	5	10	6.33	3.27	1	4.93	-24.96	3460.77	1.524	34.83	45.53	0.8572	5.5311	0.0454	1.2050	0.0401
303	3987	6	15.33	6.33	3.27	1	2.27	-39.15	2800.36	1.856	28.21	49.34	0.9288	5.6786	0.0400	1.8197	0.0194
304	4002	5	15.33	5	3.27	1	2.27	-37.95	2854.37	1.900	28.75	47.86	0.9542	5.6396	0.0437	1.9277	0.0139
305	4015	5	10	9	3.27	1	2	-23.13	3538.54	1.791	35.61	45.51	0.8576	5.6214	0.0424	1.7080	0.0298
306	4027	5	10	5	3	1	2.27	-18.14	3765.57	1.900	37.89	42.25	0.8420	5.7742	0.0386	1.8826	0.0166
307	4042	6.67	10	5	3.27	1	2	-21.32	3623.80	1.710	36.47	43.80	0.8379	5.3876	0.0491	1.6237	0.0333
308	4053	5	10	10.33	3.27	1	2.8	-25.07	3449.14	1.809	34.71	46.39	0.8600	5.4983	0.0513	1.7848	0.0240
309	4067	7	10	9	4.07	1	2.8	-29.12	3261.04	1.873	32.82	48.21	0.8819	5.7742	0.0386	1.8222	0.0244
310	4082	5	10	5	3.27	1.2	2.27	-20.26	3667.94	1.900	36.91	43.20	0.8451	5.7742	0.0386	1.9146	0.0172
311	4097	8	10	5	3.27	1	2.8	-24.08	3493.70	1.844	35.16	44.71	0.8570	5.7742	0.0386	1.7633	0.0292
312	4111	7.67	10	9	5.4	1	2.8	-34.43	3016.44	1.888	30.36	50.12	0.8922	5.6453	0.0409	1.8937	0.0164
313	4122	5.33	10	5	3	1	2.27	-18.65	3742.37	1.885	37.66	42.17	0.8529	5.7742	0.0386	1.8462	0.0200
314	4134	5	10	7	3	1.2	2	-20.68	3648.85	1.900	36.72	43.89	0.8705	5.7742	0.0386	1.8989	0.0185
315	4146	6.33	10	6.33	3.27	1	4.93	-25.43	3431.25	1.852	34.53	45.78	0.8509	5.6974	0.0398	1.8052	0.0235

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
316	4158	5.33	10	5	3.27	1	2	-19.74	3691.81	1.900	37.15	42.83	0.8363	5.7687	0.0387	1.8903	0.0179
317	4171	9	10	5	3.27	1	2	-23.78	3508.53	1.801	35.31	44.74	0.8608	5.5559	0.0454	1.7505	0.0250
318	4185	6	10	6.33	3	1	2.8	-21.11	3631.41	1.800	36.54	44.19	0.8429	5.7742	0.0386	1.6748	0.0300
319	4198	5.33	10	5	3	1.2	2.27	-19.90	3689.92	1.687	37.13	42.76	0.8436	5.1169	0.0575	1.6685	0.0306
320	4211	5	10	7	3	1	2.27	-19.69	3694.12	1.900	37.17	43.68	0.8485	5.7742	0.0386	1.8826	0.0166
321	4225	6.33	10	7	3	1.2	2	-23.26	3535.21	1.687	35.58	44.39	0.8661	5.7742	0.0386	1.4499	0.0406
322	4238	5	15.33	5	3.27	1	2	-37.14	2891.80	1.896	29.13	46.67	0.9425	5.6817	0.0400	1.8978	0.0183
323	4250	5	10	5	3.8	1	2.27	-21.57	3610.60	1.781	36.34	43.34	0.8463	5.7742	0.0386	1.6367	0.0327
324	4262	5	10	5	3.27	1.2	2	-19.99	3681.89	1.837	37.05	43.33	0.8382	5.4259	0.0470	1.8645	0.0204
325	4276	5	10	7	3.27	1	2.27	-20.66	3649.80	1.890	36.73	43.93	0.8510	5.6793	0.0401	1.8869	0.0201
326	4292	5.67	10	5	3	1	3.33	-21.08	3631.16	1.872	36.54	43.70	0.8556	5.6389	0.0413	1.8645	0.0187
327	4305	5	10	6.33	5.13	1	2.27	-28.36	3297.40	1.819	33.19	46.66	0.8692	5.3833	0.0530	1.8443	0.0226
328	4318	6.67	10	6.33	3	1	2.27	-21.54	3615.98	1.609	36.38	44.80	0.8692	5.4729	0.0439	1.3945	0.0350
329	4331	5.33	10	5	3.27	1.4	2.8	-22.71	3557.22	1.827	35.80	43.98	0.8535	5.7673	0.0387	1.7312	0.0257
330	4343	6.33	10	7	3	1.2	2.27	-22.39	3574.25	1.728	35.97	44.69	0.8542	5.6131	0.0418	1.5840	0.0344
331	4357	5.33	10	5	3	1.2	4.13	-22.51	3566.70	1.807	35.89	44.40	0.8372	5.6965	0.0398	1.7150	0.0282
332	4369	5	10.67	7	3.27	2.6	2.27	-30.04	3217.94	1.900	32.39	47.06	0.8706	5.6852	0.0410	1.9345	0.0139
333	4382	6.33	10	7	3	1	2.27	-22.70	3560.22	1.714	35.83	44.41	0.8658	5.4868	0.0459	1.5992	0.0348
334	4397	5.33	12.67	5	3.27	1	2.8	-30.66	3191.49	1.820	32.13	45.93	0.8881	5.7271	0.0403	1.7304	0.0274
335	4411	5.33	10.67	5	3.27	1	2.27	-22.26	3577.44	1.845	36.00	43.67	0.8481	5.6165	0.0413	1.8179	0.0249
336	4424	5.33	10	5	3.27	1.2	4.13	-22.34	3575.92	1.761	35.98	44.32	0.8496	5.7742	0.0386	1.5968	0.0316
337	4439	5	10	9	3	1.4	2	-23.87	3504.07	1.805	35.26	45.09	0.8625	5.7357	0.0392	1.6981	0.0326
338	4452	5.33	10	5	3	1	3.33	-20.12	3676.47	1.813	36.99	43.58	0.8494	5.7742	0.0386	1.7014	0.0315
339	4466	5.67	10	5	3	1.2	2.27	-19.81	3692.09	1.756	37.15	43.34	0.8452	5.7365	0.0392	1.5999	0.0343
340	4481	5.33	10	5	5.13	1	2.27	-27.14	3352.51	1.851	33.74	45.71	0.8466	5.6280	0.0440	1.8265	0.0205
341	4493	5.67	10	5	3.27	1	2.27	-20.90	3644.56	1.659	36.68	42.78	0.8533	5.6737	0.0415	1.4275	0.0361
342	4506	5	10	5	3	1	3.33	-19.06	3724.45	1.841	37.48	43.26	0.8310	5.7742	0.0386	1.7573	0.0296
343	4518	5.33	10	5	3.27	1	2.27	-20.44	3660.26	1.885	36.84	42.71	0.8562	5.5668	0.0429	1.9154	0.0139
344	4531	5	10	5	3	1	2.53	-18.75	3738.80	1.850	37.62	42.40	0.8482	5.7008	0.0397	1.7994	0.0212
345	4544	5	10	5	3	1.2	2.27	-18.96	3727.98	1.892	37.51	42.93	0.8244	5.7070	0.0396	1.8814	0.0181
346	4557	5.67	10	5	3.27	1.8	2.27	-22.99	3542.40	1.897	35.65	43.97	0.8498	5.7742	0.0386	1.8687	0.0203
347	4569	5	10	10.33	3	1	3.33	-24.73	3463.95	1.841	34.86	46.39	0.8515	5.7742	0.0386	1.7573	0.0296
348	4581	5.33	10	5	3.27	1.2	2	-20.29	3666.94	1.895	36.90	43.08	0.8457	5.7742	0.0386	1.8662	0.0208
349	4595	5	10	9	3	2.6	2.27	-29.24	3255.94	1.862	32.77	47.14	0.8537	5.6427	0.0422	1.8422	0.0206
350	4608	5.33	10	10.33	3.27	1	2	-25.16	3442.50	1.900	34.64	46.05	0.8562	5.7687	0.0387	1.8903	0.0179
351	4622	5	15.33	6.33	3	1	2.27	-38.19	2845.20	1.828	28.66	48.07	0.9292	5.7742	0.0386	1.7317	0.0284
352	4636	5.67	10	5	3.27	1	3.33	-21.68	3607.31	1.713	36.30	44.13	0.8561	5.5698	0.0439	1.5702	0.0284
353	4649	5	10	10.33	3	1	2.27	-22.89	3547.02	1.900	35.69	45.80	0.8711	5.7742	0.0386	1.8826	0.0166
354	4663	5.33	10	5.67	3	1	2.27	-19.39	3708.33	1.885	37.32	42.56	0.8515	5.7742	0.0386	1.8462	0.0200
355	4678	5	10	5	3.27	1	3.33	-20.87	3645.19	1.691	36.68	43.97	0.8499	5.7742	0.0386	1.4568	0.0353
356	4691	5	10	5	5.13	1	2.27	-26.82	3368.34	1.819	33.90	45.89	0.8686	5.3833	0.0530	1.8443	0.0226
357	4705	5.67	15.33	5	3	1	3.33	-38.41	2833.45	1.892	28.54	48.73	0.9344	5.7721	0.0386	1.8591	0.0185
358	4718	5	10	10.33	3.27	1.8	2.27	-28.12	3307.93	1.844	33.29	47.42	0.8706	5.7742	0.0386	1.7629	0.0284
359	4731	5.33	10	10.33	3.27	2.2	2.27	-30.07	3217.59	1.872	32.38	48.36	0.8945	5.6567	0.0402	1.8592	0.0209
360	4743	7.67	10	5	3	1	2.27	-21.49	3613.96	1.805	36.36	44.48	0.8548	5.6506	0.0412	1.7267	0.0308

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
361	4755	5	10	5	3	1.4	2.27	-20.98	3635.35	1.890	36.59	42.87	0.8415	5.5952	0.0419	1.9139	0.0140
362	4771	5	12.67	5	3	1	2.27	-28.42	3292.85	1.900	33.15	44.89	0.8955	5.7742	0.0386	1.8882	0.0208
363	4782	5	10	10.33	3.27	1	2	-24.27	3486.15	1.791	35.08	46.33	0.8668	5.6214	0.0424	1.7080	0.0298
364	4797	5.33	10	5	3.27	1.2	2.27	-21.13	3629.13	1.858	36.52	43.37	0.8584	5.7488	0.0390	1.7995	0.0267
365	4811	5	10	5	3.27	1	3.07	-20.30	3666.35	1.900	36.89	43.71	0.8405	5.7572	0.0389	1.8966	0.0160
366	4825	7.67	10	5	3	1.4	2.27	-23.73	3510.33	1.819	35.33	44.63	0.8735	5.6631	0.0402	1.7495	0.0241
367	4837	5	10	5.67	3	1	2.27	-18.56	3746.39	1.900	37.70	42.74	0.8425	5.7742	0.0386	1.8826	0.0166
368	4850	6.33	10	5	3	1	2.27	-20.40	3666.06	1.714	36.89	43.17	0.8581	5.4868	0.0459	1.5992	0.0348
369	4866	5.33	10	7.67	3	1.2	2.27	-22.31	3578.80	1.687	36.01	44.52	0.8502	5.1169	0.0575	1.6685	0.0306
370	4881	5	14	5	3	1	2.27	-32.91	3086.98	1.873	31.09	46.21	0.9195	5.7742	0.0386	1.8210	0.0251
371	4894	5	12.67	5	3.27	1	2.27	-28.85	3273.69	1.862	32.96	45.48	0.8777	5.7532	0.0390	1.8068	0.0252
372	4909	5	10	5	3.27	2.6	2	-26.16	3396.56	1.900	34.19	44.88	0.8512	5.6506	0.0432	1.9716	0.0083
373	4922	5	10	5	3	1	4.4	-21.42	3615.62	1.852	36.38	44.08	0.8410	5.7631	0.0388	1.7839	0.0258
374	4936	5	12.67	5	3.27	1	3.07	-30.20	3211.94	1.860	32.34	46.09	0.8940	5.6134	0.0417	1.8483	0.0230
375	4948	6	10	7.67	3.27	1	3.07	-24.25	3484.88	1.884	35.07	45.60	0.8616	5.7742	0.0386	1.8432	0.0184
376	4960	5	10	5.67	3.27	2.6	2	-26.82	3366.19	1.900	33.88	45.28	0.8528	5.6506	0.0432	1.9716	0.0083
377	4972	6.33	10.67	5	3	1	2.27	-23.32	3527.47	1.900	35.50	43.75	0.8644	5.7742	0.0386	1.9377	0.0158
378	4984	5.33	10	5.67	3	1	2	-19.31	3713.95	1.813	37.37	42.67	0.8487	5.7215	0.0394	1.7192	0.0308
379	4997	5	10	5.67	3.27	1	2.27	-19.59	3699.32	1.890	37.23	43.02	0.8452	5.6793	0.0401	1.8869	0.0201
380	5011	5.33	15.33	7.67	3.53	1	2.27	-40.71	2727.51	1.900	27.47	50.07	0.9477	5.7586	0.0389	2.0000	0.0000
381	5024	5	10.67	5	3	1.2	2.27	-22.19	3584.64	1.687	36.08	43.29	0.8451	5.6091	0.0432	1.5045	0.0327
382	5041	6.33	10	5.67	3	1	2.27	-21.00	3638.56	1.714	36.61	43.68	0.8637	5.4868	0.0459	1.5992	0.0348
383	5054	5	10	5	3.27	2.6	2.27	-25.67	3419.05	1.900	34.41	45.43	0.8609	5.6855	0.0399	1.9739	0.0080
384	5069	5.33	10	6.33	3.27	1	2.27	-21.78	3598.68	1.885	36.22	43.57	0.8601	5.5668	0.0429	1.9154	0.0139
385	5084	5	10.67	5	3	1	2.27	-21.43	3615.03	1.864	36.38	42.97	0.8579	5.7742	0.0386	1.8031	0.0231
386	5100	5	15.33	5	3	1	2.27	-37.23	2889.29	1.828	29.10	47.25	0.9313	5.7742	0.0386	1.7317	0.0284
387	5113	5.33	15.33	5.67	3	1	2.27	-38.21	2843.46	1.852	28.64	48.31	0.9470	5.7742	0.0386	1.7787	0.0225
388	5126	7.67	10	5.67	3.27	1	2	-22.98	3542.82	1.899	35.65	44.80	0.8543	5.7742	0.0386	1.8728	0.0170
389	5139	5	10	5	3.8	1.2	2.27	-22.85	3549.89	1.850	35.73	44.13	0.8462	5.7338	0.0393	1.7890	0.0264
390	5151	5	10	8.33	3	1	2.27	-20.81	3642.74	1.900	36.65	44.60	0.8581	5.7742	0.0386	1.8826	0.0166
391	5163	5	10	5	3	1	2	-17.86	3781.39	1.770	38.05	42.38	0.8448	5.5876	0.0409	1.6779	0.0278
392	5175	5	10	5	3	1.8	2.27	-21.70	3604.03	1.810	36.27	43.17	0.8388	5.7742	0.0386	1.6959	0.0321
393	5187	5.67	10	5.67	3.27	1	2.27	-21.41	3620.74	1.659	36.44	43.27	0.8558	5.6737	0.0415	1.4275	0.0361
394	5201	7	10	5	3.27	1	2.27	-22.05	3587.78	1.808	36.11	43.98	0.8480	5.6879	0.0405	1.7208	0.0265
395	5216	5	15.33	10.33	3.27	1.2	2	-42.07	2664.74	1.899	26.85	50.59	0.9498	5.7133	0.0396	1.8935	0.0172
396	5229	5	10	5.67	3.27	1.8	2.27	-23.76	3508.36	1.844	35.31	44.40	0.8393	5.7742	0.0386	1.7629	0.0284
397	5242	5.33	10.67	5	3	1	2.27	-21.73	3605.03	1.705	36.28	43.33	0.8488	5.7680	0.0387	1.4879	0.0343
398	5256	5	10.67	5	3.27	1	2.27	-21.91	3596.89	1.708	36.20	43.69	0.8564	5.7742	0.0386	1.4915	0.0367
399	5269	7.67	10	5	3	1	2	-21.10	3632.14	1.791	36.55	43.80	0.8630	5.7362	0.0392	1.6693	0.0300
400	5285	5.67	10	5	3	1	2.27	-18.92	3731.40	1.822	37.55	42.42	0.8357	5.7294	0.0393	1.7339	0.0296
401	5299	5	10	7.67	3.27	1	2.27	-21.28	3621.34	1.890	36.44	44.42	0.8563	5.6793	0.0401	1.8869	0.0201
402	5313	8	10	5	3	1	2	-21.46	3622.96	1.490	36.46	43.96	0.8505	5.7742	0.0386	1.0543	0.0396
403	5328	5	10	5	3	1.4	2	-20.16	3674.97	1.805	36.98	42.51	0.8431	5.7357	0.0392	1.6981	0.0326
404	5341	5	10	5	4.07	1	2.27	-22.47	3568.24	1.824	35.91	43.80	0.8415	5.7538	0.0390	1.7295	0.0285
405	5354	5	10	6.33	3	1	4.4	-22.89	3548.41	1.852	35.71	44.87	0.8453	5.7631	0.0388	1.7839	0.0258

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
406	5368	5	10	5	3	1	2.8	-18.89	3731.42	1.890	37.55	42.45	0.8273	5.7742	0.0386	1.8562	0.0220
407	5381	8	10	5	3	1	2.8	-22.93	3548.32	1.779	35.71	44.69	0.8512	5.7508	0.0390	1.6404	0.0323
408	5393	5	10	5	3	1.8	2.8	-22.78	3553.57	1.838	35.76	44.07	0.8322	5.7742	0.0386	1.7520	0.0271
409	5406	5	10	5	3.27	3	2.27	-27.08	3356.27	1.825	33.78	45.82	0.8568	5.7742	0.0386	1.7248	0.0321
410	5420	5	13.33	5	3	1	2.27	-31.33	3160.03	1.842	31.82	45.99	0.9131	5.7742	0.0386	1.7599	0.0281
411	5435	5	11.33	5.67	3.27	1.4	2	-26.38	3386.56	1.900	34.09	45.28	0.8554	5.7742	0.0386	1.8982	0.0172
412	5448	5	15.33	5	3	1	2.8	-37.64	2872.05	1.764	28.93	48.06	0.9471	5.1446	0.0587	1.8134	0.0202
413	5461	5	11.33	5	3	1	2.27	-24.16	3488.83	1.900	35.11	44.23	0.8748	5.7742	0.0386	1.9374	0.0121
414	5475	5	10	5	3	1.2	2.8	-20.48	3660.33	1.807	36.84	42.84	0.8426	5.5090	0.0466	1.7775	0.0270
415	5489	5.67	10	5.67	3	1	2.27	-19.60	3700.15	1.822	37.23	42.84	0.8360	5.7294	0.0393	1.7339	0.0296
416	5505	5	10	7.67	3	1	2.27	-20.39	3661.94	1.900	36.85	44.13	0.8534	5.7742	0.0386	1.8826	0.0166
417	5516	5.67	10	10.33	3	1	2.27	-24.06	3495.19	1.822	35.17	45.86	0.8634	5.7294	0.0393	1.7339	0.0296
418	5530	5	11.33	5	3.27	1	2.27	-24.22	3486.04	1.900	35.09	44.21	0.8524	5.7532	0.0390	1.9005	0.0183
419	5542	5	10	5.67	5.13	1	2.27	-27.64	3330.57	1.819	33.52	46.30	0.8696	5.3833	0.0530	1.8443	0.0226
420	5556	5	10	5	3	1.4	3.33	-22.00	3589.37	1.838	36.12	43.47	0.8533	5.5425	0.0464	1.8277	0.0223
421	5569	5.67	10	5	3	1	2.8	-20.22	3674.36	1.720	36.98	43.09	0.8427	5.7742	0.0386	1.5157	0.0368
422	5581	5	10	7.67	5.13	1	2.27	-29.68	3236.65	1.819	32.58	47.48	0.8735	5.3833	0.0530	1.8443	0.0226
423	5593	5	12.67	5	3.27	1.8	2	-31.88	3134.19	1.881	31.56	46.17	0.9063	5.6847	0.0399	1.8671	0.0170
424	5605	5.67	11.33	5	3	1	2.27	-24.71	3467.26	1.746	34.90	43.71	0.8672	5.6388	0.0412	1.6120	0.0345
425	5618	5	10	5.67	3	1	3.33	-19.85	3688.55	1.841	37.11	43.66	0.8316	5.7742	0.0386	1.7573	0.0296
426	5630	5.67	10	5	3	2.6	2.27	-25.99	3406.08	1.829	34.28	45.42	0.8539	5.7093	0.0396	1.7549	0.0302
427	5643	5	10	11	3	1	2.27	-23.71	3509.21	1.900	35.31	46.13	0.8738	5.7742	0.0386	1.8826	0.0166
428	5656	5.67	10	5.67	3	1	3.33	-21.57	3608.24	1.872	36.31	44.15	0.8577	5.6389	0.0413	1.8645	0.0187
429	5669	5	11.33	6.33	3.27	1	2.27	-25.58	3423.50	1.900	34.46	45.08	0.8565	5.7532	0.0390	1.9005	0.0183
430	5683	5	15.33	5.67	3	1	2.27	-37.77	2864.47	1.828	28.85	47.67	0.9310	5.7742	0.0386	1.7317	0.0284
431	5696	5.67	11.33	6.33	3.27	1	2.27	-26.53	3382.83	1.774	34.05	45.22	0.8450	5.7742	0.0386	1.6226	0.0336
432	5711	6.33	10	8.33	3	1.2	2.27	-23.87	3506.12	1.728	35.28	45.49	0.8604	5.6131	0.0418	1.5840	0.0344
433	5724	7.67	10.67	5	3	1	2.27	-24.15	3489.25	1.900	35.11	45.07	0.8774	5.7337	0.0393	1.9034	0.0189
434	5739	5	10	7.67	3	1	3.33	-21.92	3592.89	1.841	36.15	44.93	0.8398	5.7742	0.0386	1.7573	0.0296
435	5753	5	10	5.67	5.4	1	2.27	-27.32	3344.57	1.849	33.66	46.55	0.8675	5.7253	0.0405	1.7897	0.0219
436	5767	5	10	7.67	3	2.6	2.27	-28.06	3310.25	1.862	33.32	46.42	0.8493	5.6427	0.0422	1.8422	0.0206
437	5780	5	10	5	3	2	2.27	-23.27	3529.48	1.900	35.52	44.04	0.8471	5.7742	0.0386	1.9916	0.0046
438	5793	5	12.67	5.67	3	1	2.27	-28.96	3268.06	1.900	32.90	45.33	0.8962	5.7742	0.0386	1.8882	0.0208
439	5806	5	10	5	3	2.6	2.27	-25.28	3438.15	1.862	34.61	44.73	0.8390	5.6427	0.0422	1.8422	0.0206
440	5819	5	10	5.67	3	1	2	-18.54	3750.16	1.770	37.73	42.86	0.8472	5.5876	0.0409	1.6779	0.0278
441	5832	5	10	5	5.13	1.8	2.27	-29.51	3242.88	1.893	32.64	46.93	0.8437	5.7369	0.0400	1.8743	0.0186
442	5845	6.67	10	5.67	3	1	2.27	-20.87	3647.25	1.609	36.70	44.37	0.8663	5.4729	0.0439	1.3945	0.0350
443	5862	6.67	10	5.67	3	1.8	2.27	-25.34	3436.18	1.818	34.59	44.50	0.8765	5.6888	0.0399	1.7406	0.0286
444	5874	5	10	5.67	3	1	3.6	-20.79	3648.71	1.693	36.72	43.61	0.8462	5.7296	0.0393	1.4770	0.0334
445	5886	5	10	5	3.53	1	2.27	-21.07	3634.48	1.753	36.58	42.99	0.8241	5.5059	0.0453	1.6715	0.0339
446	5899	7.67	10	5.67	3	1	3.6	-23.80	3508.69	1.748	35.31	45.05	0.8373	5.6139	0.0420	1.6247	0.0326
447	5914	5	10	5	3.53	1.8	2.27	-24.18	3487.80	1.900	35.11	44.03	0.8485	5.7688	0.0387	1.9064	0.0165
448	5927	5	10	7.67	3	1.8	2.27	-23.85	3504.95	1.810	35.27	44.98	0.8469	5.7742	0.0386	1.6959	0.0321
449	5943	5	10	5.67	4.07	1	2.27	-22.99	3544.40	1.824	35.67	44.26	0.8434	5.7538	0.0390	1.7295	0.0285
450	5956	5.33	10	5	3	2.6	2.27	-25.53	3427.99	1.796	34.50	44.78	0.8362	5.6603	0.0402	1.7047	0.0302

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
451	5971	5	10	5	3	1	3.6	-20.21	3675.28	1.693	36.98	43.08	0.8412	5.7296	0.0393	1.4770	0.0334
452	5985	5.33	10	5.67	3	1.8	2.27	-23.19	3535.83	1.795	35.59	43.80	0.8447	5.5726	0.0424	1.7331	0.0286
453	5998	6.67	10	5	3	1	2.27	-20.19	3678.48	1.609	37.01	43.87	0.8617	5.4729	0.0439	1.3945	0.0350
454	6010	5	12.67	5	3	1	3.6	-29.86	3226.62	1.900	32.48	45.81	0.8945	5.7742	0.0386	1.9054	0.0160
455	6024	5	10	5	3	1.8	3.33	-23.12	3537.85	1.840	35.61	44.12	0.8420	5.6826	0.0399	1.7855	0.0244
456	6037	5	10.67	5.67	3	1	2.27	-21.99	3589.36	1.864	36.12	43.45	0.8604	5.7742	0.0386	1.8031	0.0231
457	6050	5	10	5.67	3	2.6	2.27	-26.14	3398.54	1.862	34.21	45.15	0.8394	5.6427	0.0422	1.8422	0.0206
458	6063	6.67	10	6.33	3	1	3.6	-23.89	3503.06	1.823	35.25	44.91	0.8596	5.7703	0.0387	1.7233	0.0302
459	6076	5	10	5	3.53	1.8	3.33	-24.97	3452.24	1.862	34.74	45.14	0.8432	5.7612	0.0388	1.8036	0.0233
460	6089	6.67	10	5	3.53	1	2	-22.61	3562.61	1.782	35.86	43.65	0.8683	5.6405	0.0411	1.6835	0.0298
461	6103	6.67	10	5	3	1	3.6	-22.33	3574.88	1.823	35.98	44.11	0.8597	5.7703	0.0387	1.7233	0.0302
462	6116	5	10	5.67	3.53	1	2.27	-21.62	3608.81	1.753	36.32	43.45	0.8261	5.5059	0.0453	1.6715	0.0339
463	6134	5	10	5.67	3	1.4	2.27	-21.72	3601.32	1.890	36.24	43.32	0.8429	5.5952	0.0419	1.9139	0.0140
464	6150	5	16	5	3	1.2	2.27	-39.70	2773.76	1.900	27.95	47.74	0.9367	5.7742	0.0386	1.9333	0.0139
465	6165	5	10	5	3.53	1	2	-20.83	3642.76	1.854	36.66	42.88	0.8409	5.6952	0.0398	1.8087	0.0227
466	6180	5	12.67	5.67	3	1	2	-29.44	3245.81	1.900	32.68	45.28	0.8834	5.7545	0.0389	1.9190	0.0155
467	6193	5	10	5.67	3.53	1	2	-21.59	3607.80	1.854	36.31	43.30	0.8402	5.6952	0.0398	1.8087	0.0227
468	6206	5	10	5	3.27	1	4.4	-22.25	3579.35	1.782	36.02	43.79	0.8391	5.7518	0.0390	1.6462	0.0288
469	6218	5	10	6.33	3	1.4	2.27	-22.29	3574.70	1.890	35.98	43.76	0.8443	5.5952	0.0419	1.9139	0.0140
470	6232	5	15.33	7.67	3.27	1	2.27	-39.83	2767.97	1.900	27.88	49.38	0.9521	5.6396	0.0437	1.9277	0.0139
471	6245	5	10	5	3	1.8	2	-21.68	3603.94	1.846	36.27	43.01	0.8513	5.7050	0.0397	1.7899	0.0258
472	6259	5	10	5	5.4	1	2.27	-26.68	3373.99	1.849	33.96	46.10	0.8644	5.7253	0.0405	1.7897	0.0219
473	6274	7.67	10	5	3.27	1	2.27	-22.44	3567.66	1.897	35.90	44.46	0.8622	5.6524	0.0435	1.9099	0.0192
474	6287	5	10	5	3.27	1	2.8	-19.99	3682.69	1.809	37.06	43.08	0.8348	5.4983	0.0513	1.7848	0.0240
475	6301	6.33	10	5	3	1.2	2.27	-20.53	3659.68	1.728	36.83	43.34	0.8448	5.6131	0.0418	1.5840	0.0344
476	6315	5	10	6.33	3	1.8	2	-22.92	3546.97	1.846	35.70	43.86	0.8515	5.7050	0.0397	1.7899	0.0258
477	6330	5	15.33	5	3	1	2	-36.51	2923.70	1.762	29.45	47.59	0.9334	5.7479	0.0391	1.6079	0.0352
478	6344	6.33	10	5.67	3	1	2	-20.51	3658.86	1.800	36.82	43.55	0.8456	5.5418	0.0434	1.7537	0.0256
479	6360	5	10	5	3.27	2.6	2.8	-26.50	3383.24	1.800	34.05	46.08	0.8625	5.7742	0.0386	1.6750	0.0303
480	6372	5	10	10.33	3	1	2	-23.44	3524.74	1.770	35.47	45.69	0.8748	5.5876	0.0409	1.6779	0.0278
481	6387	5	10	6.33	3	1.2	2.27	-20.30	3666.43	1.892	36.89	43.86	0.8312	5.7070	0.0396	1.8814	0.0181
482	6402	6.33	10	5	4.07	1	2.27	-24.19	3489.67	1.802	35.12	44.07	0.8392	5.7742	0.0386	1.6802	0.0266
483	6417	5	10	6.33	3	1	2	-19.08	3725.40	1.770	37.48	43.31	0.8499	5.5876	0.0409	1.6779	0.0278
484	6430	5	10	5	3	1.4	2.8	-20.79	3647.62	1.734	36.71	43.31	0.8373	5.7742	0.0386	1.5428	0.0346
485	6442	5	10	5.67	3.27	1	2	-19.75	3693.99	1.791	37.17	43.43	0.8441	5.6214	0.0424	1.7080	0.0298
486	6455	5	10	5	4.07	1	2	-22.41	3569.19	1.895	35.92	43.84	0.8414	5.6510	0.0404	1.9056	0.0158
487	6473	5	11.33	5.67	3	1	2	-24.03	3496.09	1.838	35.19	43.95	0.8507	5.7742	0.0386	1.7514	0.0244
488	6487	5	10	6.33	3	1.4	2.8	-22.07	3588.82	1.734	36.11	44.19	0.8408	5.7742	0.0386	1.5428	0.0346
489	6499	5	11.33	5.67	3	1	2.27	-24.82	3458.48	1.900	34.81	44.64	0.8757	5.7742	0.0386	1.9374	0.0121
490	6512	5	10	5	3	2.6	2.8	-25.65	3419.98	1.900	34.42	45.44	0.8450	5.7742	0.0386	1.8803	0.0204
491	6527	5	10	10.33	3	1	2.8	-23.98	3497.03	1.890	35.19	45.84	0.8504	5.7742	0.0386	1.8562	0.0220
492	6544	5.67	10	5	3	2.6	2.8	-25.98	3405.11	1.891	34.27	45.34	0.8517	5.6099	0.0439	1.9121	0.0162
493	6558	5	12.67	5	3	1	2	-28.90	3270.61	1.900	32.93	44.86	0.8832	5.7545	0.0389	1.9190	0.0155
494	6574	5	10	5	3	1	3.87	-19.96	3684.75	1.785	37.08	43.49	0.8410	5.7310	0.0396	1.6604	0.0320
495	6587	9	10	5	3	1	2.27	-23.02	3542.84	1.820	35.65	44.83	0.8540	5.5487	0.0434	1.7910	0.0255

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
496	6600	5	10	5.67	3	1	2.8	-19.63	3697.41	1.890	37.21	42.90	0.8278	5.7742	0.0386	1.8562	0.0220
497	6614	5.67	15.33	5	3	1	2.8	-37.33	2884.88	1.812	29.06	47.75	0.9303	5.3381	0.0507	1.8446	0.0196
498	6627	6.33	10	5	3	1	2.8	-21.00	3640.31	1.633	36.63	43.48	0.8493	5.5849	0.0471	1.4046	0.0415
499	6643	5	10	5.67	3	1	3.07	-19.55	3704.05	1.765	37.27	43.21	0.8185	5.5406	0.0452	1.6829	0.0279
500	6655	5	10	6.33	3	1.8	2.8	-24.14	3491.03	1.838	35.13	44.92	0.8381	5.7742	0.0386	1.7520	0.0271
501	6667	5	10	5	3.53	1	2.53	-21.60	3610.50	1.740	36.34	43.15	0.8523	5.5889	0.0416	1.6170	0.0342
502	6679	5	10.67	5	3.53	1	2.53	-23.53	3517.62	1.900	35.40	44.39	0.8507	5.7742	0.0386	1.9234	0.0148
503	6694	5	10	6.33	3.27	1.2	2.8	-23.06	3540.59	1.837	35.63	44.04	0.8495	5.7742	0.0386	1.7493	0.0308
504	6706	5	10	6.33	3	1	3.87	-21.32	3622.26	1.785	36.45	44.40	0.8491	5.7310	0.0396	1.6604	0.0320
505	6719	5	16	5	3	1	2.8	-39.57	2783.30	1.761	28.04	48.39	0.9641	5.7668	0.0387	1.5990	0.0371
506	6732	6.33	10	5.67	3	1	3.6	-22.76	3557.46	1.718	35.80	44.45	0.8505	5.7277	0.0394	1.5263	0.0347
507	6746	5	11.33	5	3	1	4.13	-26.43	3385.43	1.850	34.08	44.87	0.8782	5.7679	0.0387	1.7778	0.0259
508	6760	5	10	5	5.4	1	2	-27.78	3321.96	1.900	33.44	45.73	0.8833	5.7742	0.0386	1.8853	0.0207
509	6773	6.33	10	6.33	3.27	1	2.8	-23.05	3544.64	1.691	35.67	44.73	0.8562	5.6717	0.0401	1.4915	0.0366
510	6788	5	10	5	4.33	1	2.8	-24.51	3474.57	1.807	34.97	45.36	0.8580	5.6635	0.0401	1.7264	0.0291
511	6802	5	12.67	5	3	1.8	2.27	-32.00	3128.79	1.872	31.50	46.50	0.9095	5.7363	0.0392	1.8313	0.0238
512	6816	5	15.33	6.33	3	1	2.8	-38.76	2820.51	1.764	28.41	48.80	0.9436	5.1446	0.0587	1.8134	0.0202
513	6831	5	10.67	5	3	1	2.8	-21.62	3606.96	1.844	36.30	43.48	0.8454	5.6667	0.0401	1.7991	0.0285
514	6845	5	12.67	7	3	1	2	-30.42	3200.85	1.900	32.23	46.14	0.8880	5.7545	0.0389	1.9190	0.0155
515	6857	5	10	5.67	3.27	1	2.8	-20.71	3649.57	1.809	36.73	43.49	0.8362	5.4983	0.0513	1.7848	0.0240
516	6870	5	10	5	4.33	1	2	-23.32	3527.59	1.880	35.50	43.98	0.8454	5.6566	0.0407	1.8736	0.0200
517	6884	5	10	6.33	3	1	3.33	-20.59	3654.51	1.841	36.77	44.09	0.8346	5.7742	0.0386	1.7573	0.0296
518	6897	5	10	5	3.8	1	2	-21.57	3610.59	1.788	36.34	43.32	0.8608	5.7742	0.0386	1.6509	0.0282
519	6911	5	11.33	6.33	3	1	3.87	-26.91	3364.35	1.806	33.86	45.29	0.8627	5.7742	0.0386	1.6865	0.0264
520	6923	5	10.67	6.33	3	1	2.27	-22.53	3564.60	1.864	35.87	43.91	0.8639	5.7742	0.0386	1.8031	0.0231
521	6935	5	10	5	3.27	1.4	2.27	-20.90	3643.58	1.694	36.67	42.69	0.8314	5.3122	0.0515	1.6176	0.0315
522	6949	5	10	7.67	3	1	2.53	-21.43	3615.65	1.850	36.38	44.16	0.8603	5.7008	0.0397	1.7994	0.0212
523	6965	5	10	7.67	3.27	1	2	-21.73	3602.98	1.791	36.25	44.74	0.8522	5.6214	0.0424	1.7080	0.0298
524	6977	5	11.33	5	3.27	1	2.8	-25.16	3447.37	1.712	34.70	44.65	0.8682	5.1510	0.0639	1.7062	0.0260
525	6991	5.67	10	10.33	3	1	3.33	-26.30	3391.03	1.872	34.12	46.90	0.8781	5.6389	0.0413	1.8645	0.0187
526	7007	5	10	5	3	1	4.13	-20.46	3662.55	1.741	36.86	43.52	0.8414	5.7742	0.0386	1.5580	0.0319
527	7019	5	10	5	5.13	1	4.93	-29.89	3228.22	1.774	32.49	47.26	0.8686	5.7742	0.0386	1.6230	0.0381
528	7033	7.67	10	5	3.27	1	2	-22.34	3572.23	1.899	35.95	44.34	0.8531	5.7742	0.0386	1.8728	0.0170
529	7046	5	12.67	6.33	3	1	2	-30.10	3215.43	1.900	32.37	45.64	0.8831	5.7545	0.0389	1.9190	0.0155
530	7062	5	10	5	3	1.2	4.4	-22.37	3575.10	1.736	35.98	44.53	0.8425	5.5055	0.0430	1.6371	0.0280
531	7076	5.67	10	5	3	1	4.13	-21.62	3608.44	1.770	36.31	44.54	0.8589	5.4671	0.0472	1.7168	0.0265
532	7087	5	10	5	3	1	5.47	-22.21	3578.79	1.883	36.01	44.51	0.8335	5.7198	0.0400	1.8602	0.0198
533	7101	5.67	10	5	3	1	2	-18.72	3741.60	1.789	37.65	42.43	0.8280	5.7734	0.0386	1.6528	0.0316
534	7113	5.67	10	5	3.27	1.2	2.27	-20.31	3666.09	1.878	36.89	43.22	0.8420	5.7742	0.0386	1.8307	0.0248
535	7127	5.67	10	6.33	3	1	2	-19.88	3688.40	1.789	37.11	43.38	0.8340	5.7734	0.0386	1.6528	0.0316
536	7140	5	10.67	10.33	3	1	2.53	-27.03	3359.29	1.786	33.81	46.35	0.8639	5.5967	0.0407	1.7058	0.0297
537	7154	5	10.67	5	3	1	2	-21.03	3635.37	1.788	36.59	42.92	0.8587	5.7742	0.0386	1.6519	0.0329
538	7169	5	15.33	6.33	3	1	2	-37.63	2872.20	1.762	28.93	48.40	0.9349	5.7479	0.0391	1.6079	0.0352
539	7184	5	10	5.67	5.13	1	2.8	-28.41	3293.73	1.872	33.15	46.75	0.8514	5.7742	0.0386	1.8185	0.0261
540	7198	5	10	5	3	1	3.07	-18.81	3738.08	1.765	37.61	42.78	0.8166	5.5406	0.0452	1.6829	0.0279

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
541	7213	5	10	11.67	3	1	2.27	-24.39	3477.91	1.900	34.99	46.53	0.8805	5.7742	0.0386	1.8826	0.0166
542	7228	5.67	10	5	3	1	3.07	-19.97	3684.74	1.758	37.08	43.45	0.8333	5.7742	0.0386	1.5915	0.0304
543	7244	5	10	5	3	2.6	2	-24.87	3456.73	1.867	34.79	44.84	0.8531	5.7742	0.0386	1.8092	0.0251
544	7258	5	10	6.33	3	1	3.07	-20.23	3672.82	1.765	36.96	43.68	0.8222	5.5406	0.0452	1.6829	0.0279
545	7273	7.67	10	5	3	1	2.8	-22.89	3553.61	1.626	35.76	44.16	0.8402	5.7742	0.0386	1.3280	0.0346
546	7288	5.33	10	5	3	1	2	-18.49	3751.70	1.813	37.75	42.28	0.8487	5.7215	0.0394	1.7192	0.0308
547	7303	5.67	11.33	5	3	1	2	-24.01	3496.99	1.834	35.20	43.89	0.8622	5.7454	0.0391	1.7528	0.0285
548	7315	5	10	5	5.13	1	3.07	-27.73	3326.87	1.800	33.48	46.32	0.8731	5.3981	0.0484	1.8002	0.0244
549	7327	5	10	5	5.13	1.4	2.27	-29.15	3259.36	1.891	32.81	46.51	0.8767	5.6803	0.0400	1.8879	0.0187
550	7342	5	10	5	3.53	1	2.8	-21.38	3619.85	1.767	36.43	43.32	0.8374	5.7742	0.0386	1.6089	0.0363
551	7355	5.33	10	5	3	1	5.2	-23.12	3542.68	1.647	35.65	44.52	0.8492	5.6277	0.0413	1.4181	0.0367
552	7368	5.67	11.33	5	3	1	4.13	-26.41	3389.68	1.721	34.11	45.99	0.8566	5.7742	0.0386	1.5170	0.0384
553	7380	5.33	10	5	3	1.8	2	-22.38	3573.51	1.774	35.97	43.25	0.8443	5.6179	0.0425	1.6755	0.0333
554	7394	5.33	10	5	3	1.4	2.8	-21.13	3634.12	1.648	36.57	44.06	0.8403	5.7304	0.0393	1.3866	0.0380
555	7409	5	10	5	3	1.6	2.8	-22.03	3586.69	1.897	36.10	43.66	0.8318	5.7742	0.0386	1.8696	0.0190
556	7422	5	15.33	5.67	3	1	2	-37.09	2897.02	1.762	29.18	47.97	0.9332	5.7479	0.0391	1.6079	0.0352
557	7434	5.67	10	5.67	3	1	2	-19.38	3711.29	1.789	37.35	42.89	0.8299	5.7734	0.0386	1.6528	0.0316
558	7448	5	10	5	3	2.6	3.33	-26.02	3404.25	1.846	34.26	45.48	0.8333	5.5181	0.0466	1.8533	0.0207
559	7460	5.67	10	10.33	3	1	2	-24.14	3492.37	1.789	35.14	45.77	0.8504	5.7734	0.0386	1.6528	0.0316
560	7475	6	10	5	3	1	2.27	-19.35	3712.20	1.811	37.36	42.86	0.8465	5.6527	0.0403	1.7372	0.0307
561	7491	5.67	10	7.67	3	1	2	-21.28	3624.04	1.789	36.47	44.25	0.8404	5.7734	0.0386	1.6528	0.0316
562	7506	5	10	6.33	4.07	1	2.27	-23.63	3514.99	1.824	35.37	44.69	0.8470	5.7538	0.0390	1.7295	0.0285
563	7522	5.67	10	5	3	1.4	2	-20.01	3681.78	1.799	37.05	43.12	0.8429	5.6050	0.0443	1.7306	0.0292
564	7534	6	10	5	3	1	2	-19.57	3700.12	1.894	37.23	42.79	0.8378	5.7184	0.0395	1.8810	0.0170
565	7548	5	10	7.67	3	1.4	2.8	-23.13	3540.19	1.734	35.62	45.10	0.8483	5.7742	0.0386	1.5428	0.0346
566	7561	5.33	10	5	3	1	2.8	-19.61	3702.31	1.717	37.25	43.41	0.8485	5.0828	0.0569	1.7395	0.0270
567	7573	5.33	12.67	5	3	1	2.27	-28.40	3296.75	1.768	33.19	45.29	0.8881	5.5780	0.0444	1.6764	0.0306
568	7584	7.67	10	5	3	2.6	2	-27.50	3336.05	1.866	33.58	46.15	0.8712	5.4494	0.0474	1.9154	0.0158
569	7598	7.67	10	5	3.27	1.8	2.27	-25.41	3433.01	1.824	34.55	45.57	0.8690	5.7742	0.0386	1.7238	0.0294
570	7612	6	10	5	3	1.4	2	-20.74	3648.48	1.803	36.72	43.18	0.8350	5.7479	0.0391	1.6895	0.0323
571	7627	5	10	7.67	3	1	2	-20.52	3659.18	1.770	36.82	44.16	0.8581	5.5876	0.0409	1.6779	0.0278
572	7642	6	12.67	5	3	1	2	-29.07	3267.05	1.714	32.89	45.15	0.8928	5.7542	0.0390	1.5109	0.0373
573	7655	6	10	5	3	1	4.4	-22.55	3566.60	1.732	35.89	44.20	0.8454	5.7259	0.0398	1.5561	0.0342
574	7669	5.33	10	5.67	3	1.8	2	-23.08	3541.30	1.774	35.64	43.63	0.8431	5.6179	0.0425	1.6755	0.0333
575	7682	5.33	10	5	3	1.4	2.27	-20.99	3635.56	1.848	36.59	43.27	0.8574	5.6413	0.0424	1.8147	0.0223
576	7698	6	10	5	3	1.4	2.27	-21.33	3623.54	1.706	36.47	43.29	0.8505	5.7742	0.0386	1.4874	0.0315
577	7710	5	10	5	3	1.8	4.4	-25.01	3454.07	1.719	34.76	45.09	0.8386	5.6821	0.0399	1.5448	0.0344
578	7726	5	10	7.67	3	1.4	2	-22.72	3557.34	1.805	35.80	44.22	0.8539	5.7357	0.0392	1.6981	0.0326
579	7740	6	10	5	3	2.6	2.27	-26.24	3394.21	1.857	34.16	45.77	0.8542	5.4859	0.0456	1.8850	0.0203
580	7756	5	10	5	5.13	2.2	2	-31.01	3173.60	1.900	31.95	47.31	0.8777	5.7742	0.0386	1.9663	0.0090
581	7769	5	10.67	7.67	3	1	2.27	-23.87	3503.00	1.864	35.25	44.78	0.8698	5.7742	0.0386	1.8031	0.0231
582	7781	6.33	10	5	3	1.4	2.27	-21.02	3635.94	1.785	36.59	43.99	0.8415	5.5054	0.0440	1.7348	0.0293
583	7793	8.67	10	5	3	1	2	-22.20	3580.39	1.833	36.03	44.00	0.8646	5.6033	0.0438	1.7986	0.0238
584	7805	6	10	5	3	1.4	2.8	-21.71	3602.68	1.842	36.25	44.14	0.8401	5.5245	0.0432	1.8432	0.0197
585	7818	6.33	10	5	3	1	2	-19.87	3688.24	1.800	37.11	43.07	0.8421	5.5418	0.0434	1.7537	0.0256

Calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	Goal1	Goal2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
586	7832	6.33	10	5	3	1.4	2	-21.88	3595.95	1.804	36.19	43.65	0.8459	5.7742	0.0386	1.6828	0.0316
587	7848	6	10	5	3	1	3.33	-21.46	3615.72	1.781	36.38	44.29	0.8696	5.6688	0.0401	1.6718	0.0291
588	7862	5.33	10	5	3	1.2	2	-19.38	3713.18	1.701	37.36	42.95	0.8424	5.1916	0.0635	1.6707	0.0260
589	7874	5	10	5	3	1.2	2	-18.60	3744.45	1.900	37.68	42.57	0.8622	5.7742	0.0386	1.8989	0.0185
590	7889	5	12.67	5	3	2.6	2.27	-33.95	3038.45	1.900	30.60	47.41	0.8776	5.7742	0.0386	1.9545	0.0104
591	7907	5.67	10	5	5.13	1	2.27	-26.90	3362.83	1.883	33.85	46.00	0.8680	5.7742	0.0386	1.8412	0.0194
592	7921	5	10	5	5.13	1	2.8	-27.59	3331.54	1.872	33.53	46.42	0.8547	5.7742	0.0386	1.8185	0.0261
593	7936	5	10	5	3	2.6	4.4	-27.33	3343.73	1.868	33.65	46.73	0.8526	5.7103	0.0408	1.8324	0.0242
594	7953	5.67	10	5	3	1.4	2.27	-20.77	3645.67	1.858	36.69	43.12	0.8440	5.6204	0.0432	1.8428	0.0192
595	7968	5	11.33	5	3	1	2	-23.51	3519.92	1.838	35.43	43.48	0.8495	5.7742	0.0386	1.7514	0.0244
596	7983	5	10	5	3.27	1.8	2	-22.56	3564.56	1.801	35.87	43.85	0.8414	5.7038	0.0397	1.7005	0.0290
597	7996	5.67	12.67	7.67	3	1	4.4	-33.23	3072.09	1.870	30.93	48.10	0.8911	5.6048	0.0429	1.8712	0.0203
598	8008	5.67	12.67	5	3	1	2	-28.74	3278.99	1.856	33.01	44.96	0.8979	5.7742	0.0386	1.7881	0.0294
599	8023	5	10	7.67	4.07	1	2.27	-24.77	3462.65	1.824	34.85	45.60	0.8534	5.7538	0.0390	1.7295	0.0285
600	8035	7	10	5	3	1	2	-20.66	3660.53	1.451	36.84	43.21	0.8555	5.6085	0.0410	1.0324	0.0443
601	8049	9	10	10.33	3	1	2.27	-28.21	3304.45	1.820	33.25	47.95	0.8729	5.5487	0.0434	1.7910	0.0255
602	8062	5	10	5	6.2	1	2.27	-31.57	3148.31	1.872	31.70	46.67	0.8869	5.7742	0.0386	1.8198	0.0267
603	8075	5	12	5	3	1	2	-25.36	3434.36	1.864	34.57	44.56	0.8826	5.7742	0.0386	1.8030	0.0244
604	8089	5	10	5	3	1.4	4.13	-22.71	3559.95	1.700	35.83	43.88	0.8323	5.6070	0.0415	1.5319	0.0354
605	8103	5.67	10	5.67	3	1	4.13	-22.12	3585.51	1.770	36.08	44.97	0.8593	5.4671	0.0472	1.7168	0.0265

Reliability Domain: GA-SO Run Resulting in the BSF (Run 1)

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
1	22	15	4	8.8	7.2	-36047.01	-571.80	-0.8182	-43.74	20.29	10.17	13.28	0.6872	0.3691	0.6089	0.8182	0.0397
2	41	11.53	5.2	6.4	11.53	-32879.36	-523.63	-1.4685	-45.00	16.02	14.66	14.33	0.5788	0.5456	0.6121	1.4685	0.0333
3	65	2.87	1	2.8	2	-5617.95	-187.12	-0.8852	-11.91	5.49	2.96	3.46	0.1936	0.1095	0.1524	0.8852	0.0331
4	93	13.27	2.2	6.4	6.33	-27251.14	-461.38	-0.7758	-36.67	17.50	7.93	11.23	0.6359	0.3100	0.5608	0.7758	0.0334
5	116	2	5.2	7.6	9.8	-10463.28	-253.46	-0.4698	-29.97	9.05	13.23	7.70	0.3600	0.4853	0.2942	0.4698	0.0283
6	137	6.33	4.6	8.8	10.67	-17067.83	-332.83	-0.7818	-38.03	13.87	13.36	10.80	0.4856	0.4982	0.4061	0.7818	0.0291
7	156	2	7	8.8	13.27	-18594.80	-354.64	-0.5797	-36.44	10.05	17.19	9.21	0.3984	0.6125	0.3651	0.5797	0.0322
8	179	2.87	4	3.4	9.8	-10147.65	-245.67	-0.7720	-24.58	6.07	12.27	6.25	0.2143	0.4701	0.2475	0.7720	0.0322
9	199	10.67	5.8	4	7.2	-36072.51	-569.84	-0.9993	-39.04	13.24	11.79	14.01	0.4924	0.4106	0.6279	0.9993	0.0304
10	217	5.47	8.2	7	8.93	-29266.39	-483.98	-0.9993	-39.20	11.72	15.35	12.13	0.4125	0.5345	0.5040	0.9993	0.0350
11	237	3.73	9.4	8.2	14.13	-32959.41	-536.30	-0.5442	-42.76	11.24	19.91	11.60	0.4219	0.6999	0.4750	0.5442	0.0305
12	259	8.93	8.8	7.6	3.73	-44341.35	-669.41	-1.3760	-41.27	15.18	11.34	14.74	0.5275	0.4296	0.6728	1.3760	0.0291
13	284	11.53	1.6	7	8.93	-19572.55	-364.64	-0.7649	-36.36	16.40	9.48	10.47	0.5783	0.4081	0.4791	0.7649	0.0258
14	308	12.4	2.2	4.6	14.13	-25756.70	-439.82	-0.9893	-41.62	14.94	13.87	12.80	0.5522	0.5910	0.5409	0.9893	0.0193
15	332	13.27	5.2	3.4	5.47	-45273.22	-683.24	-1.2119	-39.42	14.42	9.82	15.18	0.5485	0.3478	0.7088	1.2119	0.0319
16	353	15	9.4	2.8	7.2	-69303.24	-984.68	-1.3470	-48.16	14.97	14.76	18.43	0.5857	0.5121	0.8265	1.3470	0.0282
17	374	3.73	4	3.4	7.2	-12116.93	-271.96	-0.6573	-24.20	6.87	10.17	7.16	0.2396	0.3691	0.2906	0.6573	0.0334
18	399	15	9.4	5.2	8.93	-59520.51	-859.10	-1.5178	-51.56	17.12	16.12	18.32	0.6320	0.5597	0.7881	1.5178	0.0250
19	420	2	3.4	7	8.07	-4975.89	-183.71	-0.5114	-25.52	8.51	10.44	6.56	0.3339	0.3977	0.2440	0.5114	0.0275
20	443	8.07	10	9.4	4.6	-43526.07	-659.49	-1.3462	-43.60	15.77	12.99	14.83	0.5506	0.4638	0.6580	1.3462	0.0319
21	469	13.27	1	4	6.33	-25127.91	-433.19	-0.8874	-31.85	14.92	6.85	10.08	0.5629	0.2988	0.5222	0.8874	0.0275
22	490	3.73	7.6	7	2	-35383.23	-558.79	-1.1877	-29.45	10.14	8.95	10.36	0.3722	0.3543	0.4839	1.1877	0.0340
23	512	2.87	5.2	1.6	7.2	-19901.22	-362.39	-1.2742	-21.49	4.36	11.24	5.89	0.1609	0.3981	0.2905	1.2742	0.0359
24	535	7.2	5.2	5.8	2	-35512.27	-562.40	-1.0302	-29.42	12.06	6.92	10.45	0.4193	0.2654	0.5262	1.0302	0.0318
25	561	4.6	7	5.2	2	-36774.32	-573.40	-1.4212	-28.12	9.34	8.33	10.45	0.3294	0.3288	0.5051	1.4212	0.0347
26	583	5.47	1	1.6	15	-13630.09	-289.17	-0.8062	-26.47	6.71	13.36	6.40	0.2593	0.6066	0.2650	0.8062	0.0250
27	605	2	5.8	4	14.13	-15564.91	-316.61	-0.5654	-29.30	5.78	17.25	6.26	0.2111	0.6485	0.2567	0.5654	0.0299
28	628	4.6	5.8	6.4	2	-28919.88	-478.36	-1.0995	-26.89	10.37	7.39	9.12	0.3726	0.2926	0.4342	1.0995	0.0344
29	655	14.13	8.2	9.4	3.73	-53958.42	-789.83	-1.4474	-46.28	20.16	10.82	15.30	0.6875	0.3949	0.7194	1.4474	0.0294
30	676	13.27	7.6	2.8	3.73	-66892.06	-953.99	-1.3688	-42.29	13.86	10.41	18.03	0.5531	0.3886	0.8558	1.3688	0.0269
31	698	12.4	8.2	7	8.93	-45288.94	-686.56	-0.9631	-48.49	16.82	15.35	16.33	0.5861	0.5345	0.6941	0.9631	0.0288
32	720	2.87	5.2	3.4	5.47	-14740.01	-301.47	-0.9414	-22.97	6.07	9.82	7.09	0.2143	0.3478	0.3137	0.9414	0.0338
33	740	2.87	1.6	4	6.33	-2909.44	-155.77	-0.6603	-19.22	6.66	7.42	5.14	0.2379	0.3061	0.1906	0.6603	0.0312
34	766	12.4	3.4	7	8.07	-26961.83	-458.75	-0.6946	-39.55	16.82	10.44	12.29	0.5861	0.3977	0.5443	0.6946	0.0290
35	789	14.13	5.8	9.4	3.73	-43740.30	-659.16	-1.5873	-41.68	20.16	8.81	12.72	0.6875	0.3128	0.6196	1.5873	0.0294
36	811	13.27	3.4	7	8.07	-29380.42	-487.96	-0.7974	-40.62	17.52	10.44	12.65	0.6213	0.3977	0.5652	0.7974	0.0349
37	839	2	1	4	6.33	-352.24	-126.16	-0.4507	-16.75	5.78	6.85	4.12	0.2111	0.2988	0.1525	0.4507	0.0278
38	862	2	7	8.8	14.13	-19733.89	-368.57	-0.6149	-37.63	10.05	18.34	9.23	0.3984	0.6607	0.3716	0.6149	0.0340
39	883	12.4	2.2	4.6	13.27	-25346.16	-434.83	-0.9742	-40.61	14.94	13.24	12.43	0.5522	0.5602	0.5388	0.9742	0.0189
40	917	2.87	5.8	6.4	2	-23228.62	-410.73	-0.7696	-24.17	8.85	7.39	7.92	0.3263	0.2926	0.3579	0.7696	0.0282
41	939	3.73	1	2.8	2	-8011.13	-218.20	-0.8148	-13.24	6.34	2.96	3.94	0.2248	0.1095	0.1847	0.8148	0.0346
42	963	2	7	8.8	12.4	-18284.74	-351.52	-0.5175	-36.00	10.05	16.79	9.15	0.3984	0.5923	0.3586	0.5175	0.0301
43	987	4.6	7	6.4	2.87	-28536.60	-474.96	-0.9853	-29.94	10.37	9.16	10.41	0.3726	0.3460	0.4918	0.9853	0.0331
44	1008	3.73	10	8.2	14.13	-35777.19	-570.00	-0.6911	-43.28	11.24	20.36	11.68	0.4219	0.6869	0.4782	0.6911	0.0269
45	1034	2.87	7	5.2	2	-31265.32	-506.79	-1.1920	-25.34	7.72	8.33	9.29	0.2791	0.3288	0.4272	1.1920	0.0325

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
46	1058	3.73	7	8.8	13.27	-21505.21	-392.11	-0.5196	-39.45	11.69	17.19	10.57	0.4483	0.6125	0.4040	0.5196	0.0355
47	1080	15	1.6	7	8.93	-28784.57	-479.89	-0.8412	-40.36	18.95	9.48	11.93	0.6533	0.4081	0.5615	0.8412	0.0250
48	1102	11.53	1	4	6.33	-20855.00	-380.64	-0.7803	-30.08	13.75	6.85	9.48	0.5067	0.2988	0.4691	0.7803	0.0282
49	1125	4.6	2.2	5.2	2	-12535.03	-271.43	-1.1182	-18.75	9.34	4.09	5.31	0.3294	0.1446	0.2508	1.1182	0.0311
50	1149	4.6	7	5.2	14.13	-25536.54	-437.90	-0.9221	-37.65	9.34	18.34	9.96	0.3294	0.6607	0.4021	0.9221	0.0381
51	1174	2	5.8	4	2	-22897.47	-403.73	-0.9939	-20.39	5.78	7.39	7.21	0.2111	0.2926	0.3337	0.9939	0.0341
52	1194	4.6	5.8	2.8	2	-40691.63	-624.06	-1.3231	-25.10	7.14	7.39	10.56	0.2631	0.2926	0.5238	1.3231	0.0303
53	1221	3.73	2.2	6.4	2.87	-7587.80	-209.85	-1.0523	-19.84	9.64	4.90	5.30	0.3460	0.1732	0.2143	1.0523	0.0348
54	1242	15	1.6	4	6.33	-31691.90	-517.39	-0.7770	-34.92	15.86	7.42	11.64	0.5963	0.3061	0.5708	0.7770	0.0298
55	1263	2.87	1.6	7	8.93	-1645.62	-143.49	-0.3690	-25.09	9.39	9.48	6.22	0.3508	0.4081	0.2306	0.3690	0.0277
56	1283	3.73	5.8	9.4	3.73	-17587.63	-336.91	-0.9793	-29.92	12.29	8.81	8.83	0.4631	0.3128	0.3514	0.9793	0.0301
57	1305	12.4	5.8	4	2	-63096.85	-905.14	-1.4465	-36.86	14.39	7.39	15.08	0.5288	0.2926	0.7852	1.4465	0.0316
58	1324	2	5.8	8.8	14.13	-14826.77	-308.09	-0.5018	-36.15	10.05	17.25	8.85	0.3984	0.6485	0.3390	0.5018	0.0311
59	1344	3.73	5.8	4	12.4	-18558.89	-350.89	-0.8422	-31.33	7.44	15.88	8.01	0.2562	0.5787	0.3222	0.8422	0.0350
60	1363	2	1.6	7.6	9.8	-199.99	-126.16	-0.2972	-25.08	9.05	10.08	5.95	0.3600	0.4327	0.2235	0.2972	0.0259
61	1384	2	4.6	4	6.33	-9523.11	-236.26	-0.8937	-21.85	5.78	10.05	6.02	0.2111	0.3589	0.2470	0.8937	0.0311
62	1410	2.87	1	4	6.33	-1322.70	-136.69	-0.5861	-18.23	6.66	6.85	4.73	0.2379	0.2988	0.1796	0.5861	0.0306
63	1432	2.87	7	4	2	-33146.26	-529.00	-1.3129	-24.24	6.66	8.33	9.25	0.2379	0.3288	0.4289	1.3129	0.0305
64	1453	2.87	1.6	4	2	-6683.57	-199.32	-0.9835	-14.16	6.66	3.53	3.97	0.2379	0.1245	0.1689	0.9835	0.0364
65	1475	2.87	1	2.8	6.33	-2464.06	-150.03	-0.6703	-16.75	5.49	6.85	4.41	0.1936	0.2988	0.1701	0.6703	0.0300
66	1497	2	5.8	4	13.27	-15026.40	-308.45	-0.6738	-28.46	5.78	16.44	6.24	0.2111	0.6096	0.2543	0.6738	0.0345
67	1520	2.87	1.6	4	7.2	-2566.83	-153.08	-0.5312	-19.94	6.66	8.10	5.19	0.2379	0.3341	0.1892	0.5312	0.0292
68	1540	2	1	4	2.87	-840.08	-129.74	-0.6555	-12.85	5.78	3.75	3.31	0.2111	0.1467	0.1251	0.6555	0.0327
69	1563	2	2.2	6.4	2.87	-3817.25	-165.36	-0.8087	-17.33	7.98	4.90	4.45	0.3130	0.1732	0.1651	0.8087	0.0334
70	1592	4.6	5.8	4	5.47	-20805.00	-375.10	-1.1698	-27.81	8.20	10.34	9.26	0.2840	0.3602	0.4035	1.1698	0.0313
71	1613	3.73	5.8	9.4	2.87	-19551.88	-363.78	-0.8134	-28.95	12.29	8.14	8.52	0.4631	0.3030	0.3588	0.8134	0.0322
72	1634	4.6	7	6.4	3.73	-25492.85	-434.42	-1.1556	-30.67	10.37	9.93	10.36	0.3726	0.3672	0.4558	1.1556	0.0370
73	1657	2.87	1.6	7	9.8	-1698.99	-143.56	-0.4168	-25.82	9.39	10.08	6.36	0.3508	0.4327	0.2401	0.4168	0.0267
74	1680	2	1	2.8	2	-2851.78	-155.00	-0.6651	-10.56	4.68	2.96	2.92	0.1657	0.1095	0.1173	0.6651	0.0318
75	1704	4.6	1	4	2.87	-6619.42	-198.65	-0.9720	-16.55	8.20	3.75	4.59	0.2840	0.1467	0.2131	0.9720	0.0306
76	1731	2	1.6	4	7.2	-769.95	-132.12	-0.3958	-18.42	5.78	8.10	4.54	0.2111	0.3341	0.1637	0.3958	0.0264
77	1757	3.73	1.6	7.6	9.8	-3420.49	-162.02	-0.6750	-27.88	10.74	10.08	7.06	0.3994	0.4327	0.2656	0.6750	0.0286
78	1783	2.87	1	2.8	2.87	-4128.51	-168.45	-0.8751	-12.99	5.49	3.75	3.75	0.1936	0.1467	0.1572	0.8751	0.0344
79	1809	2	1.6	7	8.93	-270.80	-126.27	-0.3597	-23.68	8.51	9.48	5.69	0.3339	0.4081	0.2189	0.3597	0.0293
80	1838	3.73	2.2	6.4	5.47	-5351.27	-185.42	-0.7513	-23.13	9.64	7.23	6.26	0.3460	0.2788	0.2378	0.7513	0.0339
81	1858	4.6	5.8	4	2.87	-27157.10	-454.41	-1.2360	-25.98	8.20	8.14	9.65	0.2840	0.3030	0.4435	1.2360	0.0331
82	1880	2	1	4	14.13	-253.47	-126.16	-0.3510	-23.16	5.78	12.56	4.81	0.2111	0.5763	0.1774	0.3510	0.0239
83	1897	5.47	1	4	6.33	-6488.80	-201.23	-0.6359	-22.11	8.96	6.85	6.30	0.3140	0.2988	0.2691	0.6359	0.0296
84	1916	4.6	1	4	5.47	-5034.49	-180.22	-0.8476	-19.85	8.20	6.10	5.55	0.2840	0.2604	0.2327	0.8476	0.0318
85	1936	2.87	1.6	7	2.87	-3580.70	-162.47	-0.8011	-18.05	9.39	4.32	4.34	0.3508	0.1574	0.1676	0.8011	0.0333
86	1956	2	4	4	7.2	-7301.57	-211.55	-0.6311	-21.65	5.78	10.17	5.69	0.2111	0.3691	0.2256	0.6311	0.0344
87	1978	2.87	1	7	8.93	-269.27	-126.16	-0.3666	-24.10	9.39	8.88	5.83	0.3508	0.4027	0.2241	0.3666	0.0243
88	1996	2	1	5.2	6.33	-337.13	-126.16	-0.4354	-18.28	6.95	6.85	4.48	0.2626	0.2988	0.1654	0.4354	0.0279
89	2018	2	2.2	4	6.33	-2452.37	-151.30	-0.5575	-18.49	5.78	7.93	4.77	0.2111	0.3100	0.1749	0.5575	0.0310
90	2039	2	1.6	4	6.33	-964.26	-132.94	-0.5250	-17.65	5.78	7.42	4.44	0.2111	0.3061	0.1600	0.5250	0.0302

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
91	2059	2	1.6	7	2.87	-1901.68	-141.40	-0.7918	-16.77	8.51	4.32	3.94	0.3339	0.1574	0.1439	0.7918	0.0348
92	2085	2.87	1.6	7.6	2.87	-3473.45	-161.42	-0.7766	-18.54	9.88	4.32	4.34	0.3721	0.1574	0.1670	0.7766	0.0299
93	2111	2.87	1	2.8	9.8	-2104.23	-146.67	-0.5767	-19.64	5.49	9.47	4.67	0.1936	0.4217	0.1740	0.5767	0.0265
94	2132	2	1.6	8.8	6.33	-363.81	-126.16	-0.4614	-22.87	10.05	7.42	5.40	0.3984	0.3061	0.1991	0.4614	0.0327
95	2153	2	2.2	2.2	2.87	-7208.17	-206.61	-0.9303	-13.06	4.11	4.90	4.05	0.1443	0.1732	0.1646	0.9303	0.0310
96	2176	2.87	1.6	8.8	6.33	-1683.77	-144.02	-0.3657	-24.22	10.93	7.42	5.86	0.4291	0.3061	0.2216	0.3657	0.0260
97	2202	8.93	1	4	6.33	-14329.15	-299.87	-0.6572	-26.89	11.83	6.85	8.22	0.4362	0.2988	0.3932	0.6572	0.0300
98	2227	2	1.6	7	9.8	-315.98	-126.23	-0.4074	-24.39	8.51	10.08	5.80	0.3339	0.4327	0.2148	0.4074	0.0324
99	2249	2	1.6	5.8	8.93	-247.37	-127.39	-0.2474	-22.29	7.45	9.48	5.36	0.2886	0.4081	0.1977	0.2474	0.0216
100	2271	2.87	1.6	7.6	9.8	-1595.42	-142.56	-0.3923	-26.41	9.88	10.08	6.45	0.3721	0.4327	0.2387	0.3923	0.0313
101	2295	2.87	6.4	7.6	9.8	-16565.21	-328.79	-0.5998	-33.42	9.88	14.50	9.05	0.3721	0.5177	0.3510	0.5998	0.0327
102	2317	3.73	1	4	6.33	-2929.29	-157.40	-0.5508	-19.57	7.44	6.85	5.28	0.2562	0.2988	0.2060	0.5508	0.0292
103	2339	2	1	4	13.27	-333.30	-126.16	-0.4309	-22.55	5.78	12.01	4.76	0.2111	0.5519	0.1724	0.4309	0.0292
104	2360	2.87	1.6	8.8	5.47	-1801.65	-144.89	-0.4143	-23.13	10.93	6.66	5.54	0.4291	0.2649	0.2079	0.4143	0.0263
105	2388	2	3.4	4	6.33	-5712.04	-191.32	-0.6448	-20.16	5.78	8.98	5.40	0.2111	0.3323	0.2060	0.6448	0.0325
106	2414	2	1	4	5.47	-432.51	-126.16	-0.5311	-15.86	5.78	6.10	3.97	0.2111	0.2604	0.1457	0.5311	0.0288
107	2435	2	5.8	4	6.33	-13228.73	-284.30	-0.7912	-23.38	5.78	10.99	6.61	0.2111	0.3807	0.2846	0.7912	0.0317
108	2458	2	1.6	10	9.8	-237.53	-127.24	-0.2488	-27.74	11.12	10.08	6.54	0.4617	0.4327	0.2525	0.2488	0.0221
109	2480	2	1	4	9.8	-259.08	-126.16	-0.3571	-19.77	5.78	9.47	4.51	0.2111	0.4217	0.1663	0.3571	0.0245
110	2503	2	1.6	7.6	6.33	-273.62	-126.56	-0.3398	-21.75	9.05	7.42	5.28	0.3600	0.3061	0.1948	0.3398	0.0248
111	2527	2.87	1.6	2.8	9.8	-3808.39	-166.59	-0.7019	-20.51	5.49	10.08	4.94	0.1936	0.4327	0.1838	0.7019	0.0306
112	2552	2	1.6	7	13.27	-653.53	-130.72	-0.3889	-27.43	8.51	12.64	6.28	0.3339	0.5585	0.2379	0.3889	0.0257
113	2576	2	4	7.6	9.8	-6857.60	-208.08	-0.4609	-28.55	9.05	12.27	7.23	0.3600	0.4701	0.2667	0.4609	0.0299
114	2598	2	1.6	9.4	9.8	-162.84	-126.61	-0.2242	-27.10	10.63	10.08	6.39	0.4427	0.4327	0.2455	0.2242	0.0215
115	2620	2	1	3.4	5.47	-415.98	-126.16	-0.5147	-15.19	5.26	6.10	3.82	0.1887	0.2604	0.1398	0.5147	0.0270
116	2641	2	1	8.8	5.47	-369.67	-126.16	-0.4675	-20.88	10.05	6.10	4.73	0.3984	0.2604	0.1820	0.4675	0.0323
117	2667	3.73	1	4	13.27	-2809.27	-156.13	-0.5309	-25.58	7.44	12.01	6.13	0.2562	0.5519	0.2263	0.5309	0.0300
118	2686	2	1	4	8.07	-359.20	-126.16	-0.4574	-18.34	5.78	8.23	4.33	0.2111	0.3657	0.1593	0.4574	0.0270
119	2706	2	1	8.8	13.27	-270.62	-126.16	-0.3673	-28.64	10.05	12.01	6.58	0.3984	0.5519	0.2591	0.3673	0.0241
120	2728	2	1	5.2	13.27	-318.22	-126.16	-0.4156	-24.29	6.95	12.01	5.33	0.2626	0.5519	0.1991	0.4156	0.0266
121	2758	2	1.6	6.4	2.87	-2005.73	-143.24	-0.7507	-16.25	7.98	4.32	3.95	0.3130	0.1574	0.1446	0.7507	0.0334
122	2780	8.93	1.6	9.4	9.8	-13677.81	-288.90	-0.8737	-36.09	16.33	10.08	9.69	0.5619	0.4327	0.4059	0.8737	0.0265
123	2802	2	1	2.2	9.8	-816.23	-133.12	-0.3626	-17.25	4.11	9.47	3.67	0.1443	0.4217	0.1335	0.3626	0.0253
124	2825	2	4	8.8	6.33	-6588.83	-202.90	-0.6034	-26.61	10.05	9.53	7.03	0.3984	0.3464	0.2616	0.6034	0.0314
125	2848	2	2.8	7	9.8	-3492.87	-165.11	-0.5024	-26.18	8.51	11.25	6.41	0.3339	0.4506	0.2320	0.5024	0.0294
126	2872	5.47	1.6	7.6	6.33	-6621.14	-202.83	-0.6407	-26.68	12.29	7.42	6.96	0.4280	0.3061	0.2825	0.6407	0.0351
127	2895	2	6.4	7.6	6.33	-14335.35	-296.74	-0.9110	-28.74	9.05	11.55	8.14	0.3600	0.4043	0.3242	0.9110	0.0289
128	2915	2	1.6	8.8	5.47	-492.76	-127.17	-0.5100	-21.86	10.05	6.66	5.15	0.3984	0.2649	0.1877	0.5100	0.0300
129	2939	2	1	8.8	6.33	-289.42	-126.16	-0.3871	-21.90	10.05	6.85	5.00	0.3984	0.2988	0.1959	0.3871	0.0270
130	2974	2	1	5.2	5.47	-417.38	-126.16	-0.5158	-17.27	6.95	6.10	4.21	0.2626	0.2604	0.1582	0.5158	0.0318
131	3012	2	1	5.2	2.87	-541.32	-126.16	-0.6402	-14.08	6.95	3.75	3.37	0.2626	0.1467	0.1277	0.6402	0.0332
132	3037	3.73	1.6	7.6	6.33	-3502.09	-162.52	-0.7176	-24.31	10.74	7.42	6.15	0.3994	0.3061	0.2343	0.7176	0.0314
133	3079	2	1.6	4	5.47	-1136.45	-134.50	-0.5736	-16.77	5.78	6.66	4.33	0.2111	0.2649	0.1556	0.5736	0.0312
134	3101	2	1	4.6	5.47	-439.14	-126.16	-0.5376	-16.57	6.35	6.10	4.12	0.2351	0.2604	0.1529	0.5376	0.0331
135	3123	3.73	4	7	9.8	-10132.84	-246.24	-0.7114	-30.72	10.14	12.27	8.31	0.3722	0.4701	0.3128	0.7114	0.0327

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
136	3150	2	1	4.6	6.33	-358.88	-126.16	-0.4572	-17.48	6.35	6.85	4.29	0.2351	0.2988	0.1579	0.4572	0.0274
137	3171	2	3.4	5.2	5.47	-5565.76	-189.84	-0.6161	-21.05	6.95	8.29	5.81	0.2626	0.2994	0.2164	0.6161	0.0291
138	3199	8.93	1	4	2	-18816.11	-353.78	-0.8721	-20.48	11.83	2.96	5.70	0.4362	0.1095	0.3520	0.8721	0.0347
139	3221	3.73	1	4	2.87	-4656.81	-176.62	-0.7556	-15.43	7.44	3.75	4.23	0.2562	0.1467	0.1831	0.7556	0.0362
140	3244	2	1	10	5.47	-199.89	-126.16	-0.2976	-22.05	11.12	6.10	4.83	0.4617	0.2604	0.1882	0.2976	0.0258
141	3267	2	3.4	5.2	2.87	-8263.40	-219.13	-0.9928	-18.38	6.95	6.05	5.38	0.2626	0.2138	0.2102	0.9928	0.0314
142	3287	2	1	4	2	-1814.09	-141.90	-0.6657	-11.67	5.78	2.96	2.93	0.2111	0.1095	0.1151	0.6657	0.0343
143	3308	2	1.6	7	6.33	-416.62	-126.97	-0.4500	-21.12	8.51	7.42	5.19	0.3339	0.3061	0.1913	0.4500	0.0324
144	3336	2	1	5.2	9.8	-243.98	-126.16	-0.3418	-21.38	6.95	9.47	4.95	0.2626	0.4217	0.1857	0.3418	0.0249
145	3358	8.93	1	5.2	6.33	-13324.23	-283.99	-0.9109	-27.91	12.97	6.85	8.09	0.4587	0.2988	0.3813	0.9109	0.0254
146	3379	2	1	8.8	2.87	-493.55	-126.16	-0.5919	-17.30	10.05	3.75	3.49	0.3984	0.1467	0.1333	0.5919	0.0341
147	3402	2	1	10	6.33	-119.65	-126.16	-0.2172	-23.12	11.12	6.85	5.15	0.4617	0.2988	0.2008	0.2172	0.0242
148	3424	3.73	1	5.2	6.33	-2110.71	-148.89	-0.4070	-20.76	8.49	6.85	5.42	0.2957	0.2988	0.2142	0.4070	0.0259
149	3445	2	1	4.6	8.07	-365.84	-126.16	-0.4640	-19.15	6.35	8.23	4.57	0.2351	0.3657	0.1700	0.4640	0.0300
150	3467	2	1	7	6.33	-277.81	-126.16	-0.3758	-20.11	8.51	6.85	4.75	0.3339	0.2988	0.1818	0.3758	0.0258
151	3491	2	2.2	5.2	6.33	-2112.00	-147.20	-0.5422	-20.04	6.95	7.93	5.15	0.2626	0.3100	0.1862	0.5422	0.0303
152	3518	2	1	3.4	6.33	-335.71	-126.16	-0.4343	-16.07	5.26	6.85	3.95	0.1887	0.2988	0.1461	0.4343	0.0297
153	3543	2.87	1	5.2	2.87	-1954.64	-143.61	-0.6699	-15.21	7.72	3.75	3.73	0.2791	0.1467	0.1517	0.6699	0.0316
154	3569	2	1	9.4	6.33	-94.92	-126.16	-0.1925	-22.59	10.63	6.85	5.11	0.4427	0.2988	0.2021	0.1925	0.0210
155	3594	2	1.6	10	6.33	-195.10	-126.17	-0.2914	-24.08	11.12	7.42	5.54	0.4617	0.3061	0.2071	0.2914	0.0245
156	3615	2	5.8	4	5.47	-13673.91	-289.13	-0.8533	-22.84	5.78	10.34	6.71	0.2111	0.3602	0.2942	0.8533	0.0314
157	3638	2	5.8	5.2	9.8	-13497.99	-286.72	-0.8681	-27.94	6.95	13.88	7.10	0.2626	0.5025	0.2857	0.8681	0.0301
158	3660	2	1	5.2	11.53	-271.54	-126.16	-0.3691	-22.86	6.95	10.70	5.21	0.2626	0.4826	0.1943	0.3691	0.0257
159	3679	5.47	1	5.2	9.8	-5259.64	-186.45	-0.5774	-26.70	10.09	9.47	7.14	0.3574	0.4217	0.2886	0.5774	0.0263
160	3706	2	3.4	4.6	6.33	-5474.48	-188.24	-0.6513	-20.95	6.35	8.98	5.63	0.2351	0.3323	0.2131	0.6513	0.0325
161	3728	2	1	7.6	5.47	-247.94	-126.16	-0.3460	-19.74	9.05	6.10	4.59	0.3600	0.2604	0.1754	0.3460	0.0250
162	3748	2	1	2.2	6.33	-1052.88	-134.93	-0.4563	-14.47	4.11	6.85	3.51	0.1443	0.2988	0.1290	0.4563	0.0284
163	3769	2	1	10	11.53	-54.16	-126.16	-0.1509	-28.35	11.12	10.70	6.53	0.4617	0.4826	0.2622	0.1509	0.0196
164	3794	2	1	5.2	4.6	-399.94	-126.16	-0.4985	-16.26	6.95	5.30	4.00	0.2626	0.2204	0.1499	0.4985	0.0327
165	3816	2	5.8	8.8	6.33	-12069.01	-270.46	-0.7276	-29.14	10.05	10.99	8.11	0.3984	0.3807	0.3128	0.7276	0.0358
166	3838	2	1	8.8	4.6	-352.20	-126.16	-0.4502	-19.69	10.05	5.30	4.34	0.3984	0.2204	0.1642	0.4502	0.0300
167	3860	2	1.6	9.4	6.33	-208.97	-126.66	-0.2668	-23.57	10.63	7.42	5.52	0.4427	0.3061	0.2094	0.2668	0.0222
168	3885	2	1.6	10	5.47	-384.55	-127.95	-0.3401	-23.06	11.12	6.66	5.29	0.4617	0.2649	0.1957	0.3401	0.0248
169	3909	2	1.6	8.8	2.87	-1877.23	-140.95	-0.8031	-18.40	10.05	4.32	4.03	0.3984	0.1574	0.1471	0.8031	0.0356
170	3937	2	1.6	10	11.53	-472.41	-129.76	-0.2834	-29.54	11.12	11.48	6.94	0.4617	0.5048	0.2677	0.2834	0.0241
171	3960	2	5.8	7.6	5.47	-12353.54	-274.80	-0.6682	-27.18	9.05	10.34	7.79	0.3600	0.3602	0.3108	0.6682	0.0283
172	3983	2	2.8	10	5.47	-3205.53	-163.07	-0.3769	-24.98	11.12	7.76	6.10	0.4617	0.2884	0.2252	0.3769	0.0246
173	4004	2	1	7.6	6.33	-167.69	-126.16	-0.2656	-20.75	9.05	6.85	4.85	0.3600	0.2988	0.1894	0.2656	0.0248
174	4028	3.73	1	9.4	6.33	-1780.20	-145.50	-0.3446	-24.95	12.29	6.85	5.81	0.4631	0.2988	0.2316	0.3446	0.0263
175	4049	2	5.8	5.2	6.33	-12845.14	-279.65	-0.7759	-25.18	6.95	10.99	7.24	0.2626	0.3807	0.2991	0.7759	0.0300
176	4077	8.93	1	8.8	2.87	-13371.26	-287.49	-0.6805	-24.78	15.80	3.75	5.22	0.5503	0.1467	0.2920	0.6805	0.0347
177	4097	2	1	10	8.07	-126.67	-126.16	-0.2239	-25.07	11.12	8.23	5.73	0.4617	0.3657	0.2303	0.2239	0.0231
178	4119	2	1	7.6	11.53	-102.15	-126.16	-0.1993	-25.72	9.05	10.70	5.97	0.3600	0.4826	0.2327	0.1993	0.0212
179	4141	5.47	1	8.8	6.33	-4992.56	-184.80	-0.4411	-26.59	13.22	6.85	6.52	0.4701	0.2988	0.2741	0.4411	0.0285
180	4166	2	1	5.8	6.33	-165.33	-126.16	-0.2635	-18.88	7.45	6.85	4.58	0.2886	0.2988	0.1742	0.2635	0.0241

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
181	4191	2	1	5.2	10.67	-274.25	-126.16	-0.3719	-22.26	6.95	10.21	5.10	0.2626	0.4690	0.1898	0.3719	0.0259
182	4214	3.73	1.6	10	6.33	-3353.27	-162.83	-0.5439	-26.45	12.69	7.42	6.34	0.4887	0.3061	0.2494	0.5439	0.0318
183	4236	2	5.8	10	6.33	-11886.95	-270.31	-0.5577	-30.44	11.12	10.99	8.33	0.4617	0.3807	0.3212	0.5577	0.0313
184	4263	3.73	1	5.2	4.6	-2434.44	-152.18	-0.4701	-18.64	8.49	5.30	4.85	0.2957	0.2204	0.1971	0.4701	0.0315
185	4287	2.87	1	5.2	6.33	-679.92	-130.11	-0.4651	-19.51	7.72	6.85	4.94	0.2791	0.2988	0.1876	0.4651	0.0313
186	4314	2.87	1	10	4.6	-501.14	-130.01	-0.2935	-21.98	11.96	5.30	4.72	0.4748	0.2204	0.1856	0.2935	0.0261
187	4337	2	1	9.4	8.07	-101.93	-126.16	-0.1992	-24.53	10.63	8.23	5.67	0.4427	0.3657	0.2266	0.1992	0.0216
188	4360	2	2.2	5.2	11.53	-2203.55	-150.68	-0.3569	-24.54	6.95	11.89	5.70	0.2626	0.5001	0.2071	0.3569	0.0258
189	4382	2	1	10	4.6	-182.43	-126.16	-0.2803	-20.84	11.12	5.30	4.42	0.4617	0.2204	0.1694	0.2803	0.0253
190	4405	2.87	1	9.4	6.33	-443.17	-127.72	-0.4166	-23.73	11.38	6.85	5.50	0.4442	0.2988	0.2174	0.4166	0.0273
191	4430	5.47	6.4	10	6.33	-21493.77	-384.25	-1.1320	-36.66	14.11	11.55	11.00	0.4966	0.4043	0.4286	1.1320	0.0301
192	4456	8.93	1.6	10	6.33	-13648.76	-294.09	-0.4340	-32.60	16.97	7.42	8.21	0.5795	0.3061	0.3692	0.4340	0.0295
193	4478	2.87	1	5.2	8.07	-571.59	-128.65	-0.4718	-21.23	7.72	8.23	5.28	0.2791	0.3657	0.1990	0.4718	0.0281
194	4499	2	1	8.8	8.07	-296.43	-126.16	-0.3938	-23.85	10.05	8.23	5.57	0.3984	0.3657	0.2181	0.3938	0.0283
195	4521	3.73	1	5.2	11.53	-1872.64	-146.71	-0.3407	-25.51	8.49	10.70	6.32	0.2957	0.4826	0.2343	0.3407	0.0254
196	4548	2	1	5.2	3.73	-425.81	-126.16	-0.5245	-15.22	6.95	4.57	3.70	0.2626	0.1863	0.1399	0.5245	0.0313
197	4567	2	5.8	10	11.53	-13123.39	-288.19	-0.3755	-35.13	11.12	14.98	9.03	0.4617	0.5401	0.3488	0.3755	0.0315
198	4590	10.67	1	5.2	11.53	-16468.15	-326.15	-0.7120	-35.24	14.21	10.70	10.34	0.5040	0.4826	0.4536	0.7120	0.0277
199	4613	2	1	7	8.07	-284.79	-126.16	-0.3825	-21.89	8.51	8.23	5.16	0.3339	0.3657	0.1984	0.3825	0.0269
200	4639	2	1	5.2	8.07	-344.10	-126.16	-0.4421	-19.92	6.95	8.23	4.75	0.2626	0.3657	0.1771	0.4421	0.0269
201	4665	2	3.4	9.4	6.33	-4702.30	-181.83	-0.3866	-26.33	10.63	8.98	6.73	0.4427	0.3323	0.2452	0.3866	0.0257
202	4686	2	1.6	10	8.07	-130.23	-126.16	-0.2273	-25.99	11.12	8.79	6.09	0.4617	0.3718	0.2302	0.2273	0.0214
203	4707	2	5.8	9.4	6.33	-11765.42	-269.09	-0.5330	-29.75	10.63	10.99	8.13	0.4427	0.3807	0.3150	0.5330	0.0300
204	4730	8.93	1	2.2	8.07	-20672.97	-375.46	-1.0090	-27.28	10.17	8.23	8.89	0.4077	0.3657	0.4129	1.0090	0.0246
205	4754	2	1.6	9.4	8.07	-105.49	-126.16	-0.2027	-25.42	10.63	8.79	6.00	0.4427	0.3718	0.2258	0.2027	0.0211
206	4776	2	4	10	6.33	-6349.51	-202.02	-0.4334	-27.80	11.12	9.53	7.15	0.4617	0.3464	0.2620	0.4334	0.0283
207	4798	5.47	1	10	8.07	-4916.14	-183.18	-0.4932	-29.54	14.11	8.23	7.20	0.4966	0.3657	0.2975	0.4932	0.0313
208	4822	2	1.6	7	8.07	-288.36	-126.16	-0.3859	-22.83	8.51	8.79	5.53	0.3339	0.3718	0.2044	0.3859	0.0271
209	4844	2	3.4	10	8.07	-4805.54	-183.55	-0.3528	-28.83	11.12	10.44	7.26	0.4617	0.3977	0.2725	0.3528	0.0251
210	4868	2	6.4	10	8.07	-14222.82	-299.91	-0.5469	-33.07	11.12	12.94	9.01	0.4617	0.4517	0.3584	0.5469	0.0271
211	4892	8.93	1	10	6.33	-12067.19	-275.08	-0.3597	-31.40	16.97	6.85	7.59	0.5795	0.2988	0.3605	0.3597	0.0280
212	4915	2.87	1.6	10	8.07	-1541.14	-143.79	-0.2405	-27.28	11.96	8.79	6.54	0.4748	0.3718	0.2457	0.2405	0.0238
213	4940	3.73	1	10	6.33	-1989.08	-146.55	-0.4697	-25.45	12.69	6.85	5.91	0.4887	0.2988	0.2383	0.4697	0.0309
214	4964	3.73	1	9.4	8.07	-1764.87	-145.22	-0.3513	-27.02	12.29	8.23	6.50	0.4631	0.3657	0.2585	0.3513	0.0299
215	4984	5.47	1.6	9.4	8.07	-6445.22	-200.90	-0.6173	-30.15	13.68	8.79	7.68	0.4862	0.3718	0.3050	0.6173	0.0283
216	5010	3.73	1.6	9.4	6.33	-3122.54	-161.49	-0.4189	-25.92	12.29	7.42	6.21	0.4631	0.3061	0.2409	0.4189	0.0309
217	5033	2	1	8.2	8.07	-213.28	-126.16	-0.3108	-23.12	9.50	8.23	5.40	0.3827	0.3657	0.2135	0.3108	0.0275
218	5057	2.87	1	10	8.07	-218.36	-127.15	-0.2371	-26.32	11.96	8.23	6.14	0.4748	0.3657	0.2429	0.2371	0.0215
219	5080	5.47	1	10	6.33	-5085.06	-185.40	-0.4865	-27.49	14.11	6.85	6.53	0.4966	0.2988	0.2756	0.4865	0.0290
220	5101	2	1	9.4	4.6	-157.70	-126.16	-0.2556	-20.34	10.63	5.30	4.41	0.4427	0.2204	0.1694	0.2556	0.0231
221	5123	2.87	3.4	10	8.07	-6243.39	-201.53	-0.3660	-30.13	11.96	10.44	7.74	0.4748	0.3977	0.2820	0.3660	0.0271
222	5145	2	1	10	15	-86.54	-126.16	-0.1827	-31.71	11.12	13.36	7.23	0.4617	0.6066	0.2907	0.1827	0.0212
223	5168	2	1.6	9.4	4.6	-577.23	-129.95	-0.3745	-21.45	10.63	5.93	4.89	0.4427	0.2326	0.1807	0.3745	0.0279
224	5190	2	1.6	10	15	-747.49	-134.22	-0.2048	-32.49	11.12	13.85	7.52	0.4617	0.6042	0.2964	0.2048	0.0221
225	5216	8.93	1.6	10	8.07	-13525.63	-293.34	-0.3699	-34.90	16.97	8.79	9.14	0.5795	0.3718	0.3977	0.3699	0.0241

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
226	5242	2.87	6.4	9.4	8.07	-15875.91	-318.24	-0.7464	-33.72	11.38	12.94	9.41	0.4442	0.4517	0.3658	0.7464	0.0317
227	5268	2	4	9.4	8.07	-6421.70	-204.17	-0.3347	-29.05	10.63	10.94	7.48	0.4427	0.4014	0.2843	0.3347	0.0261
228	5293	2	1	10	7.2	-139.01	-126.16	-0.2364	-24.19	11.12	7.57	5.50	0.4617	0.3347	0.2209	0.2364	0.0254
229	5318	2	1.6	10	2.87	-1712.68	-141.02	-0.6332	-19.49	11.12	4.32	4.05	0.4617	0.1574	0.1484	0.6332	0.0303
230	5338	2	1	10	9.8	-26.59	-126.16	-0.1235	-26.75	11.12	9.47	6.16	0.4617	0.4217	0.2473	0.1235	0.0165
231	5361	2	1.6	10	4.6	-612.59	-130.09	-0.3992	-21.98	11.12	5.93	4.93	0.4617	0.2326	0.1815	0.3992	0.0282
232	5386	2	1	10	2.87	-323.77	-126.16	-0.4220	-18.39	11.12	3.75	3.52	0.4617	0.1467	0.1348	0.4220	0.0273
233	5406	2	1	10	12.4	-118.11	-126.16	-0.2147	-29.39	11.12	11.53	6.74	0.4617	0.5267	0.2731	0.2147	0.0205
234	5428	5.47	1.6	10	5.47	-6679.57	-203.96	-0.6094	-27.37	14.11	6.66	6.61	0.4966	0.2649	0.2682	0.6094	0.0311
235	5449	2	1	7.6	4.6	-230.48	-126.16	-0.3287	-18.62	9.05	5.30	4.26	0.3600	0.2204	0.1606	0.3287	0.0246
236	5472	2	1	9.4	2.87	-299.04	-126.16	-0.3973	-17.87	10.63	3.75	3.49	0.4427	0.1467	0.1326	0.3973	0.0274
237	5494	3.73	1	10	7.2	-1888.20	-145.04	-0.4889	-26.45	12.69	7.57	6.19	0.4887	0.3347	0.2478	0.4889	0.0293
238	5517	2.87	1.6	10	5.47	-1832.13	-146.04	-0.3533	-24.28	11.96	6.66	5.66	0.4748	0.2649	0.2100	0.3533	0.0246
239	5539	2	1.6	10	13.27	-540.33	-131.29	-0.2302	-30.97	11.12	12.64	7.20	0.4617	0.5585	0.2803	0.2302	0.0218
240	5561	2	1.6	10	7.2	-65.08	-126.16	-0.1623	-25.04	11.12	8.10	5.82	0.4617	0.3341	0.2179	0.1623	0.0196
241	5588	2	1	9.4	15	-61.77	-126.16	-0.1581	-30.98	10.63	13.36	6.99	0.4427	0.6066	0.2829	0.1581	0.0189
242	5611	2	5.8	7.6	11.53	-13071.24	-286.93	-0.4239	-32.10	9.05	14.98	8.06	0.3600	0.5401	0.3185	0.4239	0.0261
243	5632	2	3.4	5.2	6.33	-5395.89	-187.53	-0.6294	-21.83	6.95	8.98	5.90	0.2626	0.3323	0.2230	0.6294	0.0324
244	5654	2	1	7.6	8.07	-174.69	-126.16	-0.2723	-22.60	9.05	8.23	5.32	0.3600	0.3657	0.2071	0.2723	0.0233
245	5677	2	1	7.6	2	-818.91	-131.68	-0.4805	-14.95	9.05	2.96	2.94	0.3600	0.1095	0.1121	0.4805	0.0302
246	5704	2	1.6	7.6	11.53	-453.33	-128.92	-0.3318	-26.81	9.05	11.48	6.29	0.3600	0.5048	0.2403	0.3318	0.0268
247	5726	2	3.4	10	4.6	-5334.88	-188.90	-0.4587	-25.10	11.12	7.61	6.37	0.4617	0.2729	0.2353	0.4587	0.0300
248	5749	2	1	10	13.27	-100.88	-126.16	-0.1973	-30.03	11.12	12.01	6.91	0.4617	0.5519	0.2793	0.1973	0.0216
249	5776	2	2.8	10	11.53	-3584.23	-168.74	-0.3052	-31.09	11.12	12.45	7.53	0.4617	0.5022	0.2822	0.3052	0.0254
250	5797	2	3.4	10	11.53	-5078.57	-189.63	-0.1438	-31.82	11.12	12.88	7.82	0.4617	0.5028	0.2895	0.1438	0.0185
251	5821	2.87	1	10	15	-450.48	-130.58	-0.1960	-33.04	11.96	13.36	7.73	0.4748	0.6066	0.3073	0.1960	0.0234
252	5848	2	1.6	10	14.13	-609.62	-132.54	-0.2004	-31.73	11.12	13.27	7.34	0.4617	0.5814	0.2878	0.2004	0.0212
253	5871	2	3.4	10	5.47	-4905.54	-184.25	-0.3979	-26.03	11.12	8.29	6.61	0.4617	0.2994	0.2390	0.3979	0.0290
254	5893	2.87	1	10	11.53	-179.35	-127.57	-0.1641	-29.70	11.96	10.70	7.04	0.4748	0.4826	0.2732	0.1641	0.0188
255	5916	5.47	1	7.6	9.8	-4812.05	-182.12	-0.4728	-29.32	12.29	9.47	7.55	0.4280	0.4217	0.3011	0.4728	0.0272
256	5940	2.87	1	9.4	11.53	-274.38	-126.42	-0.3503	-28.95	11.38	10.70	6.88	0.4442	0.4826	0.2711	0.3503	0.0256
257	5970	2	2.8	10	7.2	-3148.71	-161.14	-0.4727	-26.86	11.12	9.17	6.57	0.4617	0.3513	0.2417	0.4727	0.0296
258	5994	2	1.6	7.6	8.07	-178.25	-126.16	-0.2757	-23.50	9.05	8.79	5.66	0.3600	0.3718	0.2110	0.2757	0.0241
259	6017	2.87	1	9.4	9.8	-242.10	-126.36	-0.3230	-27.38	11.38	9.47	6.53	0.4442	0.4217	0.2534	0.3230	0.0246
260	6041	3.73	2.8	10	15	-7216.63	-213.08	-0.4223	-36.74	12.69	14.91	9.14	0.4887	0.6224	0.3427	0.4223	0.0274
261	6064	2	1	9.4	9.8	-2.95	-126.16	-0.1000	-26.16	10.63	9.47	6.06	0.4427	0.4217	0.2421	0.0989	0.0152
262	6086	2	1	9.4	11.53	-29.41	-126.16	-0.1262	-27.73	10.63	10.70	6.40	0.4427	0.4826	0.2555	0.1262	0.0169
263	6113	9.8	1.6	9.4	11.53	-15419.99	-313.74	-0.6464	-38.96	16.90	11.48	10.59	0.5782	0.5048	0.4424	0.6464	0.0329
264	6136	2.87	1	10	9.8	-70.55	-126.55	-0.1368	-28.02	11.96	9.47	6.59	0.4748	0.4217	0.2593	0.1368	0.0172
265	6158	2.87	1	10	13.27	-347.86	-129.11	-0.2106	-31.39	11.96	12.01	7.43	0.4748	0.5519	0.2916	0.2106	0.0239
266	6180	2	2.8	7.6	6.33	-3258.28	-162.42	-0.4819	-23.55	9.05	8.46	6.04	0.3600	0.3156	0.2203	0.4819	0.0328
267	6201	2	1	10	14.13	-21.06	-126.16	-0.1174	-30.70	11.12	12.56	7.02	0.4617	0.5763	0.2815	0.1174	0.0161
268	6227	8.93	1	10	9.8	-11757.81	-272.35	-0.2661	-35.60	16.97	9.47	9.16	0.5795	0.4217	0.4052	0.2661	0.0222
269	6248	8.93	1	10	11.53	-11768.53	-272.13	-0.2934	-37.39	16.97	10.70	9.72	0.5795	0.4826	0.4097	0.2934	0.0254
270	6270	2	2.2	10	13.27	-2110.60	-152.03	-0.1559	-31.87	11.12	13.24	7.51	0.4617	0.5602	0.2904	0.1559	0.0186

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
271	6292	3.73	1	10	8.07	-1855.71	-144.78	-0.4764	-27.43	12.69	8.23	6.51	0.4887	0.3657	0.2568	0.4764	0.0273
272	6314	2	1	7.6	15	-134.50	-126.16	-0.2311	-28.84	9.05	13.36	6.43	0.3600	0.6066	0.2577	0.2311	0.0225
273	6337	2	1	4.6	11.53	-293.26	-126.16	-0.3909	-21.94	6.35	10.70	4.89	0.2351	0.4826	0.1804	0.3909	0.0245
274	6362	2	3.4	10	9.8	-4947.98	-186.02	-0.2994	-30.40	11.12	11.73	7.55	0.4617	0.4529	0.2811	0.2994	0.0237
275	6381	2	3.4	10	15	-6487.54	-203.78	-0.4306	-35.12	11.12	15.67	8.33	0.4617	0.6277	0.3142	0.4306	0.0269
276	6401	8.93	1	10	14.13	-11905.55	-274.28	-0.2600	-40.09	16.97	12.56	10.56	0.5795	0.5763	0.4337	0.2600	0.0233
277	6424	2	5.8	8.8	9.8	-13033.38	-281.46	-0.8198	-32.43	10.05	13.88	8.50	0.3984	0.5025	0.3275	0.8198	0.0345
278	6446	2	1	7.6	14.13	-69.03	-126.16	-0.1658	-27.89	9.05	12.56	6.28	0.3600	0.5763	0.2406	0.1658	0.0190
279	6470	2	1	9.4	14.13	-3.54	-126.16	-0.1000	-30.03	10.63	12.56	6.83	0.4427	0.5763	0.2733	0.0928	0.0151
280	6490	2	1	10	10.67	-56.87	-126.16	-0.1537	-27.74	11.12	10.21	6.42	0.4617	0.4690	0.2609	0.1537	0.0191
281	6511	3.73	1	9.4	9.8	-1567.81	-143.99	-0.2510	-28.69	12.29	9.47	6.92	0.4631	0.4217	0.2711	0.2510	0.0227
282	6534	5.47	1	9.4	9.8	-4905.58	-182.79	-0.5135	-30.93	13.68	9.47	7.78	0.4862	0.4217	0.3122	0.5135	0.0264
283	6561	2	1	7	9.8	-184.70	-126.16	-0.2822	-23.46	8.51	9.47	5.48	0.3339	0.4217	0.2138	0.2822	0.0232
284	6584	5.47	1	10	15	-5093.57	-185.92	-0.4521	-36.58	14.11	13.36	9.11	0.4966	0.6066	0.3584	0.4521	0.0295
285	6610	2	2.2	9.4	15	-2663.25	-155.24	-0.4544	-32.64	10.63	14.46	7.55	0.4427	0.6181	0.2851	0.4544	0.0261
286	6632	5.47	1	10	9.8	-4777.39	-182.69	-0.3929	-31.37	14.11	9.47	7.78	0.4966	0.4217	0.3093	0.3929	0.0253
287	6656	2	2.2	9.4	9.8	-1670.84	-145.75	-0.2148	-28.03	10.63	10.59	6.81	0.4427	0.4355	0.2565	0.2148	0.0212
288	6679	2	1	9.4	8.93	-76.86	-126.16	-0.1740	-25.37	10.63	8.88	5.86	0.4427	0.4027	0.2323	0.1740	0.0195
289	6701	3.73	1	10	11.53	-1795.77	-144.94	-0.4034	-30.86	12.69	10.70	7.47	0.4887	0.4826	0.2971	0.4034	0.0265
290	6723	2	2.2	10	9.8	-1624.78	-144.86	-0.2395	-28.54	11.12	10.59	6.84	0.4617	0.4355	0.2569	0.2395	0.0225
291	6742	2	1	10	8.93	-101.61	-126.16	-0.1987	-25.98	11.12	8.88	5.98	0.4617	0.4027	0.2347	0.1987	0.0217
292	6761	2	3.4	9.4	9.8	-4969.43	-186.60	-0.2748	-29.84	10.63	11.73	7.48	0.4427	0.4529	0.2748	0.2748	0.0245
293	6785	2	1	8.2	9.8	-113.19	-126.16	-0.2105	-24.76	9.50	9.47	5.79	0.3827	0.4217	0.2254	0.2105	0.0209
294	6811	8.93	1	9.4	9.8	-11907.97	-268.16	-0.7484	-34.84	16.33	9.47	9.04	0.5619	0.4217	0.3965	0.7484	0.0245
295	6834	2	1	7.6	10.67	-104.86	-126.16	-0.2021	-25.08	9.05	10.21	5.82	0.3600	0.4690	0.2274	0.2021	0.0203
296	6855	5.47	1	10	14.13	-4994.12	-185.49	-0.3868	-35.67	14.11	12.56	9.00	0.4966	0.5763	0.3483	0.3868	0.0259
297	6875	2	1	4.6	9.8	-265.72	-126.16	-0.3636	-20.55	6.35	9.47	4.73	0.2351	0.4217	0.1753	0.3636	0.0258
298	6900	2	1	7.6	13.27	-148.85	-126.16	-0.2457	-27.26	9.05	12.01	6.21	0.3600	0.5519	0.2412	0.2457	0.0230
299	6926	8.93	1	9.4	10.67	-12001.15	-268.95	-0.7785	-35.92	16.33	10.21	9.39	0.5619	0.4690	0.4008	0.7785	0.0245
300	6951	2	5.8	9.4	9.8	-12896.12	-282.18	-0.6252	-33.24	10.63	13.88	8.73	0.4427	0.5025	0.3377	0.6252	0.0272
301	6973	2	3.4	7.6	14.13	-5884.79	-197.89	-0.2958	-31.14	9.05	14.76	7.34	0.3600	0.5836	0.2670	0.2958	0.0238
302	6999	2.87	1	7.6	14.13	-323.57	-128.17	-0.2609	-29.34	9.88	12.56	6.90	0.3721	0.5763	0.2634	0.2609	0.0237
303	7019	2	1	9.4	13.27	-76.13	-126.16	-0.1727	-29.37	10.63	12.01	6.74	0.4427	0.5519	0.2735	0.1727	0.0206
304	7039	2	1	7.6	12.4	-166.09	-126.16	-0.2631	-26.67	9.05	11.53	6.09	0.3600	0.5267	0.2377	0.2631	0.0245
305	7060	2	1	7	14.13	-179.14	-126.16	-0.2760	-27.17	8.51	12.56	6.10	0.3339	0.5763	0.2359	0.2760	0.0240
306	7081	2	1	7.6	9.8	-74.59	-126.16	-0.1719	-24.17	9.05	9.47	5.65	0.3600	0.4217	0.2183	0.1719	0.0189
307	7101	5.47	1	7.6	14.13	-4892.30	-183.20	-0.4667	-33.25	12.29	12.56	8.39	0.4280	0.5763	0.3186	0.4667	0.0265
308	7121	3.73	1	9.4	11.53	-1635.11	-144.49	-0.2783	-30.33	12.29	10.70	7.34	0.4631	0.4826	0.2856	0.2783	0.0242
309	7142	2	5.8	10	14.13	-14694.80	-308.57	-0.3318	-37.69	11.12	17.25	9.32	0.4617	0.6485	0.3536	0.3318	0.0256
310	7166	2	1	7.6	8.93	-149.61	-126.16	-0.2471	-23.40	9.05	8.88	5.47	0.3600	0.4027	0.2160	0.2471	0.0245
311	7188	2	2.2	7.6	14.13	-2221.98	-152.64	-0.2194	-29.71	9.05	13.87	6.79	0.3600	0.5910	0.2597	0.2194	0.0224
312	7207	2	1.6	9.4	11.53	-414.40	-129.34	-0.2587	-28.89	10.63	11.48	6.78	0.4427	0.5048	0.2666	0.2587	0.0240
313	7230	2	3.4	9.4	14.13	-5822.80	-198.02	-0.2228	-33.38	10.63	14.76	7.99	0.4427	0.5836	0.2985	0.2228	0.0208
314	7255	2	5.8	9.4	14.13	-14639.94	-308.19	-0.3072	-36.96	10.63	17.25	9.07	0.4427	0.6485	0.3518	0.3072	0.0251
315	7274	2	1	8.2	13.27	-187.47	-126.16	-0.2843	-27.94	9.50	12.01	6.44	0.3827	0.5519	0.2557	0.2843	0.0256

Calls	CPU Time	Fail1	Fail2	Repair1	Repair2	Z	Goal1	Goal2	Goal3	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
316	7294	3.73	1	10	9.8	-1711.18	-144.22	-0.3760	-29.17	12.69	9.47	7.01	0.4887	0.4217	0.2800	0.3760	0.0243
317	7315	3.73	1	9.4	14.13	-1780.79	-146.75	-0.2449	-32.78	12.29	12.56	7.92	0.4631	0.5763	0.3107	0.2449	0.0232
318	7336	2	2.2	9.4	14.13	-2233.16	-153.70	-0.1463	-31.97	10.63	13.87	7.46	0.4427	0.5910	0.2861	0.1463	0.0192
319	7361	2	1.6	4.6	9.8	-678.87	-129.79	-0.4889	-21.42	6.35	10.08	4.99	0.2351	0.4327	0.1803	0.4889	0.0273
320	7381	2	1	5.8	9.8	-72.20	-126.16	-0.1699	-22.07	7.45	9.47	5.15	0.2886	0.4217	0.1965	0.1699	0.0199
321	7405	2	7	9.4	9.8	-16715.20	-333.20	-0.4003	-34.59	10.63	14.79	9.17	0.4427	0.5120	0.3610	0.4003	0.0284
322	7431	2	1	7	8.93	-259.73	-126.16	-0.3573	-22.74	8.51	8.88	5.35	0.3339	0.4027	0.2067	0.3573	0.0278
323	7454	2	1	9.4	12.4	-93.36	-126.16	-0.1900	-28.74	10.63	11.53	6.58	0.4427	0.5267	0.2655	0.1900	0.0209
324	7476	3.73	1	9.4	15	-1939.53	-147.93	-0.3102	-33.74	12.29	13.36	8.09	0.4631	0.6066	0.3161	0.3102	0.0254
325	7499	2	1.6	9.4	13.27	-476.68	-130.80	-0.2056	-30.29	10.63	12.64	7.02	0.4427	0.5585	0.2702	0.2056	0.0206
326	7520	2	1.6	9.4	15	-679.34	-133.67	-0.1802	-31.79	10.63	13.85	7.31	0.4427	0.6042	0.2860	0.1802	0.0196
327	7546	2	6.4	9.4	9.8	-14912.84	-310.06	-0.4317	-34.15	10.63	14.50	9.03	0.4427	0.5177	0.3494	0.4317	0.0277
328	7573	4.6	1	9.4	9.8	-3064.39	-162.21	-0.3033	-29.77	13.00	9.47	7.29	0.4728	0.4217	0.2849	0.3033	0.0230
329	7595	2	1.6	4.6	2.87	-2506.78	-148.01	-0.8733	-14.49	6.35	4.32	3.82	0.2351	0.1574	0.1429	0.8733	0.0313
330	7618	9.8	1	9.4	9.8	-13609.67	-292.93	-0.4866	-35.84	16.90	9.47	9.47	0.5782	0.4217	0.4231	0.4866	0.0255
331	7640	5.47	1	9.4	8.93	-4965.79	-182.60	-0.5887	-30.04	13.68	8.88	7.48	0.4862	0.4027	0.3102	0.5887	0.0312
332	7661	2.87	1	5.8	9.8	-317.21	-126.38	-0.3976	-23.42	8.27	9.47	5.68	0.3033	0.4217	0.2132	0.3976	0.0254
333	7687	5.47	1.6	9.4	9.8	-6479.79	-201.06	-0.6388	-31.97	13.68	10.08	8.22	0.4862	0.4327	0.3243	0.6388	0.0292
334	7709	2.87	5.8	9.4	9.8	-14575.11	-300.53	-0.8493	-34.60	11.38	13.88	9.34	0.4442	0.5025	0.3515	0.8493	0.0326
335	7734	3.73	1.6	9.4	9.8	-3102.69	-161.77	-0.3763	-29.68	12.29	10.08	7.31	0.4631	0.4327	0.2763	0.3763	0.0300
336	7757	8.93	1	9.4	8.93	-12174.74	-270.58	-0.8235	-34.10	16.33	8.88	8.89	0.5619	0.4027	0.3911	0.8235	0.0269
337	7778	5.47	1.6	9.4	13.27	-6718.76	-204.31	-0.6202	-35.37	13.68	12.64	9.05	0.4862	0.5585	0.3466	0.6202	0.0274
338	7801	2.87	1.6	9.4	13.27	-2037.51	-147.66	-0.4297	-31.59	11.38	12.64	7.58	0.4442	0.5585	0.2932	0.4297	0.0272
339	7828	2	1	9.4	2	-665.84	-130.66	-0.4075	-16.52	10.63	2.96	2.93	0.4427	0.1095	0.1116	0.4075	0.0271
340	7851	3.73	1	9.4	8.93	-1635.26	-143.90	-0.3261	-27.84	12.29	8.88	6.67	0.4631	0.4027	0.2701	0.3261	0.0262
341	7877	2	1.6	9.4	8.93	-104.89	-126.48	-0.1764	-26.33	10.63	9.48	6.22	0.4427	0.4081	0.2399	0.1764	0.0201
342	7901	2	6.4	9.4	13.27	-16666.11	-332.87	-0.3767	-37.14	10.63	17.16	9.35	0.4427	0.6408	0.3647	0.3767	0.0268
343	7932	5.47	1	9.4	8.07	-5034.49	-183.15	-0.6139	-29.11	13.68	8.23	7.20	0.4862	0.3657	0.2971	0.6139	0.0282
344	7958	2	1	8.2	8.93	-188.22	-126.16	-0.2856	-24.01	9.50	8.88	5.63	0.3827	0.4027	0.2213	0.2856	0.0240
345	7982	8.93	1	9.4	13.27	-12151.14	-270.29	-0.8222	-38.54	16.33	12.01	10.21	0.5619	0.5519	0.4250	0.8222	0.0283
346	8005	2	1	9.4	10.67	-32.13	-126.16	-0.1290	-27.17	10.63	10.21	6.33	0.4427	0.4690	0.2530	0.1290	0.0173
347	8029	2	5.8	10	9.8	-12869.61	-281.53	-0.6499	-33.84	11.12	13.88	8.83	0.4617	0.5025	0.3419	0.6499	0.0310
348	8053	2	1	7	2.87	-481.97	-126.16	-0.5806	-15.69	8.51	3.75	3.43	0.3339	0.1467	0.1298	0.5806	0.0322

LNM-SO Method

Inventory Domain: LNM-SO Run Resulting in the BSF (Run 20)

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
0	1	6.539	42.75	3.57	4.12	7021.67	-3618.40	0.0685	0.2250	439	10	27	15.29	107.9	0.1536	0.0685	265	74
1	2	12.678	86	24	8.2	1567.68	-10325.19	0.1360	0.2250	584	7	14	34.32	107.9	0.2003	0.1360	155	39
1	3	18.957	44	6	2.8	5379.91	-5987.39	0.2347	0.2250	495	11	39	17.64	107.9	0.2016	0.2347	201	64
1	4	25.145	86	24	2.8	1704.82	-9954.21	0.0500	0.2250	595	7	39	48.17	107.9	0.1455	0.0075	147	44
1	5	31.334	100	30	5.08	1731.30	-9954.84	0.0635	0.2250	590	6	22	58.29	107.9	0.1197	0.0635	172	37
1	6	37.803	72	18	4.6	3003.72	-8717.14	0.1840	0.2250	552	7	24	29.79	107.9	0.1919	0.1840	175	50
2	7	43.892	100	30	7.92	1957.76	-9693.41	0.0678	0.2250	567	6	14	50.03	107.9	0.1405	0.0678	166	42
2	8	49.99	90.67	26	5.36	5134.40	-5838.72	0.0500	0.2250	574	6	21	49.31	107.9	0.1256	0.0000	343	70
2	9	56.159	95.33	28	6.64	2828.62	-8605.66	0.0500	0.2250	571	6	17	44.81	107.9	0.1172	0.0000	218	48
2	10	62.248	98.6	29.4	5.39	2771.66	-8674.00	0.0500	0.2250	583	6	21	48.65	107.9	0.1895	0.0347	220	55
2	11	68.426	48.2	7.8	3.34	6139.18	-4633.29	0.0500	0.2837	498	10	33	15.54	107.9	0.2837	0.0266	274	84
3	12	74.605	100	30	7.59	1095.29	-10685.65	0.0500	0.2250	570	6	15	56.42	107.9	0.0522	0.0175	121	37
3	13	80.874	90.2	25.8	5.46	2253.98	-9295.22	0.0500	0.2250	593	6	20	45.82	107.9	0.1783	0.0060	206	49
3	14	86.922	100	30	9.72	1465.49	-10318.70	0.0822	0.2250	572	6	12	49.36	107.9	0.1663	0.0822	147	36
4	15	93.141	82.73	22.6	7	4719.59	-6336.49	0.0500	0.2250	564	7	16	31.29	107.9	0.2032	0.0056	305	82
4	16	99.3	90.67	26	6.2	1799.90	-9840.12	0.0500	0.2250	564	6	18	46.07	107.9	0.1558	0.0000	150	40
4	17	105.408	94.63	27.7	5.79	1109.76	-10668.29	0.0500	0.2250	584	6	19	50.31	107.9	0.1080	0.0000	131	42
5	18	111.567	100	30	10	2420.08	-9095.91	0.0500	0.2250	564	5	11	55.82	107.9	0.1410	0.0476	196	48
5	19	117.656	93.54	27.23	7.19	2678.23	-8786.12	0.0500	0.2250	559	6	16	43.74	107.9	0.1807	0.0000	196	47
5	20	123.824	89.77	25.62	5	1591.95	-10089.66	0.0500	0.2250	580	6	22	47.12	107.9	0.1135	0.0000	151	41
6	21	129.933	97.32	28.85	9.39	2790.13	-8651.85	0.0500	0.2250	572	6	12	42.06	107.9	0.1324	0.0000	217	63
6	22	136.082	91.66	26.43	6.1	1613.22	-10064.65	0.0500	0.3214	568	6	18	35.55	107.9	0.3214	0.0078	146	36
6	23	142.26	95.17	27.93	5.97	1902.60	-9716.88	0.0500	0.2250	570	6	19	53.56	107.9	0.1075	0.0131	159	46
6	24	148.429	87.4	24.6	8.14	1388.68	-10333.58	0.0500	0.2250	592	7	14	38.05	107.9	0.1601	0.0314	163	42
6	25	154.608	90.79	26.06	5.26	1460.08	-10247.91	0.0500	0.2250	600	6	21	51.18	107.9	0.1416	0.0000	169	42
7	26	160.817	90.29	25.84	8.02	2194.18	-9366.98	0.0500	0.2250	555	6	14	41.75	107.9	0.1429	0.0020	166	45
7	27	166.915	92.73	26.89	7	1415.80	-10301.04	0.0500	0.2250	576	6	16	43.93	107.9	0.0953	0.0345	145	40
7	28	173.074	93.95	27.41	6.48	4000.06	-7199.93	0.0500	0.2250	560	6	17	46.51	107.9	0.1379	0.0000	274	49
7	29	179.152	95.65	28.14	6.13	1098.53	-10681.76	0.0500	0.2250	594	6	18	42.26	107.9	0.2125	0.0098	144	43
7	30	185.321	88.66	25.14	8.09	2440.48	-9071.42	0.0500	0.2250	566	6	14	43.05	107.9	0.1162	0.0000	195	44

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
7	31	191.41	91.71	26.45	5.49	2919.60	-8496.47	0.0500	0.2250	601	6	20	42.92	107.9	0.1758	0.0000	248	65
8	32	197.538	97.83	29.07	9.05	1937.21	-9675.35	0.0500	0.2250	583	6	12	45.54	107.9	0.1837	0.0130	186	50
8	33	203.657	94.77	27.76	7.27	2419.47	-9096.64	0.0500	0.2250	571	6	15	42.55	107.9	0.0788	0.0229	197	47
9	34	209.876	100	30	7.09	1326.86	-10703.68	0.1733	0.2250	572	6	16	37.80	107.9	0.1770	0.1733	124	32
9	35	216.044	97.83	29.07	7.59	4284.21	-6858.95	0.0500	0.2250	562	6	15	48.78	107.9	0.1659	0.0000	287	68
9	36	222.423	100	30	6.59	2308.46	-9445.86	0.1399	0.2730	581	6	17	44.34	107.9	0.2730	0.1399	187	51
10	37	228.722	99.27	29.69	4.82	1465.44	-10313.95	0.0802	0.2250	601	6	23	48.78	107.9	0.2124	0.0802	164	42
10	38	234.811	98.55	29.38	6.94	1680.01	-9983.99	0.0500	0.2250	573	6	16	48.67	107.9	0.0959	0.0463	159	36
10	39	240.879	98.91	29.54	5.88	1892.56	-9728.93	0.0500	0.2250	578	6	19	48.82	107.9	0.0896	0.0230	172	39
10	40	246.968	96.09	28.33	6.23	993.72	-10807.54	0.0500	0.2250	559	6	18	40.46	107.9	0.2142	0.0060	101	29
10	41	253.076	98.05	29.16	8.85	2841.85	-8589.77	0.0500	0.2250	590	6	13	46.86	107.9	0.1608	0.0000	242	61
11	42	259.295	99.34	29.73	5.09	798.12	-11042.26	0.0500	0.2250	591	6	22	52.69	107.9	0.1018	0.0000	120	35
11	43	265.484	98.7	29.44	6.97	2349.12	-9181.06	0.0500	0.2250	565	6	16	55.47	107.9	0.1013	0.0000	180	54
11	44	271.643	99.98	30	3.21	1270.01	-10475.99	0.0500	0.2250	594	6	34	55.64	107.9	0.1824	0.0000	136	31
12	45	277.791	96.95	28.71	4.68	1031.06	-10762.73	0.0500	0.2250	587	6	24	51.59	107.9	0.1294	0.0059	130	27
12	46	283.98	98.48	29.35	6.14	1283.71	-10459.55	0.0500	0.2250	576	6	18	47.38	107.9	0.1116	0.0202	138	32
13	47	290.199	94.92	27.85	3.58	1494.04	-10207.15	0.0500	0.2250	594	6	31	52.95	107.9	0.1399	0.0000	155	29
13	48	296.287	97.46	28.92	5.33	777.24	-11067.32	0.0500	0.2250	582	6	21	52.55	107.9	0.1095	0.0207	112	29
13	49	302.546	98.73	29.46	6.21	1696.03	-9964.77	0.0500	0.2250	571	6	18	46.88	107.9	0.1912	0.0000	153	39
13	50	308.615	96.42	28.47	6.12	1614.81	-10062.23	0.0500	0.2250	568	6	18	41.18	107.9	0.2113	0.0096	143	42
13	51	314.753	99.93	29.97	6.89	1564.82	-10122.25	0.0500	0.2315	586	6	16	41.13	107.9	0.2315	0.0254	162	49
13	52	320.952	97.19	28.81	4.72	1080.48	-10703.42	0.0500	0.2250	591	6	23	50.00	107.9	0.1513	0.0000	137	32
14	53	327.101	100	30	5.01	1852.54	-9776.96	0.0500	0.2250	588	6	22	53.39	107.9	0.1131	0.0000	178	38
14	54	333.369	98.82	29.5	5.57	1363.68	-10363.59	0.0500	0.2250	591	6	20	49.61	107.9	0.1369	0.0000	156	37
14	55	339.518	97.62	28.99	5.84	1199.23	-10560.92	0.0500	0.2250	575	6	19	55.78	107.9	0.1186	0.0000	130	32
15	56	345.717	96.17	28.38	3.54	850.81	-10979.03	0.0500	0.2250	604	6	31	48.58	107.9	0.1805	0.0000	123	41
15	57	351.805	98.05	29.18	5.22	1297.84	-10442.60	0.0500	0.2250	589	6	21	51.91	107.9	0.1065	0.0000	144	45
16	58	358.144	97.51	28.96	3.06	966.95	-10839.66	0.0500	0.2250	584	6	36	47.13	107.9	0.1646	0.0000	104	28
16	59	364.303	97.57	28.97	4.45	776.68	-11067.98	0.0500	0.2250	594	6	25	51.02	107.9	0.1631	0.0000	117	37
17	60	370.532	98.16	29.24	3.07	1555.37	-10133.56	0.0500	0.2250	594	6	36	61.26	107.9	0.0912	0.0000	145	41

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
17	61	376.7	97.67	29.02	3.9	967.27	-10839.28	0.0500	0.2250	590	6	28	48.01	107.9	0.1824	0.0000	122	31
17	62	382.879	97.43	28.92	4.31	1997.49	-9603.01	0.0500	0.2250	583	6	26	50.95	107.9	0.1335	0.0183	175	37
17	63	389.028	97.69	29.04	3.26	1328.24	-10406.11	0.0500	0.2250	602	6	34	55.36	107.9	0.1292	0.0000	140	50
17	64	395.186	96.49	28.51	3.7	2531.28	-8962.46	0.0500	0.2250	597	6	30	52.23	107.9	0.1314	0.0000	212	50
17	65	401.586	97.41	28.9	4.76	1693.36	-9967.97	0.0500	0.2250	584	6	23	51.04	107.9	0.1082	0.0000	164	34
18	66	407.794	99.8	29.94	5.04	741.10	-11110.68	0.0500	0.2250	583	6	22	56.36	107.9	0.1268	0.0000	109	30
18	67	413.943	98.15	29.22	4.37	1906.60	-9712.07	0.0500	0.2250	589	6	25	53.12	107.9	0.1721	0.0000	174	46
18	68	420.082	100	30	5.71	2908.36	-8622.53	0.0969	0.2250	579	6	19	52.06	107.9	0.1547	0.0969	223	50
19	69	426.3	100	30	4.17	1946.58	-9664.11	0.0500	0.2250	604	6	26	60.00	107.9	0.1211	0.0260	192	49
19	70	432.409	98.94	29.57	4.46	1278.49	-10465.81	0.0500	0.2250	588	6	25	56.04	107.9	0.0879	0.0000	142	32
19	71	438.588	98.18	29.24	4.61	1606.73	-10071.92	0.0500	0.2250	589	6	24	53.47	107.9	0.1241	0.0000	163	35
20	72	444.806	99.71	29.9	4.32	988.65	-10813.72	0.0500	0.2443	585	6	25	49.30	107.9	0.2443	0.0000	121	32
21	73	451.025	100	30	6.37	1025.29	-10769.65	0.0500	0.2250	593	6	17	58.74	107.9	0.0811	0.0151	142	38
21	74	457.274	99.62	29.86	4.82	1761.03	-9959.24	0.0802	0.2250	586	6	23	48.10	107.9	0.1709	0.0802	163	42
21	75	463.473	99.81	29.93	5.59	863.76	-10963.49	0.0500	0.2250	583	6	20	49.83	107.9	0.1797	0.0000	116	37
22	76	469.601	99.59	29.83	6.16	1927.43	-9687.08	0.0500	0.2250	570	6	18	53.09	107.9	0.0768	0.0000	165	40
22	77	475.79	99.65	29.87	5.24	569.45	-11316.66	0.0500	0.2250	594	6	21	54.08	107.9	0.1185	0.0000	110	38
22	78	481.959	99.68	29.88	4.78	1242.18	-10509.39	0.0500	0.2250	586	6	23	47.76	107.9	0.1901	0.0000	140	33
22	79	488.207	99.81	29.93	5.53	2041.15	-9550.62	0.0500	0.2250	581	6	20	56.48	107.9	0.1084	0.0060	179	47
22	80	494.376	99.72	29.9	4.39	1053.49	-10735.81	0.0500	0.2250	596	6	25	56.04	107.9	0.0999	0.0000	133	43
22	81	500.645	99.39	29.75	5.09	1547.26	-10143.29	0.0500	0.2250	591	6	22	50.30	107.9	0.1543	0.0000	163	39
23	82	506.804	99.46	29.8	4.15	1273.41	-10471.91	0.0500	0.2250	587	6	26	53.49	107.9	0.1008	0.0034	139	32
23	83	513.062	99.64	29.86	4.84	1208.72	-10549.54	0.0500	0.2250	594	6	23	47.41	107.9	0.1491	0.0000	145	39
24	84	519.281	99.93	30	3.96	593.60	-11287.67	0.0500	0.2250	596	6	28	51.73	107.9	0.1897	0.0000	106	33
24	85	525.35	99.66	29.88	4.53	143.98	-11827.23	0.0500	0.2250	604	6	24	52.76	107.9	0.1860	0.0000	97	29
24	86	531.559	100	30	3.39	788.09	-11054.29	0.0500	0.2250	594	6	32	59.56	107.9	0.1365	0.0408	108	35
25	87	537.767	100	30	4.78	1015.79	-10919.53	0.1077	0.2250	586	6	23	47.42	107.9	0.1901	0.1077	121	30
25	88	543.856	99.82	29.95	4.46	1213.78	-10543.46	0.0500	0.2250	592	6	25	61.49	107.9	0.0821	0.0000	143	32
26	89	550.045	100	30	4.8	2506.81	-8991.83	0.0500	0.2250	585	6	23	53.68	107.9	0.1194	0.0058	206	50
26	90	556.123	99.91	29.98	4.59	602.64	-11276.83	0.0500	0.2250	584	6	24	53.68	107.9	0.1485	0.0000	102	24

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
26	91	562.272	99.82	29.94	4.49	926.35	-10888.38	0.0500	0.2250	588	6	25	50.09	107.9	0.1569	0.0077	116	42
27	92	568.491	100	30	4.7	1894.75	-9726.30	0.0500	0.2250	600	6	23	54.92	107.9	0.1036	0.0010	188	50
27	93	574.779	99.87	29.96	4.54	760.01	-11087.99	0.0500	0.2250	584	6	24	51.00	107.9	0.1601	0.0000	111	25
28	94	581.238	99.73	29.93	4.25	1513.82	-10183.42	0.0500	0.2250	585	6	26	50.01	107.9	0.1275	0.0012	147	40
28	95	587.397	99.87	29.97	4.51	2937.95	-8474.46	0.0500	0.2250	586	6	24	59.17	107.9	0.0980	0.0000	226	61
28	96	593.576	99.93	29.98	4.65	1533.30	-10165.08	0.0521	0.2250	577	6	24	46.16	107.9	0.1550	0.0521	145	32
28	97	599.694	99.81	29.95	4.93	1133.68	-10639.59	0.0500	0.2250	587	6	22	58.88	107.9	0.1196	0.0000	137	31
28	98	605.873	99.88	29.96	4.48	1212.77	-10544.68	0.0500	0.2250	592	6	25	58.67	107.9	0.0821	0.0000	139	40
29	99	612.082	99.95	30	4.63	949.14	-10861.03	0.0500	0.2250	587	6	24	55.54	107.9	0.1026	0.0000	123	31
29	100	618.261	99.91	29.98	4.56	2036.88	-9555.75	0.0500	0.2250	584	6	24	56.19	107.9	0.1048	0.0245	176	48
30	101	624.489	100	30	3.98	458.39	-11508.01	0.0742	0.2250	604	6	28	50.96	107.9	0.1967	0.0742	110	23
30	102	630.598	99.96	30	4.46	1115.33	-10726.65	0.0771	0.2250	580	6	25	54.11	107.9	0.1602	0.0771	116	38
30	103	636.817	100	30	3.5	899.14	-10921.03	0.0500	0.2250	585	6	31	55.53	107.9	0.1297	0.0254	108	29
31	104	642.975	100	30	3.85	1458.79	-10249.45	0.0500	0.2250	583	6	29	59.19	107.9	0.1521	0.0094	137	39
31	105	649.174	99.98	30	4.24	1096.83	-10683.81	0.0500	0.2250	594	6	26	60.69	107.9	0.1084	0.0000	134	38
31	106	655.343	99.96	30	4.44	614.95	-11262.06	0.0500	0.2250	593	6	25	53.14	107.9	0.1698	0.0000	105	39
32	107	661.582	99.93	30	3.47	1091.12	-10690.65	0.0500	0.2250	587	6	32	56.41	107.9	0.1548	0.0000	119	32
32	108	667.76	99.96	30	4.13	2115.76	-9461.08	0.0500	0.2250	589	6	27	55.41	107.9	0.1241	0.0133	184	45
32	109	673.879	99.98	30	4.45	1258.94	-10489.28	0.0500	0.2250	585	6	25	59.60	107.9	0.0750	0.0000	137	32

Logistics Domain: LNM-SO Run Resulting in the BSF (Run 10)

Iteration	calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	G1	G2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
0	1	5.32	6	2.81	3.89	9.79	9.24	339.21	1795.90	-0.0278	29.18	12.10	12.55	51.38	49.15	48.90	1.5997	1.6565	1.6529	2.7790	2.1743
1	2	9.65	12	4.2	4.4	9	11	309.18	1673.11	-0.0763	25.84	12.99	10.70	51.11	49.36	48.58	1.6091	1.6411	1.6742	7.6294	3.9660
1	3	14.08	5	1.8	2.6	6	8	473.84	2347.64	-0.0318	43.06	19.03	14.33	52.97	50.57	50.19	1.5894	1.6138	1.6514	3.1783	1.8267
1	4	18.46	12	4.2	2.6	6	8	473.79	2347.64	-0.0697	43.06	19.03	14.33	52.97	50.57	50.19	1.5894	1.6138	1.6514	6.9695	4.0492
1	5	22.69	5	1.8	4.4	9	11	309.24	1673.11	-0.0314	25.84	12.99	10.70	51.11	49.36	48.58	1.6091	1.6411	1.6742	3.1350	2.2029
1	6	27.02	5	1.8	4.4	9	8	337.37	1788.39	-0.0281	25.84	12.99	14.33	51.11	49.36	50.19	1.6091	1.6411	1.6514	2.8076	2.0221
1	7	31.27	11	4.12	5	10	10.9	286.45	1579.97	-0.0807	22.87	11.92	10.82	51.50	48.84	48.25	1.6038	1.6652	1.6659	8.0736	4.3192
1	8	35.61	8	2.96	3.94	8.56	9.45	342.66	1810.17	-0.0486	28.81	13.54	12.30	51.48	49.54	49.20	1.6130	1.6626	1.6508	4.8594	3.0376
1	9	39.90	14	5	5	10	12	278.92	1549.04	-0.0680	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	6.7963	3.8164
2	10	44.23	5	2.04	5	10	12	278.97	1549.04	-0.0251	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	2.5142	1.7405
2	11	48.56	8	3.12	4.42	9.36	10.25	311.72	1683.36	-0.0497	25.69	12.47	11.45	51.31	48.90	49.12	1.6168	1.6475	1.6748	4.9689	2.5443
3	12	52.83	10	3.13	5	9.01	12	284.29	1570.93	-0.0459	22.87	12.93	9.79	51.50	49.61	48.30	1.6038	1.6387	1.6880	4.5896	2.9133
3	13	57.06	8	2.97	4.64	9.4	10.8	299.34	1632.61	-0.0409	24.46	12.47	10.85	51.47	49.51	48.22	1.6186	1.6559	1.6707	4.0873	2.1773
4	14	61.69	13	4.67	5	9.8	12	280.00	1553.46	-0.0629	22.87	12.10	9.79	51.50	49.22	48.30	1.6038	1.6571	1.6880	6.2878	3.5583
4	15	66.45	9	3.23	4.76	9.4	11.6	292.33	1603.91	-0.0490	24.02	12.47	10.16	51.80	49.51	48.67	1.6179	1.6559	1.6720	4.9032	2.5166
5	16	70.72	15	5	5	10	12	278.91	1549.04	-0.0785	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	7.8527	3.8846
5	17	74.95	11	3.81	4.88	9.56	11.8	285.24	1574.91	-0.0592	23.36	12.40	9.96	51.09	49.35	48.09	1.6159	1.6596	1.6800	5.9212	3.3166
6	18	79.24	11	3.74	5	10	12	278.93	1549.04	-0.0565	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	5.6522	3.1917
6	19	83.52	11	3.97	5	9.76	12	279.91	1553.09	-0.0660	22.87	12.10	9.79	51.50	49.04	48.30	1.6038	1.6696	1.6880	6.5968	3.0916
7	20	87.81	13	5	5	10	12	278.90	1549.04	-0.0866	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	8.6619	3.7436
7	21	92.05	12	4.09	5	9.96	12	279.06	1549.60	-0.0615	22.87	11.92	9.79	51.50	49.09	48.30	1.6038	1.6677	1.6880	6.1456	3.7562
8	22	96.39	10	3.64	5	10	12	278.92	1549.04	-0.0635	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	6.3534	3.0649
8	23	100.63	12	4.16	5	10	12	278.93	1549.04	-0.0613	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	6.1326	3.7524
8	24	104.86	11	3.9	5	10	12	278.93	1549.04	-0.0568	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	5.6845	3.1976
10	25	109.28	13	4.48	5	10	12	278.90	1549.04	-0.0846	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	8.4627	3.7275
11	26	113.63	13	4.73	5	10	12	278.90	1549.04	-0.0856	22.87	11.92	9.79	51.50	48.84	48.30	1.6038	1.6652	1.6880	8.5585	3.7363

PERT Domain: LNM-SO Run Resulting in the BSF (Run 1)

Iteration	calls	CPU Time	LOW3	LOW4	LOW6	Z	G1	G2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
0	1	5.488	9.1	6.75	7.35	0.1561	21.76	20.86	0.6550	0.1925	0.1525	0.2325	0.4225	0.2150	0.4000	0.2325	0.5525	20.8557	0.1018
1	2	10.134	10.6	5.6	5.6	0.1529	21.76	20.80	0.6050	0.1875	0.2075	0.1875	0.4175	0.1775	0.4275	0.1875	0.6350	20.7977	0.1032
1	3	14.37	6.4	2.4	2.4	0.1369	21.76	20.51	0.6350	0.2075	0.1575	0.1650	0.4700	0.1550	0.5225	0.1650	0.6800	20.5098	0.1112
1	4	18.506	10.6	5.6	2.4	0.1495	21.76	20.74	0.6025	0.1800	0.2175	0.2025	0.4000	0.1375	0.4425	0.2025	0.6600	20.7375	0.1040
1	5	22.692	9.3	2.32	1	0.1400	21.76	20.56	0.6000	0.2025	0.1975	0.1525	0.4475	0.1375	0.5125	0.1525	0.7100	20.5648	0.1096
1	6	26.838	9.2	4.53	3.47	0.1454	21.76	20.66	0.6100	0.2000	0.1900	0.1825	0.4275	0.1475	0.4800	0.1825	0.6700	20.6622	0.1068
2	7	31.034	6.93	1.28	1	0.1349	21.76	20.47	0.6125	0.2100	0.1775	0.1475	0.4650	0.1450	0.5300	0.1475	0.7075	20.4735	0.1122
2	8	35.16	8.77	3.44	1.93	0.1413	21.76	20.59	0.6075	0.2025	0.1900	0.1675	0.4400	0.1400	0.5025	0.1675	0.6925	20.5893	0.1088
2	9	39.325	5.09	1	1	0.1325	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4295	0.1133
3	10	43.571	5	1	1	0.1324	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4278	0.1134
3	11	47.717	6.93	1.91	1.47	0.1361	21.76	20.49	0.6200	0.2050	0.1750	0.1575	0.4625	0.1475	0.5200	0.1575	0.6950	20.4946	0.1116
4	12	51.943	5	1	1.93	0.1332	21.76	20.44	0.6300	0.2175	0.1525	0.1525	0.4775	0.1550	0.5400	0.1525	0.6925	20.4426	0.1132
4	13	56.089	5.5	1.47	1.47	0.1338	21.76	20.45	0.6300	0.2125	0.1575	0.1575	0.4725	0.1500	0.5350	0.1575	0.6925	20.4545	0.1127
5	14	60.285	5.03	1	1.31	0.1327	21.76	20.43	0.6300	0.2150	0.1550	0.1525	0.4775	0.1525	0.5400	0.1525	0.6950	20.4332	0.1133
6	15	64.441	5.06	1	1	0.1324	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4289	0.1133
6	16	68.867	5.03	1	1	0.1324	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4284	0.1133
7	17	73.073	5.02	1	1	0.1324	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4282	0.1133

Production Domain: LNM-SO Run Resulting in the BSF (Run 3)

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
0	1	4.66	8.11	18.48	13.67	3.88	3.24	2.28	57.92	-1937.50	-1.8273	19.55	57.77	0.9881	5.6534	0.0402	1.7702	0.0293
1	2	9.06	9	18	13	6.2	3.4	5.2	63.64	-1673.91	-1.8459	16.90	62.28	0.9608	5.7742	0.0386	1.7671	0.0257
1	3	13.52	6	12	7	3.8	1.6	2.8	34.48	-3016.69	-1.7938	30.37	48.48	0.9047	5.5394	0.0428	1.7412	0.0292
1	4	17.90	9	18	7	3.8	1.6	2.8	51.11	-2249.21	-1.8869	22.68	53.78	0.9777	5.5951	0.0419	1.9087	0.0172
1	5	22.26	6	12	13	6.2	3.4	5.2	53.58	-2136.54	-1.8408	21.54	57.58	0.9395	5.6615	0.0428	1.7944	0.0251
1	6	26.71	6	12	13	6.2	1.6	2.8	46.89	-2443.15	-1.9000	24.61	54.98	0.9416	5.7742	0.0386	1.8806	0.0188
1	7	31.05	9	18	7	3.8	3.4	5.2	56.44	-2004.81	-1.8643	20.22	56.82	0.9699	5.7742	0.0386	1.8039	0.0286
1	8	35.66	5.7	12.16	7.22	3.03	1.55	2	32.08	-3125.03	-1.8776	31.47	46.38	0.8901	5.7186	0.0395	1.8491	0.0229
1	9	40.11	7.35	15.08	10.11	4.61	2.47	3.51	51.26	-2246.02	-1.7424	22.64	55.33	0.9537	5.6182	0.0405	1.6122	0.0319
1	10	44.55	5	10	5	3	1	2	17.86	-3781.39	-1.7702	38.05	42.38	0.8448	5.5876	0.0409	1.6779	0.0278
2	11	49.14	5.56	10	5	5.05	1	4.65	30.65	-3192.26	-1.8188	32.13	47.61	0.8678	5.5883	0.0414	1.7749	0.0260
2	12	53.61	6.83	13.67	8.67	4.47	2.1	3.47	45.05	-2527.89	-1.9000	25.47	53.64	0.9106	5.7742	0.0386	1.8847	0.0204
3	13	58.20	5	10	9.67	5.55	1	2	32.24	-3119.70	-1.7919	31.40	49.02	0.8865	5.7742	0.0386	1.6591	0.0307
3	14	62.64	6.26	12.33	8.33	4.68	1.7	3.38	40.59	-2734.64	-1.8302	27.54	51.77	0.9153	5.6595	0.0429	1.7739	0.0291
4	15	67.25	6.19	12	5	3	1	2	26.43	-3385.21	-1.8656	34.08	44.41	0.8638	5.7742	0.0386	1.8066	0.0257
4	16	71.70	6.09	12	7.78	4.57	1.3	2.84	37.06	-2896.97	-1.8220	29.17	50.09	0.8954	5.6917	0.0398	1.7467	0.0264
5	17	76.32	5	10	7.89	5.07	1	2.62	29.48	-3244.06	-1.9000	32.65	47.50	0.8902	5.7521	0.0390	1.9210	0.0174
5	18	80.76	5.63	11	7.45	4.43	1.2	2.71	31.61	-3148.07	-1.8156	31.69	48.20	0.8776	5.6933	0.0403	1.7334	0.0263
6	19	85.38	5	10	5	3	1	2.56	19.00	-3727.01	-1.8542	37.50	42.84	0.8535	5.7742	0.0386	1.7838	0.0274
6	20	89.84	5.46	10.67	6.59	4.25	1.1	2.68	29.71	-3233.48	-1.8851	32.55	46.63	0.8855	5.6720	0.0405	1.8795	0.0220
7	21	94.45	5	10	5.52	4.42	1	2.48	24.54	-3472.68	-1.8290	34.95	45.27	0.8473	5.7553	0.0389	1.7396	0.0294
7	22	98.91	5.29	10.33	6.26	4.11	1	2.64	25.85	-3411.80	-1.8658	34.34	45.76	0.8640	5.7044	0.0397	1.8301	0.0220
8	23	103.54	5.58	10.67	5	3	1	3.44	23.50	-3524.04	-1.6924	35.47	44.28	0.8575	5.7359	0.0392	1.4729	0.0359
8	24	107.99	5.29	10.33	5.57	3.92	1	2.72	24.21	-3486.38	-1.9000	35.09	44.88	0.8456	5.6695	0.0401	1.9266	0.0154
9	25	112.62	5.03	10.89	6.14	3	1	2	22.44	-3569.14	-1.8335	35.92	43.75	0.8464	5.6564	0.0402	1.7816	0.0265
9	26	117.07	5.3	10.45	5.57	3.58	1	2.52	24.25	-3485.09	-1.8750	35.08	44.16	0.8646	5.7742	0.0386	1.8252	0.0196
10	27	121.97	5.6	11.19	5	3	1	2.21	24.34	-3482.15	-1.8177	35.05	43.85	0.8671	5.7742	0.0386	1.7106	0.0329
10	28	126.43	5.3	10.59	5.28	3.24	1	2.41	22.21	-3583.16	-1.6926	36.06	44.00	0.8483	5.5983	0.0422	1.5192	0.0344
11	29	131.05	5	10	5.55	3.47	1	2.9	21.69	-3602.60	-1.8776	36.25	43.98	0.8556	5.7742	0.0386	1.8305	0.0208
11	30	135.48	5.2	10.46	5.28	3.24	1	2.45	22.56	-3563.63	-1.8511	35.86	43.83	0.8534	5.7302	0.0393	1.7922	0.0249

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
12	31	140.11	5.4	10.92	5.04	3	1	2.56	22.88	-3548.80	-1.8429	35.72	43.80	0.8534	5.7057	0.0397	1.7838	0.0279
12	32	144.56	5.2	10.46	5.28	3.08	1	2.52	21.89	-3595.00	-1.8250	36.18	43.61	0.8478	5.7209	0.0395	1.7430	0.0300
13	33	149.19	5	10	5.58	3.16	1	2.94	20.19	-3671.64	-1.8847	36.95	43.20	0.8350	5.7742	0.0386	1.8446	0.0181
13	34	153.65	5.17	10.41	5.29	3.08	1	2.58	21.57	-3613.41	-1.6747	36.36	43.41	0.8655	5.6060	0.0434	1.4807	0.0345
14	35	158.26	5	10	5.77	3.21	1	2	19.76	-3692.77	-1.8378	37.16	42.90	0.8281	5.7114	0.0396	1.7719	0.0230
14	36	162.71	5.07	10.3	5.39	3.11	1	2.49	21.01	-3638.41	-1.7005	36.61	43.40	0.8575	5.4226	0.0493	1.5934	0.0335
15	37	167.34	5	10	5.97	3.28	1	2.24	20.35	-3667.18	-1.7608	36.90	43.53	0.8449	5.6252	0.0405	1.6466	0.0301
15	38	171.79	5.01	10.15	5.51	3.14	1	2.4	20.98	-3642.70	-1.5879	36.66	42.78	0.8419	5.0324	0.0627	1.4984	0.0362
16	39	176.43	5	10	5	3.37	1	2.88	20.81	-3644.96	-1.8075	36.68	43.53	0.8504	5.7009	0.0397	1.7147	0.0261
16	40	180.88	5	10	5.48	3.19	1	2.44	19.94	-3683.73	-1.8663	37.07	43.29	0.8402	5.5938	0.0435	1.8680	0.0176
17	41	185.51	5	10	5.22	3	1	2	17.98	-3775.94	-1.7702	37.99	42.58	0.8471	5.5876	0.0409	1.6779	0.0278
17	42	189.96	5	10	5.39	3.17	1	2.44	20.50	-3662.28	-1.6894	36.86	43.01	0.8429	5.6463	0.0407	1.4968	0.0347
18	43	194.57	5	10	5.85	3	1	2	18.66	-3744.67	-1.7702	37.68	42.98	0.8474	5.5876	0.0409	1.6779	0.0278
18	44	199.01	5	10	5.42	3.11	1	2.29	19.39	-3710.21	-1.8148	37.34	42.62	0.8467	5.7742	0.0386	1.7049	0.0296
19	45	203.64	5	10	5	3	1	2.26	17.92	-3780.33	-1.7094	38.04	42.15	0.8148	5.3883	0.0476	1.6228	0.0352
19	46	208.08	5	10	5.4	3.06	1	2.25	19.43	-3708.53	-1.8057	37.32	42.93	0.8392	5.4941	0.0477	1.7801	0.0251
20	47	212.70	5	10	5.03	3	1	2	17.86	-3781.40	-1.7702	38.05	42.41	0.8451	5.5876	0.0409	1.6779	0.0278
20	48	217.14	5	10	5.31	3.04	1	2.14	18.82	-3737.10	-1.7815	37.61	42.68	0.8330	5.6108	0.0430	1.6927	0.0265
21	49	221.77	5	10	5	3	1	2.27	18.14	-3765.57	-1.9000	37.89	42.25	0.8420	5.7742	0.0386	1.8826	0.0166
21	50	226.22	5	10	5.18	3	1	2.14	17.87	-3783.26	-1.6914	38.07	42.42	0.8308	5.7395	0.0392	1.4696	0.0382
22	51	230.83	5	10	5.37	3	1	2	18.12	-3769.51	-1.7702	37.93	42.67	0.8468	5.5876	0.0409	1.6779	0.0278
22	52	235.28	5	10	5.18	3	1	2.09	18.31	-3761.81	-1.7302	37.86	42.14	0.8394	5.4728	0.0486	1.6362	0.0264
23	53	239.91	5	10	5	3	1	2.18	17.21	-3808.23	-1.9000	38.32	42.35	0.8230	5.7382	0.0392	1.8905	0.0187
23	54	244.36	5	10	5.1	3	1	2.09	18.25	-3764.54	-1.7302	37.88	42.07	0.8385	5.4728	0.0486	1.6362	0.0264
24	55	249.00	5	10	5.21	3	1	2	17.98	-3775.94	-1.7702	37.99	42.57	0.8470	5.5876	0.0409	1.6779	0.0278
24	56	253.46	5	10	5.1	3	1	2.07	17.61	-3794.11	-1.7279	38.18	41.95	0.8248	5.6235	0.0448	1.5813	0.0307
25	57	258.08	5	10	5	3	1	2.15	17.44	-3800.60	-1.7829	38.24	41.97	0.8333	5.3056	0.0567	1.7973	0.0234
25	58	262.55	5	10	5.08	3	1	2.07	17.59	-3795.02	-1.7279	38.19	41.93	0.8247	5.6235	0.0448	1.5813	0.0307
26	59	267.19	5	10	5	3	1	2.2	18.26	-3764.55	-1.7063	37.88	42.08	0.8435	5.6139	0.0424	1.5413	0.0345
26	60	271.64	5	10	5.04	3	1	2.1	18.19	-3770.02	-1.6297	37.94	41.92	0.8322	5.4120	0.0472	1.4554	0.0306
26	61	276.09	5	10	5.13	3	1	2.05	18.83	-3736.76	-1.7837	37.61	42.06	0.8513	5.5377	0.0455	1.7214	0.0281

Reliability Domain: LNM-SO Run Resulting in the BSF (Run 10)

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
0	1	4.36	12.54	4.84	9.92	4.88	33499.59	535.58	1.1423	40.28	1627	4239	2564	19.48	9.06	11.75	0.6729	0.3203	0.5407	1.1423	0.0310
1	2	8.52	12.4	4.2	8.2	10	29839.63	490.89	1.0241	43.95	1785	2959	2306	17.92	12.51	13.53	0.6314	0.4764	0.5638	1.0241	0.0296
1	3	12.68	4.6	1.8	2.8	4	10631.25	249.25	0.9729	18.66	5482	7303	6452	7.14	5.54	5.98	0.2631	0.2088	0.2662	0.9729	0.0343
1	4	16.81	12.4	4.2	2.8	4	47392.08	708.57	1.3231	35.73	2462	5036	1953	13.46	7.77	14.50	0.5330	0.2748	0.6953	1.3231	0.0263
1	5	20.95	4.6	1.8	8.2	10	5047.87	186.82	0.3364	30.22	3131	3664	4925	11.93	10.51	7.78	0.4332	0.4522	0.3017	0.3364	0.0262
1	6	25.14	4.67	2.12	10	10.44	6214.16	200.37	0.4281	32.99	2721	3441	4545	13.50	11.09	8.40	0.4975	0.4640	0.3173	0.4281	0.0296
1	7	29.29	8.54	3.16	7.28	7.22	17435.13	338.70	0.6843	34.17	2447	4080	3445	14.55	9.42	10.21	0.5046	0.3576	0.4346	0.6843	0.0277
2	8	33.45	2	1	4.68	12	473.47	126.16	0.5710	22.65	6227	3419	8122	6.46	11.20	4.99	0.2429	0.5095	0.1839	0.5710	0.0318
2	9	37.62	6.57	2.48	7.3	8.61	10784.24	257.50	0.4696	31.82	2830	3887	4168	12.93	9.99	8.90	0.4483	0.3936	0.3493	0.4696	0.0259
2	10	41.82	2	1	2.06	12	1497.27	137.48	0.6980	18.83	10345	3419	11180	3.97	11.20	3.66	0.1392	0.5095	0.1307	0.6980	0.0303
3	11	46.01	2	1	4.64	8.22	319.35	126.16	0.4174	19.27	6294	4750	8876	6.41	8.27	4.59	0.2402	0.3667	0.1682	0.4174	0.0278
3	12	50.18	3.97	1.68	6.42	9.11	4030.64	169.92	0.6591	26.32	3901	4050	5708	9.87	9.60	6.85	0.3525	0.4053	0.2531	0.6591	0.0311
3	13	54.34	2	1	2.86	7.33	434.47	126.16	0.5330	16.26	8623	5202	10619	4.75	7.65	3.86	0.1688	0.3379	0.1399	0.5330	0.0290
4	14	58.55	2.03	1.16	10	12	110.24	126.16	0.2068	29.30	3395	3365	5860	11.21	11.31	6.77	0.4674	0.5052	0.2733	0.2068	0.0229
4	15	62.69	3.32	1.48	6.88	10.16	2006.40	149.37	0.2639	26.49	4034	3760	5954	9.58	10.30	6.61	0.3518	0.4495	0.2448	0.2639	0.0258
4	16	67.13	2	1	10	12	149.15	126.16	0.2458	28.93	3452	3419	6020	11.12	11.20	6.61	0.4617	0.5095	0.2692	0.2458	0.0250
5	17	71.31	2	1	3.76	10.67	494.06	126.16	0.5920	20.28	7291	3798	9080	5.58	10.21	4.50	0.2026	0.4690	0.1656	0.5920	0.0257
5	18	75.47	2.66	1.24	6.88	10.56	776.66	130.33	0.5429	25.72	4324	3704	6459	9.06	10.46	6.20	0.3473	0.4602	0.2303	0.5429	0.0304
6	19	79.69	2	1	3.34	11.45	418.41	126.16	0.5164	20.18	7836	3577	9390	5.20	10.64	4.35	0.1869	0.4774	0.1582	0.5164	0.0287
6	20	83.83	2.01	1.04	5.77	10.72	273.15	126.16	0.3707	23.10	5357	3773	7590	7.45	10.30	5.34	0.2870	0.4741	0.2021	0.3707	0.0261
7	21	88.03	2.01	1.08	7.57	11.17	208.85	126.16	0.3060	25.48	4349	3627	6822	9.03	10.56	5.89	0.3577	0.4696	0.2286	0.3060	0.0235
7	22	92.18	2.01	1.04	5.66	10.92	117.56	126.16	0.2151	23.08	5424	3694	7617	7.34	10.43	5.31	0.2808	0.4715	0.1994	0.2151	0.0205
8	23	96.42	2.02	1.12	8.1	9.42	232.20	126.16	0.3295	24.57	4134	4166	6988	9.44	9.36	5.77	0.3779	0.4189	0.2278	0.3295	0.0248
8	24	100.54	2.01	1.06	6.39	10.71	178.97	126.16	0.2764	23.81	4949	3752	7348	7.99	10.32	5.49	0.3124	0.4697	0.2132	0.2764	0.0241
9	25	104.77	2.03	1.18	10	8.96	119.16	126.16	0.2162	26.44	3395	4320	6538	11.21	9.10	6.13	0.4674	0.4059	0.2431	0.2162	0.0240
9	26	108.92	2.01	1.09	7.58	10.2	144.65	126.16	0.2419	24.85	4351	3887	6948	9.04	9.99	5.81	0.3619	0.4544	0.2239	0.2419	0.0217
10	27	113.12	2.04	1.27	10	12	212.09	126.16	0.3086	29.65	3412	3316	5757	11.18	11.57	6.90	0.4683	0.5175	0.2723	0.3086	0.0248
10	28	117.29	2.02	1.14	8.92	10.39	183.02	126.16	0.2800	26.59	3800	3816	6449	10.18	10.18	6.23	0.4172	0.4612	0.2470	0.2800	0.0257
11	29	121.49	2.03	1.22	10	12	151.72	126.16	0.2483	29.49	3395	3338	5857	11.21	11.46	6.82	0.4674	0.5137	0.2686	0.2483	0.0254
11	30	125.63	2.03	1.17	9.39	11.03	311.16	126.16	0.4080	27.55	3630	3627	6235	10.63	10.50	6.42	0.4319	0.4663	0.2563	0.4080	0.0267
12	31	129.85	2.01	1.05	8.79	10.07	276.10	126.16	0.3732	25.77	3845	3954	6688	10.05	9.73	6.00	0.3998	0.4372	0.2369	0.3732	0.0279
12	32	134.04	2.02	1.16	9.39	11.03	102.25	126.16	0.1991	27.75	3610	3606	6235	10.66	10.67	6.42	0.4389	0.4834	0.2547	0.1991	0.0224
12	33	138.17	2.03	1.22	9.7	11.52	212.40	126.16	0.3091	28.45	3541	3472	6025	10.82	11.01	6.62	0.4476	0.4938	0.2683	0.3091	0.0234
12	34	142.32	2.01	1.09	7.81	11.25	201.66	126.16	0.2987	25.95	4235	3610	6638	9.29	10.61	6.05	0.3733	0.4748	0.2359	0.2987	0.0236
13	35	146.53	2.01	1.06	8.91	10.11	162.61	126.16	0.2597	26.03	3803	3936	6612	10.11	9.86	6.06	0.4129	0.4451	0.2416	0.2597	0.0242
13	36	150.66	2.02	1.16	9.45	11.05	174.85	126.16	0.2717	27.43	3609	3619	6249	10.58	10.51	6.35	0.4339	0.4669	0.2517	0.2717	0.0225
14	37	154.89	2.04	1.22	10	10.29	197.44	126.16	0.2942	27.78	3412	3839	6198	11.18	10.14	6.46	0.4683	0.4584	0.2590	0.2942	0.0236
14	38	159.04	2.02	1.16	9.73	10.77	137.74	126.16	0.2345	27.93	3508	3662	6181	10.88	10.60	6.45	0.4540	0.4780	0.2603	0.2345	0.0212
14	39	163.24	2.03	1.19	9.86	10.53	236.37	126.16	0.3332	27.82	3489	3746	6193	11.04	10.31	6.48	0.4608	0.4605	0.2572	0.3332	0.0265
14	40	167.42	2.01	1.07	9.02	10.3	185.94	126.16	0.2830	26.44	3768	3878	6486	10.24	10.03	6.17	0.4217	0.4540	0.2487	0.2830	0.0230
14	41	171.60	2.01	1.1	8.03	11.33	207.92	126.16	0.3050	26.14	4146	3588	6608	9.40	10.67	6.06	0.3729	0.4777	0.2333	0.3050	0.0233
15	42	175.79	2.04	1.22	10	10.3	153.79	126.16	0.2506	27.81	3412	3803	6163	11.18	10.18	6.46	0.4683	0.4616	0.2620	0.2506	0.0231
15	43	179.93	2.02	1.16	9.75	10.82	152.32	126.16	0.2491	27.90	3501	3659	6176	10.91	10.53	6.46	0.4504	0.4732	0.2587	0.2491	0.0216
16	44	184.08	2.05	1.32	10	11.33	234.22	126.16	0.3309	28.78	3414	3501	5873	11.11	10.94	6.73	0.4564	0.4792	0.2610	0.3309	0.0249
16	45	188.24	2.03	1.19	10	10.81	61.82	126.16	0.1585	28.30	3395	3643	6011	11.21	10.47	6.62	0.4674	0.4657	0.2655	0.1585	0.0183
16	46	192.41	2.02	1.13	9.51	10.56	128.42	126.16	0.2253	27.35	3577	3766	6316	10.74	10.27	6.34	0.4483	0.4611	0.2502	0.2253	0.0239
17	47	196.63	2.01	1.12	9.75	11.46	54.58	126.16	0.1513	28.38	3508	3504	6046	10.92	10.89	6.57	0.4555	0.4886	0.2620	0.1513	0.0179
17	48	200.77	2.03	1.17	9.88	10.88	158.97	126.16	0.2557	28.06	3459	3633	6147	11.01	10.56	6.48	0.4595	0.4706	0.2525	0.2557	0.0218
17	49	204.93	2	1.07	9.62	12	190.02	126.16	0.2867	28.70	3569	3385	6030	10.81	11.27	6.62	0.4527	0.5138	0.2656	0.2867	0.0242
18	50	209.14	2.02	1.08	9.63	9.49	184.56	126.16	0.2816	26.19	3565	4156	6616	10.76	9.37	6.06	0.4507	0.4195	0.2407	0.2816	0.0244
18	51	213.33	2.02	1.15	9.82	10.75	229.24	126.16	0.3260	27.76	3489	3707	6180	10.93	10.39	6.44	0.4546	0.4637	0.2542	0.3260	0.0275
18	52	217.48	2.03	1.18	9.91	11.37	139.65	126.16	0.2363	28.54	3459	3523	6014	11.04	10.88	6.62	0.4520	0.4891	0.2623	0.2363	0.0235
19	53	221.67	2.02	1.12	9.72	10.12	163.27	126.16	0.2602	26.95	3508	3934	6389	10.87	9.84	6.24	0.4536	0.4389	0.2516	0.2602	0.0219
19	54	225.89	2.03	1.16	9.86	11.06	194.62	126.16	0.2914	28.10	3489	3619	6112	11.04	10.54	6.53	0.4608	0.4696	0.2634	0.2914	0.0258
19	55	229.98	2.03	1.17	9.89	11.38	143.62	126.16	0.2403	28.45	3468	3526	6001	10.98	10.85	6.62	0.4488	0.4866	0.2639	0.2403	0.0232

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
19	56	234.13	2.02	1.13	9.53	10.65	149.42	126.16	0.2463	27.52	3583	3725	6270	10.73	10.38	6.40	0.4397	0.4681	0.2566	0.2463	0.0248
19	57	238.30	2.03	1.16	9.98	11.95	204.65	126.16	0.3012	29.23	3414	3370	5879	11.12	11.35	6.76	0.4616	0.5132	0.2688	0.3012	0.0255
19	58	242.48	2.03	1.17	9.98	9.21	294.47	126.16	0.3915	26.55	3414	4216	6524	11.12	9.27	6.16	0.4616	0.4084	0.2410	0.3915	0.0296
20	59	247.00	2.02	1.12	9.6	12	175.06	126.16	0.2717	28.70	3569	3376	6014	10.77	11.30	6.63	0.4480	0.5113	0.2625	0.2717	0.0240
20	60	251.14	2.02	1.15	9.79	11.36	253.04	126.16	0.3498	28.26	3498	3543	6054	10.88	10.80	6.58	0.4518	0.4812	0.2612	0.3498	0.0245
21	61	255.36	2.01	1.11	9.41	10.8	84.35	126.16	0.1812	27.32	3602	3706	6271	10.56	10.40	6.36	0.4316	0.4658	0.2518	0.1812	0.0204
21	62	259.51	2.02	1.14	9.69	11.37	181.13	126.16	0.2779	28.15	3550	3554	6082	10.80	10.79	6.57	0.4489	0.4814	0.2622	0.2779	0.0246
22	63	263.74	2.02	1.15	9.69	10.15	161.76	126.16	0.2587	27.09	3550	3879	6360	10.80	10.00	6.29	0.4489	0.4529	0.2494	0.2587	0.0245
22	64	267.89	2.02	1.13	9.65	11.07	242.37	126.16	0.3392	27.80	3549	3627	6206	10.84	10.51	6.44	0.4508	0.4676	0.2580	0.3392	0.0243
22	65	272.03	2.02	1.14	9.67	10.61	325.42	126.16	0.4222	27.68	3549	3746	6276	10.88	10.39	6.41	0.4453	0.4667	0.2543	0.4222	0.0299
22	66	276.22	2.03	1.17	9.88	11.39	202.54	126.16	0.2992	28.48	3459	3518	6016	11.01	10.87	6.60	0.4595	0.4875	0.2586	0.2992	0.0254
22	67	280.36	2.02	1.13	9.55	10.73	114.26	126.16	0.2111	27.47	3570	3725	6227	10.73	10.33	6.41	0.4469	0.4610	0.2534	0.2111	0.0219
23	68	284.58	2	1.07	9.28	11.11	79.69	126.16	0.1766	27.29	3681	3613	6341	10.37	10.62	6.30	0.4235	0.4787	0.2479	0.1766	0.0202
23	69	288.73	2.02	1.12	9.58	11.25	24.47	126.16	0.1213	27.81	3578	3606	6170	10.71	10.64	6.46	0.4421	0.4747	0.2558	0.1213	0.0163
24	70	292.91	2	1.1	9.39	10.05	51.51	126.16	0.1485	26.69	3610	3941	6453	10.61	9.88	6.20	0.4402	0.4487	0.2446	0.1485	0.0178
24	71	297.07	2.01	1.11	9.5	11.03	107.98	126.16	0.2048	27.62	3598	3634	6248	10.71	10.51	6.40	0.4485	0.4721	0.2540	0.2048	0.0212
24	72	301.25	2	1.09	9.28	9.08	111.69	126.16	0.2089	25.39	3681	4309	6777	10.37	9.10	5.92	0.4235	0.4088	0.2320	0.2089	0.0222
25	73	305.50	2	1.07	9.36	10.98	222.44	126.16	0.3193	27.57	3642	3641	6288	10.60	10.60	6.37	0.4429	0.4850	0.2567	0.3193	0.0250
25	74	309.65	2.01	1.1	9.46	10.86	214.59	126.16	0.3115	27.38	3609	3683	6308	10.58	10.46	6.34	0.4343	0.4725	0.2549	0.3115	0.0274
25	75	313.73	2.01	1.12	9.5	10.79	119.59	126.16	0.2164	27.54	3598	3712	6222	10.71	10.41	6.42	0.4485	0.4662	0.2603	0.2164	0.0206
25	76	317.89	2.02	1.13	9.53	10.66	62.33	126.16	0.1592	27.52	3583	3710	6260	10.73	10.38	6.40	0.4397	0.4674	0.2578	0.1592	0.0185
25	77	322.02	2.01	1.11	9.41	10.73	12.76	126.16	0.1097	27.20	3602	3714	6283	10.56	10.32	6.32	0.4316	0.4657	0.2486	0.1097	0.0156
25	78	326.18	2.01	1.12	9.71	11.32	204.93	126.16	0.3017	27.96	3550	3599	6168	10.81	10.67	6.48	0.4483	0.4799	0.2587	0.3017	0.0232
26	79	330.41	2.01	1.09	9.09	9.96	218.18	126.16	0.3152	26.13	3738	3975	6610	10.26	9.81	6.06	0.4148	0.4468	0.2372	0.3152	0.0257
26	80	334.56	2.01	1.1	9.4	10.64	117.02	126.16	0.2139	27.36	3610	3752	6309	10.66	10.33	6.37	0.4394	0.4662	0.2546	0.2139	0.0205
26	81	338.74	2.01	1.11	9.56	10.98	93.46	126.16	0.1903	27.63	3577	3633	6286	10.72	10.56	6.35	0.4428	0.4739	0.2560	0.1903	0.0196
27	82	342.97	2.01	1.1	9.24	10.29	65.04	126.16	0.1620	26.58	3699	3878	6466	10.40	9.98	6.20	0.4226	0.4511	0.2472	0.1620	0.0193
28	83	347.21	2.02	1.15	9.51	9.76	89.64	126.16	0.1866	26.52	3577	4024	6464	10.74	9.61	6.16	0.4483	0.4287	0.2440	0.1866	0.0242
28	84	351.36	2.01	1.11	9.39	10.43	78.45	126.16	0.1754	27.08	3603	3816	6379	10.63	10.17	6.28	0.4423	0.4629	0.2511	0.1754	0.0206
28	85	355.55	2.01	1.09	9.34	10.77	310.90	126.16	0.4078	27.15	3630	3710	6318	10.50	10.36	6.30	0.4254	0.4676	0.2475	0.4078	0.0249
28	86	359.70	2	1.07	9.29	11.07	48.70	126.16	0.1456	27.37	3685	3613	6291	10.41	10.62	6.35	0.4229	0.4858	0.2553	0.1456	0.0176
28	87	363.82	2.02	1.13	9.52	10.67	125.53	126.16	0.2224	27.33	3582	3731	6289	10.70	10.27	6.36	0.4382	0.4576	0.2490	0.2224	0.0208
28	88	367.99	2	1.1	9.39	10.12	91.09	126.16	0.1881	26.68	3610	3905	6441	10.61	9.87	6.20	0.4402	0.4411	0.2502	0.1881	0.0195
29	89	372.23	2	1.06	9.15	10.43	137.18	126.16	0.2342	26.63	3731	3839	6495	10.36	10.09	6.18	0.4233	0.4614	0.2438	0.2342	0.0216
29	90	376.42	2.01	1.1	9.33	10.55	337.07	126.16	0.4340	27.08	3631	3773	6346	10.54	10.21	6.32	0.4318	0.4653	0.2533	0.4340	0.0292
29	91	380.58	2.01	1.11	9.43	10.61	180.24	126.16	0.2771	27.28	3611	3753	6289	10.62	10.30	6.37	0.4319	0.4640	0.2531	0.2771	0.0239
29	92	384.76	2	1.07	9.3	11.04	220.57	126.16	0.3174	27.35	3675	3629	6261	10.41	10.56	6.38	0.4267	0.4763	0.2569	0.3174	0.0243
29	93	388.92	2.02	1.13	9.51	10.68	37.81	126.16	0.1347	27.43	3577	3714	6299	10.74	10.34	6.34	0.4483	0.4640	0.2530	0.1347	0.0172
29	94	393.06	2	1.1	9.39	10.18	29.47	126.16	0.1264	26.89	3610	3887	6455	10.61	10.05	6.23	0.4402	0.4556	0.2449	0.1264	0.0175
30	95	397.28	2.02	1.15	9.47	9.9	165.51	126.16	0.2625	26.69	3582	3964	6406	10.67	9.78	6.24	0.4394	0.4388	0.2488	0.2625	0.0226
30	96	401.42	2.01	1.11	9.39	10.47	26.30	126.16	0.1232	27.07	3603	3800	6395	10.63	10.17	6.26	0.4423	0.4624	0.2471	0.1232	0.0165
30	97	405.56	2.01	1.13	9.43	10.19	226.82	126.16	0.3238	26.79	3611	3898	6407	10.62	9.93	6.25	0.4319	0.4448	0.2432	0.3238	0.0245
30	98	409.75	2	1.07	9.31	11.01	218.03	126.16	0.3149	27.49	3659	3633	6300	10.48	10.68	6.34	0.4285	0.4850	0.2578	0.3149	0.0262
30	99	413.93	2.02	1.13	9.5	10.69	132.05	126.16	0.2289	27.50	3578	3723	6247	10.74	10.36	6.40	0.4483	0.4653	0.2525	0.2289	0.0232
30	100	418.11	2	1.1	9.39	10.24	79.39	126.16	0.1763	26.81	3610	3885	6435	10.61	9.98	6.22	0.4402	0.4518	0.2453	0.1763	0.0196
31	101	422.34	2.02	1.15	9.46	9.97	223.00	126.16	0.3200	26.54	3609	3945	6453	10.56	9.79	6.19	0.4326	0.4339	0.2434	0.3200	0.0249
31	102	426.51	2.01	1.11	9.39	10.49	78.42	126.16	0.1753	27.08	3603	3797	6397	10.63	10.20	6.26	0.4423	0.4652	0.2467	0.1753	0.0198
31	103	430.99	2.01	1.09	9.35	10.75	208.60	126.16	0.3055	27.30	3642	3732	6325	10.58	10.38	6.34	0.4399	0.4721	0.2506	0.3055	0.0239
32	104	435.22	2.01	1.13	9.42	10.23	50.56	126.16	0.1475	26.87	3602	3878	6387	10.59	10.01	6.27	0.4353	0.4524	0.2443	0.1475	0.0177
33	105	439.50	2	1.09	9.23	10.06	281.74	126.16	0.3787	26.44	3711	3934	6551	10.42	9.89	6.13	0.4270	0.4499	0.2405	0.3787	0.0285
33	106	443.66	2.01	1.11	9.37	10.37	288.66	126.16	0.3856	26.95	3630	3832	6423	10.57	10.14	6.24	0.4399	0.4635	0.2472	0.3856	0.0258
33	107	447.82	2.01	1.12	9.43	10.53	226.73	126.16	0.3236	27.10	3611	3773	6383	10.62	10.21	6.27	0.4319	0.4660	0.2483	0.3236	0.0235
33	108	452.00	2.02	1.13	9.49	10.69	132.04	126.16	0.2289	27.48	3578	3723	6251	10.72	10.36	6.39	0.4467	0.4653	0.2520	0.2289	0.0232
33	109	456.15	2	1.1	9.39	10.29	55.47	126.16	0.1524	26.86	3610	3878	6387	10.61	9.98	6.26	0.4402	0.4511	0.2469	0.1524	0.0180
34	110	460.39	2	1.09	9.24	10.08	235.60	126.16	0.3326	26.33	3711	3925	6569	10.41	9.82	6.10	0.4262	0.4355	0.2420	0.3326	0.0244

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
34	111	464.56	2.01	1.11	9.37	10.39	325.19	126.16	0.4221	26.93	3630	3832	6405	10.57	10.09	6.26	0.4399	0.4562	0.2489	0.4221	0.0257
34	112	468.71	2.01	1.12	9.43	10.54	226.73	126.16	0.3236	27.12	3611	3773	6360	10.62	10.22	6.28	0.4319	0.4677	0.2485	0.3236	0.0235
34	113	472.86	2.02	1.13	9.48	10.69	319.64	126.16	0.4165	27.33	3590	3723	6284	10.61	10.36	6.36	0.4380	0.4653	0.2552	0.4165	0.0285
34	114	477.03	2	1.1	9.39	10.33	90.84	126.16	0.1878	26.90	3610	3858	6406	10.61	10.04	6.25	0.4402	0.4563	0.2518	0.1878	0.0207
35	115	481.29	2	1.09	9.25	10.1	114.52	126.16	0.2115	26.44	3699	3925	6466	10.40	9.86	6.18	0.4226	0.4392	0.2441	0.2115	0.0238
35	116	485.43	2.01	1.11	9.36	10.4	178.16	126.16	0.2751	27.07	3642	3819	6386	10.55	10.23	6.29	0.4367	0.4664	0.2520	0.2751	0.0244
35	117	489.62	2	1.1	9.31	10.25	217.44	126.16	0.3144	26.63	3659	3885	6479	10.48	9.98	6.17	0.4285	0.4505	0.2476	0.3144	0.0275
35	118	493.78	2.02	1.13	9.47	10.69	214.77	126.16	0.3116	27.38	3582	3723	6304	10.67	10.36	6.35	0.4394	0.4653	0.2537	0.3116	0.0263
35	119	497.92	2	1.1	9.39	10.37	86.85	126.16	0.1838	27.04	3610	3819	6449	10.61	10.20	6.23	0.4402	0.4668	0.2448	0.1838	0.0194
36	120	502.17	2	1.09	9.26	10.12	209.43	126.16	0.3064	26.53	3682	3905	6451	10.43	9.89	6.21	0.4250	0.4423	0.2421	0.3064	0.0248
36	121	506.36	2.01	1.11	9.37	10.41	307.75	126.16	0.4047	27.01	3630	3816	6373	10.57	10.15	6.28	0.4399	0.4594	0.2502	0.4047	0.0296
36	122	510.53	2	1.1	9.31	10.26	216.52	126.16	0.3135	26.63	3659	3885	6475	10.48	9.98	6.17	0.4285	0.4503	0.2470	0.3135	0.0274
36	123	514.68	2.02	1.13	9.46	10.69	320.12	126.16	0.4170	27.28	3609	3723	6285	10.56	10.36	6.36	0.4326	0.4653	0.2493	0.4170	0.0265
36	124	518.86	2	1.1	9.39	10.41	71.36	126.16	0.1683	27.08	3610	3819	6451	10.61	10.24	6.23	0.4402	0.4674	0.2457	0.1683	0.0187
37	125	523.11	2	1.09	9.27	10.14	225.42	126.16	0.3224	26.48	3682	3934	6524	10.45	9.89	6.14	0.4255	0.4461	0.2420	0.3224	0.0269
37	126	527.25	2.01	1.11	9.37	10.42	279.99	126.16	0.3769	26.98	3630	3828	6409	10.57	10.15	6.26	0.4399	0.4614	0.2486	0.3769	0.0255
37	127	531.40	2	1.1	9.32	10.28	213.84	126.16	0.3108	26.64	3659	3885	6467	10.47	9.99	6.18	0.4258	0.4506	0.2478	0.3108	0.0272
38	128	535.67	2.01	1.12	9.41	10.55	58.67	126.16	0.1556	27.15	3602	3773	6333	10.56	10.27	6.31	0.4316	0.4662	0.2497	0.1556	0.0184
39	129	539.93	2.02	1.13	9.35	10.49	219.26	126.16	0.3162	27.05	3654	3792	6399	10.54	10.24	6.27	0.4360	0.4668	0.2503	0.3162	0.0262
39	130	544.11	2.01	1.12	9.37	10.45	316.90	126.16	0.4138	27.05	3630	3812	6408	10.57	10.21	6.27	0.4399	0.4674	0.2523	0.4138	0.0268
39	131	548.21	2.01	1.11	9.38	10.43	158.65	126.16	0.2556	27.05	3618	3816	6406	10.63	10.17	6.25	0.4454	0.4629	0.2483	0.2556	0.0243
39	132	552.37	2	1.1	9.39	10.44	89.60	126.16	0.1865	27.08	3610	3797	6376	10.61	10.20	6.27	0.4402	0.4660	0.2528	0.1865	0.0201
40	133	556.61	2.02	1.13	9.35	10.46	255.36	126.16	0.3523	26.99	3654	3797	6370	10.54	10.16	6.29	0.4360	0.4635	0.2507	0.3523	0.0299
40	134	560.79	2.01	1.11	9.38	10.45	116.80	126.16	0.2137	27.09	3618	3800	6408	10.63	10.20	6.27	0.4454	0.4629	0.2520	0.2137	0.0218
40	135	564.96	2	1.1	9.39	10.47	43.52	126.16	0.1404	27.02	3610	3800	6424	10.61	10.17	6.24	0.4402	0.4621	0.2490	0.1404	0.0183
41	136	569.22	2.01	1.13	9.57	10.7	3.15	126.16	0.1000	27.50	3570	3710	6249	10.76	10.34	6.39	0.4480	0.4665	0.2574	0.0855	0.0142
41	137	573.40	2.01	1.12	9.41	10.5	28.78	126.16	0.1257	27.13	3602	3781	6358	10.56	10.28	6.29	0.4316	0.4692	0.2507	0.1257	0.0182
41	138	577.55	2.01	1.14	9.73	10.91	161.70	126.16	0.2585	27.90	3518	3641	6181	10.90	10.54	6.47	0.4542	0.4734	0.2597	0.2585	0.0221
42	139	581.80	2.01	1.11	9.49	10.51	90.93	126.16	0.1878	27.23	3598	3773	6353	10.62	10.31	6.31	0.4371	0.4714	0.2516	0.1878	0.0209
42	140	586.00	2.01	1.12	9.45	10.53	84.14	126.16	0.1811	27.04	3599	3773	6375	10.57	10.21	6.26	0.4341	0.4660	0.2493	0.1811	0.0217
42	141	590.18	2	1.1	9.41	10.49	145.08	126.16	0.2420	27.09	3613	3800	6367	10.62	10.18	6.28	0.4367	0.4631	0.2508	0.2420	0.0230
43	142	594.44	2.02	1.15	9.5	10.62	58.89	126.16	0.1557	27.52	3578	3726	6232	10.74	10.37	6.41	0.4483	0.4692	0.2585	0.1557	0.0189
43	143	598.61	2.01	1.12	9.45	10.55	84.16	126.16	0.1811	27.15	3599	3773	6333	10.57	10.27	6.31	0.4341	0.4662	0.2487	0.1811	0.0217
43	144	602.71	2.01	1.14	9.48	10.59	59.43	126.16	0.1563	27.26	3598	3753	6330	10.64	10.28	6.33	0.4407	0.4595	0.2478	0.1563	0.0191
43	145	606.88	2	1.1	9.43	10.51	36.51	126.16	0.1334	27.07	3602	3800	6375	10.59	10.22	6.26	0.4361	0.4675	0.2541	0.1334	0.0211
44	146	611.46	2.01	1.11	9.5	10.54	104.61	126.16	0.2015	27.21	3598	3773	6363	10.71	10.22	6.29	0.4485	0.4650	0.2507	0.2015	0.0206
44	147	615.64	2.01	1.12	9.46	10.54	148.83	126.16	0.2457	27.07	3609	3773	6358	10.58	10.22	6.28	0.4343	0.4677	0.2479	0.2457	0.0233
44	148	619.78	2.01	1.12	9.43	10.55	226.74	126.16	0.3236	27.21	3611	3773	6355	10.62	10.27	6.31	0.4319	0.4662	0.2489	0.3236	0.0235
44	149	623.94	2	1.1	9.44	10.53	224.67	126.16	0.3216	27.05	3611	3773	6380	10.60	10.20	6.25	0.4333	0.4664	0.2492	0.3216	0.0235
45	150	628.22	2.02	1.15	9.47	10.58	108.98	126.16	0.2059	27.33	3582	3751	6300	10.67	10.31	6.35	0.4394	0.4605	0.2517	0.2059	0.0203
45	151	632.41	2.01	1.14	9.46	10.57	200.05	126.16	0.2969	27.18	3609	3766	6366	10.58	10.30	6.29	0.4343	0.4641	0.2441	0.2969	0.0233
45	152	636.56	2	1.1	9.45	10.55	72.77	126.16	0.1697	27.11	3618	3773	6408	10.64	10.21	6.26	0.4374	0.4653	0.2468	0.1697	0.0200
46	153	640.86	2.02	1.15	9.46	10.56	187.35	126.16	0.2842	27.12	3609	3753	6367	10.56	10.28	6.28	0.4326	0.4619	0.2472	0.2842	0.0230
46	154	645.07	2.01	1.11	9.45	10.55	83.51	126.16	0.1804	27.09	3599	3773	6342	10.57	10.23	6.30	0.4341	0.4668	0.2500	0.1804	0.0216
46	155	649.23	2	1.1	9.46	10.57	138.13	126.16	0.2350	27.21	3618	3766	6365	10.62	10.28	6.31	0.4345	0.4677	0.2517	0.2350	0.0220
47	156	653.49	2.02	1.15	9.45	10.54	219.85	126.16	0.3167	27.16	3609	3766	6351	10.58	10.29	6.29	0.4339	0.4674	0.2495	0.3167	0.0241
47	157	657.69	2.01	1.11	9.46	10.56	214.13	126.16	0.3110	27.15	3609	3766	6367	10.58	10.29	6.28	0.4343	0.4673	0.2480	0.3110	0.0247
47	158	661.78	2	1.1	9.47	10.58	294.38	126.16	0.3913	27.17	3609	3766	6354	10.58	10.28	6.31	0.4333	0.4660	0.2468	0.3913	0.0280
48	159	666.04	2.02	1.15	9.44	10.53	154.86	126.16	0.2518	27.14	3599	3766	6333	10.57	10.27	6.30	0.4337	0.4622	0.2454	0.2518	0.0219
48	160	670.24	2.01	1.14	9.45	10.54	38.33	126.16	0.1352	27.15	3599	3753	6339	10.57	10.29	6.29	0.4341	0.4646	0.2451	0.1352	0.0171
49	161	674.54	2.01	1.13	9.52	10.55	31.31	126.16	0.1282	27.40	3577	3753	6281	10.74	10.29	6.37	0.4486	0.4650	0.2529	0.1282	0.0169
49	162	678.70	2.01	1.13	9.46	10.55	102.79	126.16	0.1997	27.18	3609	3753	6325	10.58	10.29	6.32	0.4343	0.4650	0.2462	0.1997	0.0200
50	163	682.95	2.01	1.13	9.56	11.03	70.06	126.16	0.1669	27.72	3577	3621	6226	10.72	10.60	6.40	0.4428	0.4782	0.2557	0.1669	0.0189
50	164	687.12	2.01	1.13	9.49	10.63	147.48	126.16	0.2444	27.29	3598	3746	6298	10.62	10.31	6.36	0.4371	0.4641	0.2547	0.2444	0.0218
50	165	691.27	2.01	1.13	9.45	10.43	129.85	126.16	0.2268	27.04	3599	3797	6381	10.57	10.21	6.26	0.4341	0.4637	0.2533	0.2268	0.0214

TS-SO Method

Inventory Domain: TS-SO Run Resulting in the BSF (Run 11)

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
1	1	6.18	89.74	28.87	5.31	1969.29	-9636.85	0.0500	0.2250	572	6	21	53.63	107.9	0.1232	0.0083	166	39
1	2	12.34	91.74	28.87	5.31	1240.11	-10558.37	0.0693	0.2585	597	6	21	38.53	107.9	0.2585	0.0693	150	42
1	3	18.48	87.74	28.87	5.31	2110.54	-9513.95	0.0693	0.2758	574	6	21	37.20	107.9	0.2758	0.0693	173	43
1	4	24.62	89.74	26.87	5.31	1482.51	-10220.99	0.0500	0.2250	586	6	21	44.06	107.9	0.1932	0.0000	153	42
1	5	30.94	89.74	28.87	7.31	1777.11	-9867.46	0.0500	0.2250	573	6	15	40.63	107.9	0.1904	0.0301	163	43
1	6	37.29	89.74	28.87	3.31	1312.49	-10425.01	0.0500	0.2250	599	7	33	46.32	107.9	0.2226	0.0074	140	39
1	7	43.42	90.74	28.87	5.31	1279.22	-10464.94	0.0500	0.2250	597	7	21	46.48	107.9	0.1932	0.0000	154	38
1	8	49.56	88.74	28.87	5.31	1559.33	-10128.81	0.0500	0.2250	592	7	21	49.10	107.9	0.1768	0.0298	164	39
1	9	55.69	89.74	29.87	5.31	1506.91	-10191.70	0.0500	0.2250	595	7	21	53.98	107.9	0.1232	0.0083	166	36
1	10	61.83	89.74	27.87	5.31	1524.67	-10170.40	0.0500	0.2250	596	7	21	45.42	107.9	0.1407	0.0000	166	41
1	11	67.97	89.74	28.87	6.31	1385.92	-10336.90	0.0500	0.2250	569	6	18	41.94	107.9	0.1920	0.0131	133	37
1	12	74.11	89.74	28.87	4.31	2354.55	-9174.54	0.0500	0.2250	587	7	26	47.41	107.9	0.1327	0.0073	196	43
2	13	80.25	93.74	28.87	5.31	1552.81	-10136.63	0.0500	0.2250	577	6	21	42.57	107.9	0.1748	0.0187	150	34
2	14	86.43	91.74	26.87	5.31	1619.65	-10056.42	0.0500	0.2250	585	6	21	43.06	107.9	0.2027	0.0057	163	36
2	15	93.17	91.74	28.87	7.31	2714.76	-8742.29	0.0500	0.2250	568	6	15	41.03	107.9	0.1634	0.0000	206	57
2	16	99.30	91.74	28.87	3.31	2164.36	-9402.77	0.0500	0.2250	603	6	33	51.52	107.9	0.2126	0.0000	198	40
2	17	105.44	92.74	28.87	5.31	1425.96	-10288.85	0.0500	0.2250	592	6	21	49.01	107.9	0.1407	0.0000	155	46
2	18	111.62	91.74	29.87	5.31	1390.40	-10331.53	0.0500	0.2250	594	6	21	42.81	107.9	0.1663	0.0000	162	33
2	19	117.72	91.74	27.87	5.31	1490.43	-10211.49	0.0500	0.2250	588	6	21	42.89	107.9	0.1663	0.0000	155	44
2	20	123.84	91.74	28.87	6.31	1995.10	-9605.87	0.0500	0.2250	563	6	18	44.31	107.9	0.1738	0.0091	161	39
2	21	129.97	91.74	28.87	4.31	964.17	-10885.48	0.0677	0.2250	601	7	26	45.23	107.9	0.2170	0.0677	133	33
3	22	136.10	93.74	28.87	4.31	1514.03	-10183.17	0.0500	0.2250	591	6	26	50.59	107.9	0.1420	0.0147	156	37
3	23	142.28	91.74	26.87	4.31	4720.78	-6335.07	0.0500	0.2250	599	7	26	46.18	107.9	0.1377	0.0234	333	83
3	24	148.46	91.74	28.87	2.31	1532.37	-10161.57	0.0500	0.3040	602	6	47	42.69	107.9	0.3040	0.0405	137	42
3	25	154.71	92.74	28.87	4.31	532.90	-11360.95	0.0500	0.3046	602	7	26	44.14	107.9	0.3046	0.0165	111	32
3	26	160.85	90.74	28.87	4.31	2424.56	-9090.53	0.0500	0.2250	581	6	26	56.69	107.9	0.1131	0.0000	190	53
3	27	166.98	91.74	29.87	4.31	598.18	-11282.19	0.0500	0.2250	601	7	26	52.86	107.9	0.1444	0.0000	113	33
3	28	173.11	91.74	27.87	4.31	1657.93	-10010.52	0.0500	0.2318	598	7	26	45.13	107.9	0.2318	0.0183	168	43
4	29	179.34	94.74	28.87	4.31	1731.07	-9992.32	0.0790	0.2250	596	6	26	40.97	107.9	0.2248	0.0790	168	45
4	30	185.52	92.74	26.87	4.31	1538.08	-10154.30	0.0500	0.2250	599	6	26	47.95	107.9	0.1371	0.0000	165	42

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
4	31	191.65	92.74	28.87	6.31	2045.65	-9545.22	0.0500	0.2250	567	6	18	45.83	107.9	0.1738	0.0091	162	53
4	32	197.79	92.74	28.87	2.31	5670.53	-5195.37	0.0500	0.2250	600	6	47	57.07	107.9	0.1112	0.0000	358	91
4	33	204.02	92.74	29.87	4.31	4958.23	-6050.12	0.0500	0.2250	587	6	26	57.64	107.9	0.1001	0.0000	337	78
4	34	210.14	92.74	27.87	4.31	1773.63	-9871.64	0.0500	0.2250	587	6	26	54.12	107.9	0.1444	0.0000	160	50
4	35	216.42	92.74	28.87	3.31	989.57	-10812.52	0.0500	0.2250	603	6	33	52.09	107.9	0.1742	0.0000	132	31
5	36	222.56	94.74	28.87	3.31	1116.22	-10660.54	0.0500	0.2250	600	6	33	45.09	107.9	0.1975	0.0298	135	33
5	37	228.71	90.74	28.87	3.31	886.72	-10935.94	0.0500	0.2250	596	7	33	55.75	107.9	0.1117	0.0000	111	38
5	38	234.84	92.74	26.87	3.31	2593.19	-8888.18	0.0500	0.2250	588	6	33	50.57	107.9	0.1682	0.0000	200	50
5	39	241.13	92.74	28.87	1.31	1813.74	-9823.68	0.0500	0.2560	603	7	83	49.40	107.9	0.2560	0.0000	102	35
5	40	247.28	93.74	28.87	3.31	624.51	-11250.58	0.0500	0.2250	605	6	33	56.57	107.9	0.1591	0.0000	116	24
5	41	253.46	92.74	29.87	3.31	3111.00	-8266.80	0.0500	0.2250	594	6	33	53.75	107.9	0.1204	0.0000	235	57
5	42	259.59	92.74	27.87	3.31	1409.06	-10309.13	0.0500	0.2250	576	6	33	48.36	107.9	0.1886	0.0000	126	26
6	43	265.82	95.74	28.87	3.31	1147.23	-10623.33	0.0500	0.2250	591	6	33	50.21	107.9	0.1317	0.0000	123	38
6	44	272.01	93.74	26.87	3.31	733.96	-11119.24	0.0500	0.2250	591	6	33	59.68	107.9	0.0962	0.0003	103	28
6	45	278.81	93.74	28.87	1.31	2020.82	-9575.01	0.0500	0.2250	610	6	83	57.89	107.9	0.1495	0.0000	124	38
6	46	285.03	93.74	29.87	3.31	1624.10	-10051.08	0.0500	0.2250	603	6	33	55.43	107.9	0.1565	0.0000	162	47
6	47	291.17	93.74	27.87	3.31	5719.45	-5136.67	0.0500	0.2250	592	6	33	54.07	107.9	0.1009	0.0126	372	91
6	48	297.35	93.74	28.87	2.31	1444.87	-10266.16	0.0500	0.2250	606	6	47	55.24	107.9	0.1222	0.0000	140	35
7	49	303.49	95.74	26.87	3.31	1501.40	-10198.32	0.0500	0.2250	590	6	33	50.23	107.9	0.1317	0.0000	144	36
7	50	309.61	91.74	26.87	3.31	5515.25	-5381.70	0.0500	0.2250	603	6	33	53.98	107.9	0.0811	0.0126	372	94
7	51	315.89	93.74	24.87	3.31	1231.60	-10522.08	0.0500	0.2250	591	6	33	53.11	107.9	0.1474	0.0000	128	38
7	52	322.03	93.74	26.87	5.31	895.69	-10925.17	0.0500	0.2250	593	6	21	45.94	107.9	0.1321	0.0000	128	39
7	53	328.36	93.74	26.87	1.31	3353.68	-7975.59	0.0500	0.2250	602	6	83	56.56	107.9	0.1498	0.0000	190	46
7	54	334.49	94.74	26.87	3.31	1718.33	-9938.01	0.0500	0.2250	580	6	33	50.95	107.9	0.1998	0.0000	145	35
7	55	340.82	93.74	25.87	3.31	1556.95	-10131.66	0.0500	0.2250	591	6	33	54.08	107.9	0.0989	0.0000	142	49
7	56	346.97	93.74	26.87	4.31	1299.69	-10440.41	0.0500	0.2318	590	6	26	45.38	107.9	0.2318	0.0183	141	39
7	57	353.10	93.74	26.87	2.31	1091.56	-10690.13	0.0500	0.2250	601	6	47	57.63	107.9	0.1296	0.0000	112	36
8	58	359.21	95.74	26.87	5.31	910.67	-10907.20	0.0500	0.2250	586	6	21	52.83	107.9	0.1152	0.0000	120	39
8	59	365.35	93.74	24.87	5.31	1787.73	-9854.73	0.0500	0.2250	587	6	21	46.97	107.9	0.1689	0.0000	169	49
8	60	371.49	93.74	26.87	7.31	1529.03	-10165.17	0.0500	0.2250	567	6	15	45.95	107.9	0.1087	0.0319	138	48

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
8	61	377.68	94.74	26.87	5.31	507.37	-11391.15	0.0500	0.2250	587	6	21	55.25	107.9	0.1136	0.0000	101	31
8	62	383.87	92.74	26.87	5.31	1075.72	-10709.13	0.0500	0.2250	592	6	21	48.35	107.9	0.1604	0.0118	138	38
8	63	390.02	93.74	27.87	5.31	960.70	-10847.16	0.0500	0.2250	581	6	21	41.33	107.9	0.2212	0.0269	120	33
8	64	396.14	93.74	25.87	5.31	1079.92	-10704.09	0.0500	0.2250	587	6	21	48.44	107.9	0.1084	0.0000	131	40
8	65	402.49	93.74	26.87	6.31	1710.44	-9947.47	0.0500	0.2250	569	6	18	40.61	107.9	0.1771	0.0000	152	38
9	66	408.59	96.74	26.87	5.31	1648.03	-10022.36	0.0500	0.2250	587	6	21	52.46	107.9	0.1084	0.0000	163	44
9	67	414.77	94.74	28.87	5.31	920.87	-10894.95	0.0500	0.2250	585	6	21	46.44	107.9	0.1352	0.0000	122	34
9	68	420.92	94.74	24.87	5.31	2517.35	-8979.18	0.0500	0.2250	572	6	21	48.25	107.9	0.0800	0.0000	193	51
9	69	427.05	94.74	26.87	7.31	2436.04	-9076.78	0.0500	0.2295	551	6	15	42.22	107.9	0.2295	0.0000	175	43
9	70	433.32	94.74	27.87	5.31	1921.50	-9694.20	0.0500	0.2250	581	6	21	48.19	107.9	0.1288	0.0000	170	48
9	71	439.45	94.74	25.87	5.31	2005.19	-9593.77	0.0500	0.2250	587	6	21	49.19	107.9	0.1084	0.0000	186	41
9	72	445.58	94.74	26.87	6.31	1751.21	-9898.54	0.0500	0.2250	578	6	18	49.72	107.9	0.1192	0.0000	160	49
9	73	451.73	94.74	26.87	4.31	1370.58	-10355.31	0.0500	0.2250	605	6	26	45.62	107.9	0.1722	0.0000	167	33
10	74	457.87	96.74	28.87	5.31	914.73	-10902.33	0.0500	0.2250	589	6	21	52.53	107.9	0.0738	0.0000	122	43
10	75	464.51	94.74	28.87	7.31	2123.80	-9451.44	0.0500	0.2250	574	6	15	42.99	107.9	0.1517	0.0256	181	51
10	76	470.61	95.74	28.87	5.31	2682.55	-8780.94	0.0500	0.2250	570	6	21	44.17	107.9	0.2165	0.0488	204	44
10	77	476.71	94.74	29.87	5.31	983.75	-10819.50	0.0500	0.2250	580	6	21	47.51	107.9	0.1801	0.0118	120	33
10	78	482.91	94.74	28.87	6.31	1775.80	-9869.04	0.0500	0.2250	576	6	18	48.56	107.9	0.1074	0.0000	160	47
11	79	489.09	98.74	28.87	5.31	1164.28	-10602.87	0.0500	0.2250	581	6	21	51.28	107.9	0.0894	0.0237	132	33
11	80	495.32	96.74	28.87	7.31	1968.14	-9638.24	0.0500	0.2250	572	6	15	50.43	107.9	0.0584	0.0000	171	47
11	81	501.47	96.74	28.87	3.31	1535.25	-10157.70	0.0500	0.2250	594	6	33	51.67	107.9	0.1317	0.0000	150	38
11	82	507.62	97.74	28.87	5.31	1161.64	-10606.03	0.0500	0.2250	584	6	21	55.54	107.9	0.1079	0.0000	135	34
11	83	513.80	96.74	29.87	5.31	296.62	-11644.06	0.0500	0.2250	589	6	21	48.52	107.9	0.1591	0.0023	94	25
11	84	519.90	96.74	27.87	5.31	1510.21	-10187.75	0.0500	0.2250	572	6	21	51.56	107.9	0.1420	0.0000	140	36
11	85	526.14	96.74	28.87	6.31	1627.94	-10046.47	0.0500	0.2250	577	6	18	42.93	107.9	0.1771	0.0000	157	38
11	86	532.28	96.74	28.87	4.31	1204.83	-10554.21	0.0500	0.2250	605	6	26	48.18	107.9	0.1722	0.0000	154	39
12	87	538.43	98.74	29.87	5.31	855.72	-10973.14	0.0500	0.2250	599	6	21	50.66	107.9	0.1721	0.0000	135	35
12	88	544.57	96.74	29.87	7.31	1820.55	-9815.34	0.0500	0.2250	577	6	15	45.55	107.9	0.1142	0.0379	171	42
12	89	550.71	96.74	29.87	3.31	1789.42	-9852.69	0.0500	0.2250	599	6	33	51.69	107.9	0.1317	0.0000	165	51
12	90	556.85	97.74	29.87	5.31	1043.67	-10747.60	0.0500	0.2250	592	6	21	51.91	107.9	0.1362	0.0000	140	30

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
12	91	562.99	95.74	29.87	5.31	567.67	-11318.79	0.0500	0.2250	586	6	21	49.13	107.9	0.1450	0.0159	105	28
12	92	569.18	96.74	29.87	6.31	1782.40	-9861.12	0.0500	0.2250	578	6	18	43.74	107.9	0.1771	0.0000	166	41
12	93	575.32	96.74	29.87	4.31	1188.84	-10573.39	0.0500	0.2250	585	6	26	50.07	107.9	0.1502	0.0000	129	37
13	94	581.51	93.74	29.87	5.31	1874.74	-9750.31	0.0500	0.2250	590	6	21	45.62	107.9	0.2027	0.0057	179	47
13	95	587.75	95.74	27.87	5.31	1322.12	-10413.46	0.0500	0.2250	587	6	21	49.91	107.9	0.1084	0.0000	148	35
13	96	593.80	95.74	29.87	7.31	1440.68	-10271.18	0.0500	0.2250	578	6	15	43.60	107.9	0.1142	0.0379	148	45
13	97	599.94	95.74	29.87	3.31	1167.45	-10599.06	0.0500	0.2250	595	6	33	60.10	107.9	0.0989	0.0000	134	28
13	98	606.22	95.74	29.87	6.31	1609.40	-10068.72	0.0500	0.2250	578	6	18	49.29	107.9	0.1880	0.0417	153	46
13	99	612.37	95.74	29.87	4.31	1391.42	-10330.30	0.0500	0.2250	607	6	26	57.22	107.9	0.1279	0.0000	163	48
14	100	618.51	99.74	29.87	5.31	1550.50	-10139.39	0.0500	0.2250	587	6	21	59.33	107.9	0.0959	0.0000	163	32
14	101	624.69	97.74	27.87	5.31	1935.25	-9677.70	0.0500	0.2250	582	6	21	51.66	107.9	0.1072	0.0000	173	46
14	102	630.83	97.74	29.87	7.31	2521.09	-8974.69	0.0500	0.2250	573	6	15	47.05	107.9	0.1573	0.0303	196	66
14	103	636.97	97.74	29.87	3.31	1056.68	-10731.98	0.0500	0.2250	597	6	33	53.33	107.9	0.1317	0.0000	127	34
14	104	643.26	97.74	29.87	6.31	1574.09	-10111.09	0.0500	0.2250	578	6	18	43.82	107.9	0.1771	0.0000	155	38
14	105	649.77	97.74	29.87	4.31	1397.83	-10322.60	0.0500	0.2250	591	6	26	51.90	107.9	0.1586	0.0000	150	35
15	106	656.40	98.74	27.87	5.31	3460.92	-7846.90	0.0500	0.2250	580	6	21	53.53	107.9	0.0894	0.0237	246	78
15	107	662.54	98.74	29.87	7.31	2139.74	-9432.31	0.0500	0.2250	571	6	15	52.57	107.9	0.0985	0.0000	177	53
15	108	668.68	98.74	29.87	3.31	486.22	-11416.53	0.0500	0.2250	600	6	33	54.37	107.9	0.1666	0.0165	101	25
15	109	674.95	98.74	29.87	6.31	1799.43	-9840.68	0.0500	0.2250	574	6	18	44.76	107.9	0.1771	0.0000	160	45
15	110	681.09	98.74	29.87	4.31	4654.10	-6415.25	0.0500	0.2565	595	6	26	45.75	107.9	0.2565	0.0000	327	82
16	111	687.28	98.74	27.87	3.31	824.25	-11010.90	0.0500	0.2250	595	6	33	55.84	107.9	0.1030	0.0000	113	29
16	112	693.67	98.74	29.87	1.31	1729.31	-9924.83	0.0500	0.2250	600	6	83	58.32	107.9	0.1868	0.0137	100	26
16	113	699.82	99.74	29.87	3.31	1799.39	-9840.73	0.0500	0.2250	601	6	33	56.24	107.9	0.1394	0.0000	166	55
16	114	706.02	98.74	28.87	3.31	1698.64	-10023.31	0.0757	0.2250	601	6	33	50.26	107.9	0.1931	0.0757	166	37
16	115	712.41	98.74	29.87	2.31	1037.18	-10796.66	0.0672	0.2250	594	6	47	54.08	107.9	0.2003	0.0672	105	22
17	116	718.56	96.74	27.87	3.31	1193.45	-10567.86	0.0500	0.2250	594	6	33	51.30	107.9	0.1317	0.0000	133	31
17	117	724.74	98.74	25.87	3.31	618.42	-11257.89	0.0500	0.2250	598	6	33	51.23	107.9	0.1456	0.0000	102	34
17	118	731.03	98.74	27.87	1.31	1941.76	-9669.89	0.0500	0.2250	601	6	83	58.19	107.9	0.1273	0.0096	111	32
17	119	737.17	99.74	27.87	3.31	358.50	-11569.79	0.0500	0.2250	597	6	33	58.40	107.9	0.1110	0.0000	90	24
17	120	743.32	97.74	27.87	3.31	910.09	-10907.89	0.0500	0.2250	590	6	33	51.23	107.9	0.1610	0.0030	113	27

Iteration	calls	CPU Time	Q	R	TBREV	Z	G1	G2	G3	TS	TO	TR	AvgInv	Runtime	BWT	CWT	LS	BO
17	121	749.52	98.74	26.87	3.31	1065.88	-10720.94	0.0500	0.2250	595	6	33	55.74	107.9	0.1394	0.0000	124	36
17	122	755.66	98.74	27.87	4.31	1824.32	-9810.82	0.0500	0.2250	585	6	26	56.03	107.9	0.1203	0.0000	166	39
17	123	761.80	98.74	27.87	2.31	3912.91	-7304.71	0.0500	0.2641	597	6	47	47.01	107.9	0.2641	0.0020	268	53
18	124	768.03	99.74	25.87	3.31	1198.01	-10562.39	0.0500	0.2250	587	6	33	52.39	107.9	0.1121	0.0000	122	36
18	125	774.27	99.74	27.87	5.31	683.83	-11179.40	0.0500	0.2250	593	6	21	59.31	107.9	0.1266	0.0039	120	29
18	126	780.62	99.74	27.87	1.31	1852.00	-9836.16	0.0744	0.2250	594	6	83	55.23	107.9	0.1819	0.0744	95	30
18	127	786.82	99.74	28.87	3.31	480.50	-11423.40	0.0500	0.2250	597	6	33	50.05	107.9	0.2244	0.0000	98	23
18	128	792.97	99.74	26.87	3.31	719.94	-11136.07	0.0500	0.2250	606	6	33	55.44	107.9	0.0995	0.0000	120	30
18	129	799.11	99.74	27.87	4.31	1179.15	-10585.03	0.0500	0.2250	582	6	26	57.86	107.9	0.0876	0.0052	129	28
18	130	805.25	99.74	27.87	2.31	3094.18	-8286.98	0.0500	0.2250	590	6	47	53.34	107.9	0.1725	0.0256	211	51
19	131	811.40	97.74	28.87	3.31	301.94	-11637.67	0.0500	0.2250	596	6	33	51.73	107.9	0.2218	0.0226	86	23
19	132	817.58	99.74	28.87	5.31	1060.10	-10727.88	0.0500	0.2250	596	6	21	51.25	107.9	0.1287	0.0000	142	38
19	133	823.92	99.74	28.87	1.31	5262.66	-5684.81	0.0500	0.2250	596	6	83	58.36	107.9	0.1538	0.0000	281	78
19	134	830.18	99.74	28.87	4.31	1137.48	-10635.03	0.0500	0.2250	586	6	26	57.86	107.9	0.0876	0.0052	129	33
19	135	836.74	99.74	28.87	2.31	787.74	-11054.71	0.0500	0.2250	594	6	47	58.59	107.9	0.1257	0.0000	91	24
20	136	842.99	97.74	26.87	3.31	1059.30	-10728.84	0.0500	0.2250	592	6	33	49.03	107.9	0.2202	0.0000	117	42
20	137	849.38	97.74	28.87	1.31	1765.18	-9881.79	0.0500	0.2250	608	6	83	65.37	107.9	0.0856	0.0000	107	36
20	138	855.72	97.74	28.87	4.31	761.30	-11086.44	0.0500	0.2250	588	6	26	54.58	107.9	0.1533	0.0000	110	31
20	139	861.87	97.74	28.87	2.31	1755.26	-9893.69	0.0500	0.2250	592	6	47	60.96	107.9	0.1033	0.0000	139	39
21	140	868.07	95.74	28.87	4.31	1450.58	-10259.30	0.0500	0.2250	588	6	26	47.96	107.9	0.1722	0.0000	148	38
21	141	874.21	97.74	26.87	4.31	2240.70	-9311.16	0.0500	0.2250	591	6	26	55.24	107.9	0.1484	0.0000	193	50
21	142	880.35	97.74	28.87	6.31	1877.34	-9747.19	0.0500	0.2250	564	6	18	52.85	107.9	0.1000	0.0000	156	37
21	143	886.54	98.74	28.87	4.31	804.54	-11034.55	0.0500	0.2250	584	6	26	58.97	107.9	0.0796	0.0000	109	28
21	144	892.78	97.74	27.87	4.31	1521.39	-10174.34	0.0500	0.2250	597	6	26	47.88	107.9	0.1651	0.0000	165	35
22	145	899.18	98.74	26.87	4.31	1750.07	-9899.92	0.0500	0.2250	587	6	26	58.11	107.9	0.1023	0.0052	164	39
22	146	905.32	98.74	28.87	6.31	1958.77	-9649.48	0.0500	0.2250	574	6	18	47.55	107.9	0.1491	0.0034	168	48
22	147	911.46	98.74	28.87	2.31	974.03	-10831.16	0.0500	0.2250	595	6	47	55.23	107.9	0.1747	0.0000	104	23
23	148	917.91	96.74	28.87	2.31	1077.12	-10707.46	0.0500	0.2250	595	6	47	63.81	107.9	0.0920	0.0000	109	25
23	149	924.07	98.74	26.87	2.31	2249.75	-9300.30	0.0500	0.2250	595	6	47	57.22	107.9	0.1039	0.0000	173	38
23	150	930.62	98.74	28.87	1.31	4284.25	-6858.91	0.0500	0.2250	597	6	83	60.46	107.9	0.1264	0.0000	233	59
24	151	936.83	99.74	26.87	2.31	3203.47	-8155.84	0.0500	0.2250	596	6	47	55.98	107.9	0.1171	0.0000	226	49
24	152	943.09	99.74	29.87	2.31	1425.95	-10289.05	0.0500	0.2593	592	6	47	48.55	107.9	0.2593	0.0020	125	28
25	153	949.44	99.74	26.87	4.31	911.82	-10905.91	0.0500	0.2435	587	6	26	44.23	107.9	0.2435	0.0000	115	37
25	154	955.59	99.74	28.87	6.31	2032.08	-9698.54	0.1071	0.2250	570	6	18	49.72	107.9	0.2223	0.1071	160	49
25	155	961.95	99.74	29.87	4.31	713.45	-11143.86	0.0500	0.2250	588	6	26	60.56	107.9	0.1023	0.0052	109	27
26	156	968.35	99.74	29.87	6.31	1653.56	-10015.73	0.0500	0.2250	570	6	18	56.23	107.9	0.1157	0.0000	151	35
27	157	974.86	99.74	25.87	4.31	845.24	-10985.81	0.0500	0.2435	583	6	26	44.45	107.9	0.2435	0.0000	109	31
27	158	981.01	99.74	27.87	6.31	1097.15	-10683.41	0.0500	0.2250	593	6	18	50.01	107.9	0.1276	0.0000	144	40
28	159	987.58	99.74	25.87	5.31	1108.22	-10670.14	0.0500	0.2250	574	6	21	46.02	107.9	0.1241	0.0000	119	35
28	160	993.72	99.74	27.87	7.31	2705.87	-8843.68	0.0878	0.2250	561	6	15	49.39	107.9	0.1744	0.0878	199	43
28	161	1000.03	99.74	26.87	5.31	849.28	-10980.86	0.0500	0.2250	593	6	21	55.93	107.9	0.0783	0.0203	124	41

Logistics Domain: TS-SO Run Resulting in the BSF (Run 16)

Iteration	calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	G1	G2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
1	1	8.63	9	4.54	4.92	6.08	11.84	316.16	1701.57	-0.0524	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	5.2406	2.6887
1	2	18.08	11	4.54	4.92	6.08	11.84	316.13	1701.57	-0.0738	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	7.3840	3.2675
1	3	26.92	7	4.54	4.92	6.08	11.84	316.16	1701.57	-0.0511	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	5.1104	2.9316
1	4	36.20	9	2.54	4.92	6.08	11.84	316.16	1701.57	-0.0496	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	4.9612	2.6733
1	5	45.63	9	4.54	2.92	6.08	11.84	408.73	2080.98	-0.0605	38.33	18.80	9.93	52.23	50.39	47.85	1.5894	1.6234	1.6697	6.0457	3.2459
1	6	59.17	9	4.54	4.92	8.08	11.84	293.92	1610.43	-0.0495	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	4.9525	2.5224
1	7	73.98	9	4.54	4.92	6.08	9.84	332.02	1766.55	-0.0466	23.28	18.80	11.89	51.18	50.39	49.41	1.5999	1.6234	1.6452	4.6599	2.7118
1	8	84.73	10	4.54	4.92	6.08	11.84	316.16	1701.57	-0.0532	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	5.3200	2.9997
1	9	94.59	8	4.54	4.92	6.08	11.84	316.17	1701.57	-0.0464	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	4.6369	2.5522
1	10	105.80	9	3.54	4.92	6.08	11.84	316.16	1701.57	-0.0510	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	5.1009	2.6847
1	11	118.41	9	4.54	3.92	6.08	11.84	351.17	1845.17	-0.0766	28.96	18.80	9.93	51.65	50.39	47.85	1.6080	1.6234	1.6697	7.6570	2.9584
1	12	129.89	9	4.54	4.92	7.08	11.84	304.19	1652.47	-0.0436	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	4.3566	2.4988
1	13	142.16	9	4.54	4.92	5.08	11.84	333.52	1772.70	-0.0517	23.28	22.34	9.93	51.18	50.61	47.85	1.5999	1.6244	1.6697	5.1652	3.0000
1	14	151.80	9	4.54	4.92	6.08	10.84	322.94	1729.39	-0.0541	23.28	18.80	10.85	51.18	50.39	47.92	1.5999	1.6234	1.6619	5.4138	3.4757
2	15	162.55	11	4.54	4.92	8.08	11.84	293.90	1610.43	-0.0694	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.9373	3.2764
2	16	173.52	7	4.54	4.92	8.08	11.84	293.92	1610.43	-0.0508	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	5.0794	2.5925
2	17	185.89	9	2.54	4.92	8.08	11.84	293.93	1610.43	-0.0456	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	4.5648	2.5007
2	18	198.48	9	4.54	2.92	8.08	11.84	386.50	1989.85	-0.0547	38.33	14.31	9.93	52.23	49.70	47.85	1.5894	1.6374	1.6697	5.4699	3.0520
2	19	207.63	9	4.54	4.92	8.08	9.84	309.78	1675.41	-0.0445	23.28	14.31	11.89	51.18	49.70	49.41	1.5999	1.6374	1.6452	4.4525	2.5707
2	20	217.16	10	4.54	4.92	8.08	11.84	293.92	1610.43	-0.0533	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	5.3281	3.1326
2	21	226.88	8	4.54	4.92	8.08	11.84	293.93	1610.43	-0.0454	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	4.5358	2.7283
2	22	236.19	9	3.54	4.92	8.08	11.84	293.92	1610.43	-0.0476	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	4.7586	2.5190
2	23	253.06	9	4.54	3.92	8.08	11.84	328.95	1754.04	-0.0622	28.96	14.31	9.93	51.65	49.70	47.85	1.6080	1.6374	1.6697	6.2154	3.2842
2	24	263.81	9	4.54	4.92	9.08	11.84	287.14	1582.70	-0.0637	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.3658	2.9279
2	25	274.80	9	4.54	4.92	8.08	10.84	300.71	1638.25	-0.0483	23.28	14.31	10.85	51.18	49.70	47.92	1.5999	1.6374	1.6619	4.8292	2.9085
3	26	284.06	11	4.54	4.92	9.08	11.84	287.13	1582.70	-0.0687	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.8691	3.5825
3	27	293.34	7	4.54	4.92	9.08	11.84	287.18	1582.70	-0.0283	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	2.8285	2.1794
3	28	304.63	9	2.54	4.92	9.08	11.84	287.15	1582.70	-0.0529	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.2901	2.6287
3	29	317.22	9	4.54	2.92	9.08	11.84	379.73	1962.11	-0.0577	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	5.7703	3.1589
3	30	328.88	9	4.54	4.92	9.08	9.84	303.01	1647.68	-0.0501	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	5.0121	2.8397
3	31	342.17	10	4.54	4.92	9.08	11.84	287.14	1582.70	-0.0578	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.7819	3.3502
3	32	353.03	8	4.54	4.92	9.08	11.84	287.16	1582.70	-0.0421	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	4.2072	2.3010
3	33	362.81	9	3.54	4.92	9.08	11.84	287.14	1582.70	-0.0583	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.8279	2.7381
3	34	374.03	9	4.54	3.92	9.08	11.84	322.18	1726.30	-0.0595	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	5.9514	2.9226
3	35	386.09	9	4.54	4.92	9.08	10.84	293.93	1610.52	-0.0595	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	5.9509	2.8295
4	36	397.50	13	4.54	4.92	9.08	11.84	287.13	1582.70	-0.0724	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	7.2369	3.7174
4	37	405.97	11	2.54	4.92	9.08	11.84	287.14	1582.70	-0.0618	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.1814	3.4506
4	38	414.78	11	4.54	2.92	9.08	11.84	379.73	1962.11	-0.0573	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	5.7312	3.6532
4	39	424.03	11	4.54	4.92	7.08	11.84	304.17	1652.47	-0.0563	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	5.6252	3.6680
4	40	433.88	11	4.54	4.92	9.08	9.84	303.00	1647.68	-0.0593	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	5.9343	2.9672
4	41	447.48	12	4.54	4.92	9.08	11.84	287.14	1582.70	-0.0612	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.1217	3.2384
4	42	460.06	11	3.54	4.92	9.08	11.84	287.14	1582.70	-0.0653	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.5252	3.5153
4	43	471.48	11	4.54	3.92	9.08	11.84	322.17	1726.30	-0.0689	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	6.8859	3.7523
4	44	482.47	11	4.54	4.92	9.08	10.84	293.93	1610.52	-0.0646	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.4585	3.4709
5	45	491.92	15	4.54	4.92	9.08	11.84	287.11	1582.70	-0.0907	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	9.0708	4.2535

Iteration	calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	G1	G2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
5	47	515.39	13	4.54	2.92	9.08	11.84	379.69	1962.11	-0.0939	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	9.3856	4.9490
5	48	525.28	13	4.54	4.92	7.08	11.84	304.15	1652.47	-0.0741	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.4113	3.2274
5	49	536.16	13	4.54	4.92	9.08	9.84	302.97	1647.68	-0.0876	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	8.7617	4.3347
5	50	548.88	14	4.54	4.92	9.08	11.84	287.12	1582.70	-0.0800	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	7.9953	3.9535
5	51	558.14	13	3.54	4.92	9.08	11.84	287.13	1582.70	-0.0689	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.8904	3.7062
5	52	570.88	13	4.54	3.92	9.08	11.84	322.13	1726.30	-0.1037	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	10.3674	5.2656
5	53	583.30	13	4.54	4.92	8.08	11.84	293.89	1610.43	-0.0753	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.5324	3.4971
5	54	593.14	13	4.54	4.92	9.08	10.84	293.93	1610.52	-0.0629	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.2856	3.3788
6	55	602.25	15	2.54	4.92	9.08	11.84	287.11	1582.70	-0.0841	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	8.4069	4.2392
6	56	611.42	15	4.54	2.92	9.08	11.84	379.69	1962.11	-0.0927	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	9.2726	4.1339
6	57	624.39	15	4.54	4.92	7.08	11.84	304.13	1652.47	-0.0960	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	9.5991	4.2719
6	58	636.88	15	4.54	4.92	9.08	9.84	302.99	1647.68	-0.0698	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.9841	4.8897
6	59	648.22	15	3.54	4.92	9.08	11.84	287.11	1582.70	-0.0874	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	8.7388	4.2499
6	60	657.83	15	4.54	3.92	9.08	11.84	322.16	1726.30	-0.0824	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	8.2355	4.6364
6	61	671.38	15	4.54	4.92	8.08	11.84	293.89	1610.43	-0.0745	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.4524	3.9055
6	62	680.77	15	4.54	4.92	9.08	10.84	293.89	1610.52	-0.0913	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	9.1291	5.4435
7	63	690.47	15	1.54	4.92	9.08	11.84	287.12	1582.70	-0.0807	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	8.0749	4.2267
7	64	703.34	15	3.54	2.92	9.08	11.84	379.69	1962.11	-0.0882	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	8.8224	4.1542
7	65	714.11	15	3.54	4.92	7.08	11.84	304.13	1652.47	-0.0894	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	8.9438	4.0684
7	66	727.47	15	3.54	4.92	9.08	9.84	302.99	1647.68	-0.0675	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.7465	4.8588
7	67	738.59	14	3.54	4.92	9.08	11.84	287.12	1582.70	-0.0755	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	7.5520	3.9921
7	68	751.61	15	3.54	3.92	9.08	11.84	322.16	1726.30	-0.0794	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.9360	4.4716
7	69	764.05	15	3.54	4.92	8.08	11.84	293.89	1610.43	-0.0743	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.4257	3.8998
7	70	773.44	15	3.54	4.92	9.08	10.84	293.90	1610.52	-0.0846	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	8.4627	5.1559
8	71	782.81	13	1.54	4.92	9.08	11.84	287.14	1582.70	-0.0620	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.1975	3.5846
8	72	792.14	15	1.54	2.92	9.08	11.84	379.70	1962.11	-0.0792	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	7.9221	4.2492
8	73	801.88	15	1.54	4.92	7.08	11.84	304.15	1652.47	-0.0763	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.6331	4.0931
8	74	818.88	15	1.54	4.92	9.08	9.84	303.00	1647.68	-0.0627	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.2711	4.7608
8	75	828.73	14	1.54	4.92	9.08	11.84	287.13	1582.70	-0.0667	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.6653	4.0162
8	76	839.92	15	1.54	3.92	9.08	11.84	322.17	1726.30	-0.0734	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.3370	4.3211
8	77	853.08	15	1.54	4.92	8.08	11.84	293.89	1610.43	-0.0737	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.3724	3.8879
8	78	862.80	15	1.54	4.92	9.08	10.84	293.92	1610.52	-0.0713	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	7.1299	4.4984
9	79	872.59	15	2.54	2.92	9.08	11.84	379.70	1962.11	-0.0837	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	8.3723	4.2059
9	80	885.61	15	2.54	4.92	7.08	11.84	304.14	1652.47	-0.0829	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	8.2884	4.0187
9	81	896.47	15	2.54	4.92	9.08	9.84	302.99	1647.68	-0.0651	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.5088	4.8159
9	82	909.47	14	2.54	4.92	9.08	11.84	287.13	1582.70	-0.0711	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	7.1086	4.0286
9	83	918.92	15	2.54	3.92	9.08	11.84	322.16	1726.30	-0.0764	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.6365	4.3424
9	84	929.63	15	2.54	4.92	8.08	11.84	293.89	1610.43	-0.0740	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.3991	3.8939
9	85	942.58	15	2.54	4.92	9.08	10.84	293.91	1610.52	-0.0780	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	7.7963	4.8249
10	86	951.78	12	2.54	4.92	9.08	11.84	287.14	1582.70	-0.0606	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.0597	3.2402
10	87	961.28	14	2.54	2.92	9.08	11.84	379.72	1962.11	-0.0664	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	6.6402	4.0729
10	88	971.92	14	2.54	4.92	7.08	11.84	304.16	1652.47	-0.0681	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	6.8123	3.6449
10	89	981.42	14	2.54	4.92	9.08	9.84	303.00	1647.68	-0.0625	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.2538	4.2986
10	90	995.30	14	2.54	3.92	9.08	11.84	322.17	1726.30	-0.0755	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.5466	4.0244

Iteration	calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	G1	G2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
10	91	1008.30	14	2.54	4.92	8.08	11.84	293.90	1610.43	-0.0712	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.1246	3.5710
10	92	1019.81	14	2.54	4.92	9.08	10.84	293.92	1610.52	-0.0667	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.6664	3.7094
11	93	1032.50	14	4.54	2.92	9.08	11.84	379.72	1962.11	-0.0687	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	6.8706	4.1306
11	94	1045.55	14	4.54	4.92	7.08	11.84	304.16	1652.47	-0.0705	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.0505	3.7086
11	95	1056.41	14	4.54	4.92	9.08	9.84	302.99	1647.68	-0.0667	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.6746	4.4395
11	96	1069.28	14	4.54	3.92	9.08	11.84	322.15	1726.30	-0.0860	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	8.5981	4.2680
11	97	1078.52	14	4.54	4.92	8.08	11.84	293.89	1610.43	-0.0742	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.4235	3.6960
11	98	1089.59	14	4.54	4.92	9.08	10.84	293.92	1610.52	-0.0697	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.9652	3.7964
12	99	1101.25	12	3.54	4.92	9.08	11.84	287.14	1582.70	-0.0609	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.0907	3.2395
12	100	1113.75	14	3.54	2.92	9.08	11.84	379.72	1962.11	-0.0676	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	6.7554	4.1024
12	101	1126.59	14	3.54	4.92	7.08	11.84	304.16	1652.47	-0.0693	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	6.9314	3.6789
12	102	1135.78	14	3.54	4.92	9.08	9.84	302.99	1647.68	-0.0646	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.4642	4.3747
12	103	1144.83	14	3.54	3.92	9.08	11.84	322.16	1726.30	-0.0807	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	8.0723	4.1483
12	104	1154.33	14	3.54	4.92	8.08	11.84	293.90	1610.43	-0.0727	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.2741	3.5936
12	105	1163.75	14	3.54	4.92	9.08	10.84	293.92	1610.52	-0.0682	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.8158	3.7525
13	106	1173.30	13	3.54	2.92	9.08	11.84	379.69	1962.11	-0.0874	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	8.7447	4.4402
13	107	1187.09	13	3.54	4.92	7.08	11.84	304.15	1652.47	-0.0735	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.3460	3.2451
13	108	1197.34	13	3.54	4.92	9.08	9.84	302.97	1647.68	-0.0818	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	8.1773	4.1232
13	109	1210.28	13	3.54	3.92	9.08	11.84	322.14	1726.30	-0.0946	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	9.4604	4.7493
13	110	1223.59	13	3.54	4.92	8.08	11.84	293.90	1610.43	-0.0725	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.2482	3.4262
13	111	1234.89	13	3.54	4.92	9.08	10.84	293.93	1610.52	-0.0625	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.2548	3.3767
14	112	1246.08	13	2.54	2.92	9.08	11.84	379.70	1962.11	-0.0810	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	8.1038	3.9075
14	113	1259.02	13	2.54	4.92	7.08	11.84	304.15	1652.47	-0.0728	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.2807	3.2614
14	114	1272.06	13	2.54	4.92	9.08	9.84	302.98	1647.68	-0.0759	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	7.5930	3.9761
14	115	1281.94	13	2.54	3.92	9.08	11.84	322.15	1726.30	-0.0848	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	8.4782	4.1945
14	116	1294.45	13	2.54	4.92	8.08	11.84	293.90	1610.43	-0.0696	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.9641	3.3661
14	117	1303.33	13	2.54	4.92	9.08	10.84	293.93	1610.52	-0.0622	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.2240	3.3743
15	118	1312.75	11	2.54	2.92	9.08	11.84	379.74	1962.11	-0.0532	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	5.3195	3.5751
15	119	1322.11	11	2.54	4.92	7.08	11.84	304.18	1652.47	-0.0531	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	5.3075	3.6651
15	120	1331.44	11	2.54	4.92	9.08	9.84	303.00	1647.68	-0.0586	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	5.8567	2.9775
15	121	1341.53	10	2.54	4.92	9.08	11.84	287.15	1582.70	-0.0520	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.2015	3.2112
15	122	1354.61	11	1.54	4.92	9.08	11.84	287.14	1582.70	-0.0584	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.8375	3.4007
15	123	1364.06	11	2.54	3.92	9.08	11.84	322.18	1726.30	-0.0603	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	6.0286	3.2989
15	124	1378.69	11	2.54	4.92	8.08	11.84	293.91	1610.43	-0.0635	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.3456	3.4301
15	125	1391.56	11	2.54	4.92	9.08	10.84	293.93	1610.52	-0.0584	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	5.8351	3.2327
18	126	1402.20	12	4.54	2.92	9.08	11.84	379.68	1962.11	-0.1038	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	10.3842	3.8364
18	127	1413.44	12	4.54	4.92	7.08	11.84	304.14	1652.47	-0.0803	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	8.0261	3.3986
18	128	1426.28	12	4.54	4.92	9.08	9.84	302.99	1647.68	-0.0634	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.3433	3.3722
18	129	1437.22	12	4.54	3.92	9.08	11.84	322.14	1726.30	-0.0935	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	9.3518	4.8033
18	130	1448.20	12	4.54	4.92	8.08	11.84	293.89	1610.43	-0.0757	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.5699	3.1108
18	131	1460.50	12	4.54	4.92	9.08	10.84	293.90	1610.52	-0.0837	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	8.3678	3.6235
19	132	1472.66	10	3.54	4.92	9.08	11.84	287.15	1582.70	-0.0549	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	5.4917	3.2588
19	133	1485.30	12	1.54	4.92	9.08	11.84	287.14	1582.70	-0.0603	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	6.0287	3.2406
19	134	1494.91	12	3.54	2.92	9.08	11.84	379.69	1962.11	-0.0950	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	9.5029	3.5816
19	135	1504.25	12	3.54	4.92	7.08	11.84	304.15	1652.47	-0.0777	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.7652	3.3661

Iteration	calls	CPU Time	FAIL	ARRIVE	DISC1	DISC2	DISC3	Z	G1	G2	ORD1_AVG	ORD2_AVG	ORD3_AVG	OUT1_AVG	OUT2_AVG	OUT3_AVG	OUT1_STD	OUT2_STD	OUT3_STD	UTIL_AVG	UTIL_STD
19	136	1513.50	12	3.54	4.92	9.08	9.84	303.00	1647.68	-0.0610	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.1019	3.2815
19	137	1523.38	12	3.54	3.92	9.08	11.84	322.15	1726.30	-0.0868	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	8.6791	4.2496
19	138	1534.38	12	3.54	4.92	8.08	11.84	293.90	1610.43	-0.0724	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	7.2420	3.1212
19	139	1543.95	12	3.54	4.92	9.08	10.84	293.91	1610.52	-0.0791	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	7.9069	3.5939
20	140	1553.59	11	3.54	2.92	9.08	11.84	379.73	1962.11	-0.0553	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	5.5253	3.6161
20	141	1570.97	11	3.54	4.92	7.08	11.84	304.17	1652.47	-0.0547	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	5.4664	3.6700
20	142	1583.80	11	3.54	4.92	9.08	9.84	303.00	1647.68	-0.0590	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	5.8955	2.9726
20	143	1595.72	11	3.54	3.92	9.08	11.84	322.18	1726.30	-0.0646	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	6.4572	3.5186
20	144	1608.44	11	3.54	4.92	8.08	11.84	293.90	1610.43	-0.0666	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.6573	3.3599
20	145	1621.42	11	3.54	4.92	9.08	10.84	293.93	1610.52	-0.0615	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.1468	3.3507
21	146	1632.56	9	1.54	4.92	9.08	11.84	287.16	1582.70	-0.0475	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	4.7523	2.6447
21	147	1643.56	11	1.54	2.92	9.08	11.84	379.74	1962.11	-0.0511	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	5.1137	3.5345
21	148	1654.91	11	1.54	4.92	7.08	11.84	304.18	1652.47	-0.0515	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	5.1487	3.6533
21	149	1667.66	11	1.54	4.92	9.08	9.84	303.00	1647.68	-0.0582	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	5.8179	2.9819
21	150	1693.11	10	1.54	4.92	9.08	11.84	287.16	1582.70	-0.0491	23.28	12.88	9.93	51.18	50.21	47.85	1.5999	1.6533	1.6697	4.9113	3.1362
21	151	1705.83	11	1.54	3.92	9.08	11.84	322.19	1726.30	-0.0560	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	5.5999	3.2407
21	152	1715.41	11	1.54	4.92	8.08	11.84	293.91	1610.43	-0.0603	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.0339	3.4710
21	153	1724.72	11	1.54	4.92	9.08	10.84	293.94	1610.52	-0.0552	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	5.5234	3.1325
22	154	1734.47	13	1.54	2.92	9.08	11.84	379.71	1962.11	-0.0746	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	7.4629	3.6728
22	155	1744.06	13	1.54	4.92	7.08	11.84	304.15	1652.47	-0.0722	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	7.2153	3.2764
22	156	1758.75	13	1.54	4.92	9.08	9.84	302.99	1647.68	-0.0701	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	7.0086	3.7957
22	157	1768.38	13	1.54	3.92	9.08	11.84	322.17	1726.30	-0.0750	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.4960	3.6932
22	158	1781.48	13	1.54	4.92	8.08	11.84	293.90	1610.43	-0.0668	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.6799	3.3316
22	159	1792.27	13	1.54	4.92	9.08	10.84	293.93	1610.52	-0.0619	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.1932	3.3716
23	160	1805.06	14	1.54	2.92	9.08	11.84	379.72	1962.11	-0.0653	38.33	12.88	9.93	52.23	50.21	47.85	1.5894	1.6533	1.6697	6.5250	4.0610
23	161	1817.73	14	1.54	4.92	7.08	11.84	304.16	1652.47	-0.0669	23.28	16.32	9.93	51.18	50.62	47.85	1.5999	1.6317	1.6697	6.6932	3.6068
23	162	1827.45	14	1.54	4.92	9.08	9.84	303.00	1647.68	-0.0604	23.28	12.88	11.89	51.18	50.21	49.41	1.5999	1.6533	1.6452	6.0434	4.2106
23	163	1840.97	14	1.54	3.92	9.08	11.84	322.17	1726.30	-0.0702	28.96	12.88	9.93	51.65	50.21	47.85	1.6080	1.6533	1.6697	7.0208	3.9383
23	164	1853.75	14	1.54	4.92	8.08	11.84	293.90	1610.43	-0.0698	23.28	14.31	9.93	51.18	49.70	47.85	1.5999	1.6374	1.6697	6.9752	3.5731
23	165	1866.80	14	1.54	4.92	9.08	10.84	293.92	1610.52	-0.0652	23.28	12.88	10.85	51.18	50.21	47.92	1.5999	1.6533	1.6619	6.5170	3.6692
29	166	1880.50	15	1.54	2.92	8.08	11.84	386.47	1989.85	-0.0821	38.33	14.31	9.93	52.23	49.70	47.85	1.5894	1.6374	1.6697	8.2065	4.5061
29	167	1889.53	15	1.54	4.92	6.08	11.84	316.13	1701.57	-0.0732	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	7.3172	4.7462
29	168	1899.19	15	1.54	4.92	8.08	9.84	309.74	1675.41	-0.0781	23.28	14.31	11.89	51.18	49.70	49.41	1.5999	1.6374	1.6452	7.8087	4.4317
29	169	1909.05	15	1.54	3.92	8.08	11.84	328.94	1754.04	-0.0738	28.96	14.31	9.93	51.65	49.70	47.85	1.6080	1.6374	1.6697	7.3843	4.1594
29	170	1918.56	15	1.54	4.92	8.08	10.84	300.68	1638.25	-0.0764	23.28	14.31	10.85	51.18	49.70	47.92	1.5999	1.6374	1.6619	7.6387	3.9500
30	171	1929.52	15	3.54	2.92	8.08	11.84	386.46	1989.85	-0.0925	38.33	14.31	9.93	52.23	49.70	47.85	1.5894	1.6374	1.6697	9.2542	4.9103
30	172	1943.45	15	3.54	4.92	6.08	11.84	316.13	1701.57	-0.0778	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	7.7750	4.8195
30	173	1954.67	15	3.54	4.92	8.08	9.84	309.73	1675.41	-0.0876	23.28	14.31	11.89	51.18	49.70	49.41	1.5999	1.6374	1.6452	8.7587	4.5963
30	174	1968.25	15	3.54	3.92	8.08	11.84	328.93	1754.04	-0.0770	28.96	14.31	9.93	51.65	49.70	47.85	1.6080	1.6374	1.6697	7.7049	4.1575
30	175	1979.47	15	3.54	4.92	8.08	10.84	300.68	1638.25	-0.0790	23.28	14.31	10.85	51.18	49.70	47.92	1.5999	1.6374	1.6619	7.8982	3.9325
31	176	1992.20	15	4.54	2.92	8.08	11.84	386.45	1989.85	-0.0978	38.33	14.31	9.93	52.23	49.70	47.85	1.5894	1.6374	1.6697	9.7781	5.1741
31	177	2004.92	15	4.54	4.92	6.08	11.84	316.13	1701.57	-0.0800	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	8.0039	4.8464
31	178	2016.72	15	4.54	4.92	8.08	9.84	309.73	1675.41	-0.0923	23.28	14.31	11.89	51.18	49.70	49.41	1.5999	1.6374	1.6452	9.2337	4.6413
31	179	2029.66	15	4.54	3.92	8.08	11.84	328.93	1754.04	-0.0787	28.96	14.31	9.93	51.65	49.70	47.85	1.6080	1.6374	1.6697	7.8653	4.1551
31	180	2040.58	15	4.54	4.92	8.08	10.84	300.67	1638.25	-0.0803	23.28	14.31	10.85	51.18	49.70	47.92	1.5999	1.6374	1.6619	8.0280	3.9390
32	181	2053.45	13	4.54	2.92	8.08	11.84	386.48	1989.85	-0.0711	38.33	14.31	9.93	52.23	49.70	47.85	1.5894	1.6374	1.6697	7.1131	3.8567
32	182	2066.28	13	4.54	4.92	6.08	11.84	316.14	1701.57	-0.0647	23.28	18.80	9.93	51.18	50.39	47.85	1.5999	1.6234	1.6697	6.4691	3.7906
32	183	2076.17	13	4.54	4.92	8.08	9.84	309.75	1675.41	-0.0699	23.28	14.31	11.89	51.18	49.70	49.41	1.5999	1.6374	1.6452	6.9876	3.6471
32	184	2086.05	13	4.54	3.92	8.08	11.84	328.93	1754.04	-0.0811	28.96	14.31	9.93	51.65	49.70	47.85	1.6080	1.6374	1.6697	8.1126	5.4981
32	185	2095.45	13	4.54	4.92	8.08	10.84	300.69	1638.25	-0.0656	23.28	14.31	10.85	51.18	49.70	47.92	1.5999	1.6374	1.6619	6.5575	3.4924

PERT Domain: TS-SO Run Resulting in the BSF (Run 2)

Iteration	calls	CPU Time	LOW3	LOW4	LOW6	Z	G1	G2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
1	1	11.31	10.07	4.01	5.04	0.1480	21.76	20.71	0.5975	0.1950	0.2075	0.1675	0.4300	0.1750	0.4500	0.1675	0.6575	20.7096	0.1059
1	2	22.72	8.07	4.01	5.04	0.1446	21.76	20.65	0.6225	0.2050	0.1725	0.1800	0.4425	0.1800	0.4675	0.1800	0.6400	20.6491	0.1076
1	3	33.84	10.07	6.01	5.04	0.1519	21.76	20.78	0.6125	0.1875	0.2000	0.2025	0.4100	0.1700	0.4275	0.2025	0.6275	20.7803	0.1034
1	4	46.72	10.07	2.01	5.04	0.1453	21.76	20.66	0.5875	0.1975	0.2150	0.1400	0.4475	0.1750	0.4700	0.1400	0.6850	20.6610	0.1078
1	5	59.50	10.07	4.01	7.04	0.1512	21.76	20.77	0.6025	0.2000	0.1975	0.1675	0.4350	0.2125	0.4225	0.1675	0.6200	20.7682	0.1051
1	6	73.92	10.07	4.01	3.04	0.1457	21.76	20.67	0.5925	0.1925	0.2150	0.1725	0.4200	0.1500	0.4625	0.1725	0.6775	20.6682	0.1067
1	7	88.03	11.07	4.01	5.04	0.1503	21.76	20.75	0.5825	0.1900	0.2275	0.1625	0.4200	0.1675	0.4425	0.1625	0.6700	20.7511	0.1047
1	8	103.53	9.07	4.01	5.04	0.1461	21.76	20.68	0.6125	0.2025	0.1850	0.1725	0.4400	0.1800	0.4625	0.1725	0.6475	20.6763	0.1068
1	9	116.55	10.07	5.01	5.04	0.1498	21.76	20.74	0.6050	0.1925	0.2025	0.1775	0.4275	0.1750	0.4450	0.1775	0.6475	20.7422	0.1047
1	10	127.72	10.07	3.01	5.04	0.1465	21.76	20.68	0.5925	0.1975	0.2100	0.1500	0.4425	0.1750	0.4650	0.1500	0.6750	20.6831	0.1069
1	11	138.95	10.07	4.01	6.04	0.1494	21.76	20.74	0.5975	0.1975	0.2050	0.1675	0.4300	0.1900	0.4375	0.1675	0.6425	20.7357	0.1056
1	12	148.67	10.07	4.01	4.04	0.1467	21.76	20.69	0.5925	0.1925	0.2150	0.1675	0.4250	0.1600	0.4575	0.1675	0.6725	20.6870	0.1063
2	13	158.61	6.07	4.01	5.04	0.1424	21.76	20.61	0.6450	0.2100	0.1450	0.1900	0.4550	0.1850	0.4800	0.1900	0.6250	20.6081	0.1085
2	14	169.75	8.07	6.01	5.04	0.1488	21.76	20.73	0.6475	0.1975	0.1550	0.2250	0.4225	0.1750	0.4450	0.2250	0.6000	20.7252	0.1048
2	15	179.47	8.07	2.01	5.04	0.1417	21.76	20.60	0.6100	0.2075	0.1825	0.1475	0.4625	0.1800	0.4900	0.1475	0.6725	20.5968	0.1096
2	16	189.33	8.07	4.01	7.04	0.1481	21.76	20.71	0.6200	0.2125	0.1675	0.1775	0.4425	0.2200	0.4350	0.1775	0.6025	20.7118	0.1065
2	17	202.14	8.07	4.01	3.04	0.1422	21.76	20.61	0.6200	0.2025	0.1775	0.1850	0.4350	0.1525	0.4850	0.1850	0.6625	20.6050	0.1083
2	18	215.16	7.07	4.01	5.04	0.1434	21.76	20.63	0.6350	0.2100	0.1550	0.1875	0.4475	0.1850	0.4725	0.1875	0.6275	20.6273	0.1081
2	19	226.31	8.07	5.01	5.04	0.1466	21.76	20.68	0.6325	0.2025	0.1650	0.1925	0.4400	0.1800	0.4625	0.1925	0.6275	20.6843	0.1062
2	20	237.59	8.07	3.01	5.04	0.1431	21.76	20.62	0.6150	0.2075	0.1775	0.1600	0.4550	0.1800	0.4825	0.1600	0.6600	20.6206	0.1087
2	21	248.75	8.07	4.01	6.04	0.1462	21.76	20.68	0.6225	0.2100	0.1675	0.1800	0.4425	0.2000	0.4525	0.1800	0.6200	20.6769	0.1072
2	22	264.27	8.07	4.01	4.04	0.1433	21.76	20.63	0.6225	0.2050	0.1725	0.1800	0.4425	0.1700	0.4775	0.1800	0.6500	20.6251	0.1080
3	23	275.77	6.07	2.01	5.04	0.1392	21.76	20.55	0.6350	0.2125	0.1525	0.1600	0.4750	0.1850	0.5025	0.1600	0.6550	20.5518	0.1108
3	24	286.41	8.07	2.01	7.04	0.1453	21.76	20.66	0.6050	0.2200	0.1750	0.1375	0.4675	0.2300	0.4575	0.1375	0.6325	20.6619	0.1085
3	25	297.67	8.07	2.01	3.04	0.1392	21.76	20.55	0.6075	0.2050	0.1875	0.1475	0.4600	0.1575	0.5075	0.1475	0.6950	20.5520	0.1104
3	26	308.59	9.07	2.01	5.04	0.1433	21.76	20.63	0.6050	0.2050	0.1900	0.1450	0.4600	0.1800	0.4850	0.1450	0.6750	20.6259	0.1088
3	27	319.97	7.07	2.01	5.04	0.1404	21.76	20.57	0.6225	0.2125	0.1650	0.1550	0.4675	0.1850	0.4950	0.1550	0.6600	20.5721	0.1103
3	28	329.63	8.07	1.01	5.04	0.1406	21.76	20.58	0.6000	0.2150	0.1850	0.1350	0.4650	0.1800	0.5000	0.1350	0.6850	20.5765	0.1104
3	29	339.25	8.07	2.01	6.04	0.1433	21.76	20.62	0.6100	0.2125	0.1775	0.1475	0.4625	0.2000	0.4750	0.1475	0.6525	20.6246	0.1092
3	30	350.53	8.07	2.01	4.04	0.1404	21.76	20.57	0.6100	0.2075	0.1825	0.1475	0.4625	0.1700	0.5000	0.1475	0.6825	20.5727	0.1100
4	31	361.72	6.07	2.01	7.04	0.1431	21.76	20.62	0.6300	0.2250	0.1450	0.1475	0.4825	0.2375	0.4700	0.1475	0.6150	20.6209	0.1094
4	32	372.89	6.07	2.01	3.04	0.1367	21.76	20.51	0.6350	0.2100	0.1550	0.1600	0.4750	0.1650	0.5200	0.1600	0.6750	20.5054	0.1116
4	33	384.06	5.07	2.01	5.04	0.1383	21.76	20.53	0.6375	0.2125	0.1500	0.1600	0.4775	0.1875	0.5025	0.1600	0.6525	20.5343	0.1112
4	34	395.22	6.07	3.01	5.04	0.1407	21.76	20.58	0.6375	0.2125	0.1500	0.1700	0.4675	0.1850	0.4950	0.1700	0.6450	20.5774	0.1098

Iteration	calls	CPU Time	LOW3	LOW4	LOW6	Z	G1	G2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
4	35	406.83	6.07	1.01	5.04	0.1380	21.76	20.53	0.6225	0.2200	0.1575	0.1450	0.4775	0.1850	0.5125	0.1450	0.6700	20.5292	0.1118
4	36	418.22	6.07	2.01	6.04	0.1409	21.76	20.58	0.6350	0.2200	0.1450	0.1575	0.4775	0.2100	0.4875	0.1575	0.6325	20.5811	0.1103
4	37	430.72	6.07	2.01	4.04	0.1378	21.76	20.53	0.6350	0.2125	0.1525	0.1600	0.4750	0.1750	0.5125	0.1600	0.6650	20.5268	0.1112
5	38	440.64	6.07	4.01	3.04	0.1398	21.76	20.56	0.6450	0.2075	0.1475	0.1950	0.4500	0.1600	0.4975	0.1950	0.6450	20.5624	0.1094
5	39	454.92	6.07	2.01	1.04	0.1348	21.76	20.47	0.6325	0.2050	0.1625	0.1675	0.4650	0.1400	0.5300	0.1675	0.6925	20.4722	0.1120
5	40	465.00	7.07	2.01	3.04	0.1378	21.76	20.53	0.6225	0.2075	0.1700	0.1550	0.4675	0.1625	0.5125	0.1550	0.6825	20.5267	0.1111
5	41	475.95	5.07	2.01	3.04	0.1357	21.76	20.49	0.6375	0.2125	0.1500	0.1600	0.4775	0.1675	0.5225	0.1600	0.6725	20.4872	0.1120
5	42	486.92	6.07	3.01	3.04	0.1381	21.76	20.53	0.6375	0.2100	0.1525	0.1725	0.4650	0.1625	0.5125	0.1725	0.6650	20.5312	0.1106
5	43	496.69	6.07	1.01	3.04	0.1354	21.76	20.48	0.6225	0.2175	0.1600	0.1450	0.4775	0.1650	0.5300	0.1450	0.6900	20.4828	0.1125
5	44	506.45	6.07	2.01	2.04	0.1357	21.76	20.49	0.6350	0.2075	0.1575	0.1625	0.4725	0.1525	0.5275	0.1625	0.6850	20.4875	0.1118
6	45	516.72	8.07	2.01	1.04	0.1375	21.76	20.52	0.6075	0.2050	0.1875	0.1550	0.4525	0.1400	0.5175	0.1550	0.7050	20.5197	0.1108
6	46	528.03	6.07	4.01	1.04	0.1381	21.76	20.53	0.6425	0.2025	0.1550	0.1975	0.4450	0.1400	0.5075	0.1975	0.6625	20.5306	0.1097
6	47	538.03	7.07	2.01	1.04	0.1360	21.76	20.49	0.6200	0.2050	0.1750	0.1625	0.4575	0.1400	0.5225	0.1625	0.6975	20.4940	0.1115
6	48	547.59	5.07	2.01	1.04	0.1338	21.76	20.45	0.6350	0.2075	0.1575	0.1675	0.4675	0.1425	0.5325	0.1675	0.6900	20.4531	0.1124
6	49	558.92	6.07	3.01	1.04	0.1363	21.76	20.50	0.6350	0.2050	0.1600	0.1775	0.4575	0.1400	0.5225	0.1775	0.6825	20.4988	0.1109
6	50	569.89	6.07	1.01	1.04	0.1335	21.76	20.45	0.6200	0.2125	0.1675	0.1475	0.4725	0.1450	0.5400	0.1475	0.7075	20.4492	0.1129
7	51	582.95	8.07	1.01	1.04	0.1363	21.76	20.50	0.5975	0.2125	0.1900	0.1375	0.4600	0.1450	0.5275	0.1375	0.7175	20.4990	0.1116
7	52	595.44	7.07	1.01	1.04	0.1348	21.76	20.47	0.6100	0.2125	0.1775	0.1450	0.4650	0.1450	0.5325	0.1450	0.7100	20.4723	0.1123
7	53	608.52	5.07	1.01	1.04	0.1325	21.76	20.43	0.6275	0.2150	0.1575	0.1525	0.4750	0.1475	0.5425	0.1525	0.7000	20.4299	0.1133
7	54	621.58	6.07	1.01	2.04	0.1344	21.76	20.46	0.6225	0.2150	0.1625	0.1475	0.4750	0.1525	0.5375	0.1475	0.7000	20.4647	0.1128
8	55	631.69	5.07	3.01	1.04	0.1353	21.76	20.48	0.6400	0.2075	0.1525	0.1800	0.4600	0.1425	0.5250	0.1800	0.6775	20.4803	0.1113
8	56	646.66	5.07	1.01	3.04	0.1344	21.76	20.46	0.6300	0.2200	0.1500	0.1500	0.4800	0.1675	0.5325	0.1500	0.6825	20.4645	0.1129
8	57	658.34	5.07	1.01	2.04	0.1334	21.76	20.45	0.6300	0.2175	0.1525	0.1525	0.4775	0.1550	0.5400	0.1525	0.6925	20.4459	0.1132
9	58	669.58	7.07	1.01	2.04	0.1357	21.76	20.49	0.6100	0.2125	0.1775	0.1450	0.4650	0.1475	0.5300	0.1450	0.7075	20.4876	0.1121
9	59	681.17	5.07	3.01	2.04	0.1361	21.76	20.50	0.6425	0.2100	0.1475	0.1775	0.4650	0.1525	0.5225	0.1775	0.6700	20.4955	0.1112
9	60	691.19	5.07	1.01	4.04	0.1356	21.76	20.49	0.6300	0.2200	0.1500	0.1500	0.4800	0.1750	0.5250	0.1500	0.6750	20.4863	0.1125
9	61	702.45	5.07	2.01	2.04	0.1346	21.76	20.47	0.6375	0.2100	0.1525	0.1625	0.4750	0.1550	0.5300	0.1625	0.6825	20.4690	0.1122
10	62	715.66	8.07	1.01	2.04	0.1372	21.76	20.51	0.5975	0.2125	0.1900	0.1375	0.4600	0.1475	0.5250	0.1375	0.7150	20.5143	0.1114
10	63	728.52	6.07	3.01	2.04	0.1371	21.76	20.51	0.6375	0.2075	0.1550	0.1750	0.4625	0.1500	0.5200	0.1750	0.6750	20.5135	0.1108
10	64	741.70	6.07	1.01	4.04	0.1366	21.76	20.50	0.6225	0.2200	0.1575	0.1450	0.4775	0.1750	0.5225	0.1450	0.6800	20.5042	0.1122
11	65	754.86	8.07	1.01	3.04	0.1381	21.76	20.53	0.5975	0.2125	0.1900	0.1350	0.4625	0.1575	0.5175	0.1350	0.7075	20.5317	0.1111
11	66	768.14	7.07	1.01	3.04	0.1367	21.76	20.51	0.6125	0.2150	0.1725	0.1425	0.4700	0.1625	0.5225	0.1425	0.6950	20.5054	0.1119
12	67	779.22	5.07	3.01	3.04	0.1371	21.76	20.51	0.6425	0.2125	0.1450	0.1750	0.4675	0.1650	0.5150	0.1750	0.6600	20.5136	0.1109
12	68	790.42	5.07	1.01	5.04	0.1370	21.76	20.51	0.6300	0.2200	0.1500	0.1500	0.4800	0.1875	0.5125	0.1500	0.6625	20.5115	0.1121

Iteration	calls	CPU Time	LOW3	LOW4	LOW6	Z	G1	G2	CRIT1	CRIT2	CRIT3	CRIT4	CRIT5	CRIT6	CRIT7	CRIT8	CRIT9	PT_AVG	PT_STD
13	69	800.58	7.07	1.01	4.04	0.1378	21.76	20.53	0.6125	0.2150	0.1725	0.1425	0.4700	0.1700	0.5150	0.1425	0.6875	20.5263	0.1115
13	70	811.66	5.07	3.01	4.04	0.1383	21.76	20.54	0.6425	0.2125	0.1450	0.1725	0.4700	0.1750	0.5075	0.1725	0.6525	20.5352	0.1105
13	71	827.14	5.07	1.01	6.04	0.1388	21.76	20.54	0.6225	0.2325	0.1450	0.1425	0.4800	0.2200	0.4925	0.1425	0.6375	20.5432	0.1115
13	72	837.50	5.07	2.01	4.04	0.1369	21.76	20.51	0.6375	0.2125	0.1500	0.1600	0.4775	0.1750	0.5150	0.1600	0.6650	20.5091	0.1116
14	73	850.50	8.07	1.01	4.04	0.1393	21.76	20.55	0.6000	0.2150	0.1850	0.1350	0.4650	0.1700	0.5100	0.1350	0.6950	20.5525	0.1107
14	74	863.69	6.07	3.01	4.04	0.1393	21.76	20.55	0.6375	0.2125	0.1500	0.1700	0.4675	0.1750	0.5050	0.1700	0.6550	20.5524	0.1102
14	75	876.55	6.07	1.01	6.04	0.1397	21.76	20.56	0.6200	0.2300	0.1500	0.1400	0.4800	0.2175	0.4925	0.1400	0.6425	20.5601	0.1112
15	76	890.59	6.07	4.01	4.04	0.1410	21.76	20.58	0.6450	0.2100	0.1450	0.1900	0.4550	0.1750	0.4900	0.1900	0.6350	20.5831	0.1090
15	77	903.95	7.07	2.01	4.04	0.1390	21.76	20.55	0.6225	0.2075	0.1700	0.1550	0.4675	0.1700	0.5050	0.1550	0.6750	20.5475	0.1107
16	78	915.52	8.07	2.01	2.04	0.1383	21.76	20.53	0.6075	0.2050	0.1875	0.1500	0.4575	0.1475	0.5150	0.1500	0.7025	20.5349	0.1106
16	79	927.19	6.07	4.01	2.04	0.1389	21.76	20.55	0.6450	0.2050	0.1500	0.1975	0.4475	0.1475	0.5050	0.1975	0.6550	20.5453	0.1096
16	80	937.06	7.07	2.01	2.04	0.1369	21.76	20.51	0.6200	0.2050	0.1750	0.1575	0.4625	0.1475	0.5200	0.1575	0.6950	20.5092	0.1113
17	81	949.05	5.07	4.01	2.04	0.1379	21.76	20.53	0.6525	0.2075	0.1400	0.2025	0.4500	0.1500	0.5075	0.2025	0.6475	20.5284	0.1099
18	82	962.36	5.07	4.01	1.04	0.1831	21.77	20.51	0.6525	0.2025	0.1450	0.2050	0.4475	0.1400	0.5100	0.2050	0.6550	20.5133	0.1100
24	83	973.05	9.07	2.01	1.04	0.1392	21.76	20.55	0.6000	0.2025	0.1975	0.1525	0.4475	0.1375	0.5125	0.1525	0.7100	20.5506	0.1100
24	84	982.99	7.07	4.01	1.04	0.1392	21.76	20.55	0.6325	0.2025	0.1650	0.1950	0.4375	0.1400	0.5000	0.1950	0.6650	20.5512	0.1092
24	85	994.38	7.07	3.01	1.04	0.1375	21.76	20.52	0.6250	0.2050	0.1700	0.1750	0.4500	0.1400	0.5150	0.1750	0.6850	20.5202	0.1104
25	86	1004.22	9.07	1.01	1.04	0.1381	21.76	20.53	0.5850	0.2100	0.2050	0.1300	0.4550	0.1425	0.5225	0.1300	0.7275	20.5309	0.1107
26	87	1018.53	9.07	1.01	2.04	0.1389	21.76	20.55	0.5850	0.2100	0.2050	0.1300	0.4550	0.1450	0.5200	0.1300	0.7250	20.5458	0.1106
26	88	1029.25	7.07	3.01	2.04	0.1383	21.76	20.53	0.6250	0.2050	0.1700	0.1725	0.4525	0.1450	0.5125	0.1725	0.6825	20.5347	0.1103
27	89	1042.44	9.07	2.01	2.04	0.1400	21.76	20.57	0.6000	0.2025	0.1975	0.1475	0.4525	0.1450	0.5100	0.1475	0.7075	20.5654	0.1098
27	90	1055.34	7.07	4.01	2.04	0.1400	21.76	20.57	0.6325	0.2025	0.1650	0.1950	0.4375	0.1425	0.4975	0.1950	0.6625	20.5657	0.1091
36	91	1069.56	8.07	3.01	2.04	0.1396	21.76	20.56	0.6125	0.2050	0.1825	0.1650	0.4475	0.1450	0.5075	0.1650	0.6900	20.5592	0.1097
36	92	1084.00	6.07	5.01	2.04	0.1411	21.76	20.59	0.6550	0.2025	0.1425	0.2150	0.4400	0.1475	0.4950	0.2150	0.6375	20.5852	0.1080
37	93	1095.49	5.07	5.01	2.04	0.1402	21.76	20.57	0.6600	0.2050	0.1350	0.2200	0.4400	0.1500	0.4950	0.2200	0.6300	20.5689	0.1083
38	94	1105.64	5.07	5.01	1.04	0.1968	21.7725	20.55	0.6625	0.1975	0.1400	0.2275	0.4350	0.1375	0.4950	0.2275	0.6350	20.5541	0.1084
39	95	1115.47	8.07	3.01	1.04	0.1388	21.76	20.54	0.6125	0.2050	0.1825	0.1675	0.4450	0.1400	0.5100	0.1675	0.6925	20.5446	0.1099
39	96	1126.92	6.07	5.01	1.04	0.1403	21.76	20.57	0.6575	0.1975	0.1450	0.2225	0.4350	0.1375	0.4950	0.2225	0.6400	20.5708	0.1082
40	97	1138.41	8.07	4.01	1.04	0.1405	21.76	20.57	0.6200	0.2025	0.1775	0.1875	0.4325	0.1400	0.4950	0.1875	0.6725	20.5742	0.1087
40	98	1150.11	6.07	6.01	1.04	0.1891	21.77	20.62	0.6750	0.1875	0.1375	0.2575	0.4175	0.1300	0.4750	0.2575	0.6125	20.6218	0.1061
41	99	1161.91	8.07	4.01	2.04	0.1413	21.76	20.59	0.6200	0.2025	0.1775	0.1875	0.4325	0.1425	0.4925	0.1875	0.6700	20.5887	0.1086
41	100	1174.78	6.07	6.01	2.04	0.1439	21.76	20.64	0.6725	0.1925	0.1350	0.2500	0.4225	0.1400	0.4750	0.2500	0.6100	20.6353	0.1060
42	101	1188.58	5.07	6.01	2.04	0.4073	21.8175	20.62	0.6800	0.1950	0.1250	0.2575	0.4225	0.1425	0.4750	0.2575	0.6000	20.6206	0.1062
42	102	1205.83	5.07	4.01	4.04	0.1401	21.76	20.57	0.6525	0.2100	0.1375	0.1950	0.4575	0.1750	0.4925	0.1950	0.6300	20.5671	0.1093
42	103	1215.72	5.07	4.01	3.04	0.1389	21.76	20.55	0.6525	0.2100	0.1375	0.2000	0.4525	0.1625	0.5000	0.2000	0.6375	20.5459	0.1097
55	104	1228.99	9.07	3.01	1.04	0.1405	21.76	20.57	0.6025	0.2025	0.1950	0.1625	0.4400	0.1375	0.5050	0.1625	0.7000	20.5745	0.1091
55	105	1238.80	7.07	5.01	1.04	0.1414	21.76	20.59	0.6425	0.1975	0.1600	0.2150	0.4275	0.1375	0.4875	0.2150	0.6475	20.5899	0.1078
55	106	1250.08	7.07	3.01	3.04	0.1393	21.76	20.55	0.6275	0.2075	0.1650	0.1700	0.4575	0.1600	0.5050	0.1700	0.6700	20.5521	0.1101

Production Domain: TS-SO Run Resulting in the BSF (Run 20)

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
1	1	12.59	5.42	15.14	7.24	6.42	3.31	5.91	55.53	-2045.72	-1.90	20.63	57.37	0.9419	5.7742	0.0386	1.9827	0.0065
1	2	25.66	7.42	15.14	7.24	6.42	3.31	5.91	56.91	-1982.00	-1.90	19.99	57.93	0.9542	5.7742	0.0386	1.9780	0.0073
1	3	40.17	5.42	17.14	7.24	6.42	3.31	5.91	58.20	-1922.82	-1.90	19.40	58.22	0.9641	5.7742	0.0386	1.9151	0.0172
1	4	51.61	5.42	13.14	7.24	6.42	3.31	5.91	51.32	-2239.21	-1.90	22.57	57.20	0.9422	5.7742	0.0386	1.9459	0.0113
1	5	63.14	5.42	15.14	9.24	6.42	3.31	5.91	56.77	-1988.59	-1.90	20.06	58.31	0.9381	5.7742	0.0386	1.9827	0.0065
1	6	77.53	5.42	15.14	5.24	6.42	3.31	5.91	54.37	-2099.09	-1.90	21.17	56.28	0.9423	5.7742	0.0386	1.9827	0.0065
1	7	89.52	5.42	15.14	7.24	4.42	3.31	5.91	51.59	-2226.67	-1.90	22.45	55.60	0.9425	5.6732	0.0404	1.9208	0.0152
1	8	102.16	5.42	15.14	7.24	6.42	1.31	5.91	51.49	-2231.48	-1.90	22.49	56.19	0.9392	5.7742	0.0386	1.9725	0.0082
1	9	116.33	5.42	15.14	7.24	6.42	3.31	3.91	54.63	-2087.03	-1.90	21.05	56.20	0.9707	5.7742	0.0386	1.9371	0.0167
1	10	127.55	6.42	15.14	7.24	6.42	3.31	5.91	55.23	-2060.64	-1.85	20.78	57.71	0.9524	5.7742	0.0386	1.7707	0.0297
1	11	142.48	5.42	16.14	7.24	6.42	3.31	5.91	56.91	-1981.94	-1.90	19.99	57.72	0.9638	5.7403	0.0392	1.9365	0.0162
1	12	157.09	5.42	14.14	7.24	6.42	3.31	5.91	53.79	-2125.44	-1.90	21.43	57.62	0.9594	5.7742	0.0386	1.8816	0.0214
1	13	168.42	5.42	15.14	8.24	6.42	3.31	5.91	56.13	-2018.08	-1.90	20.35	57.83	0.9378	5.7742	0.0386	1.9827	0.0065
1	14	181.16	5.42	15.14	6.24	6.42	3.31	5.91	54.89	-2075.19	-1.90	20.93	56.83	0.9396	5.7742	0.0386	1.9827	0.0065
1	15	195.55	5.42	15.14	7.24	5.42	3.31	5.91	53.44	-2141.86	-1.89	21.59	56.81	0.9488	5.7475	0.0391	1.8629	0.0207
1	16	209.83	5.42	15.14	7.24	6.42	2.31	5.91	53.52	-2138.04	-1.90	21.56	56.45	0.9421	5.7083	0.0396	1.9115	0.0175
1	17	222.47	5.42	15.14	7.24	6.42	3.31	4.91	54.69	-2084.35	-1.90	21.02	56.51	0.9451	5.7742	0.0386	1.9577	0.0101
2	18	235.33	7.42	13.14	7.24	6.42	3.31	5.91	52.97	-2163.38	-1.90	21.81	57.54	0.9339	5.7500	0.0390	1.9318	0.0179
2	19	246.56	5.42	11.14	7.24	6.42	3.31	5.91	47.61	-2409.96	-1.90	24.28	55.42	0.9054	5.7098	0.0396	1.9067	0.0189
2	20	258.31	5.42	13.14	9.24	6.42	3.31	5.91	52.70	-2175.60	-1.90	21.93	58.15	0.9423	5.7742	0.0386	1.9459	0.0113
2	21	270.94	5.42	13.14	5.24	6.42	3.31	5.91	49.98	-2300.89	-1.90	23.19	56.04	0.9366	5.7742	0.0386	1.9459	0.0113
2	22	283.03	5.42	13.14	7.24	4.42	3.31	5.91	47.12	-2432.67	-1.90	24.51	54.09	0.9257	5.7742	0.0386	1.8944	0.0178
2	23	295.59	5.42	13.14	7.24	6.42	1.31	5.91	47.54	-2415.09	-1.82	24.34	54.15	0.9162	5.7742	0.0386	1.7158	0.0308
2	24	308.22	5.42	13.14	7.24	6.42	3.31	3.91	51.39	-2236.74	-1.88	22.55	55.25	0.9263	5.7013	0.0413	1.8552	0.0243
2	25	320.75	6.42	13.14	7.24	6.42	3.31	5.91	52.44	-2195.48	-1.58	22.13	56.52	0.9347	5.7742	0.0386	1.2435	0.0419
2	26	333.52	5.42	12.14	7.24	6.42	3.31	5.91	50.09	-2296.20	-1.89	23.14	55.85	0.9298	5.7742	0.0386	1.8544	0.0230
2	27	344.63	5.42	13.14	8.24	6.42	3.31	5.91	52.06	-2205.08	-1.90	22.22	57.66	0.9408	5.7742	0.0386	1.9459	0.0113
2	28	357.27	5.42	13.14	6.24	6.42	3.31	5.91	50.56	-2274.23	-1.90	22.92	56.67	0.9406	5.7742	0.0386	1.9459	0.0113
2	29	368.70	5.42	13.14	7.24	5.42	3.31	5.91	49.30	-2332.06	-1.90	23.50	55.07	0.9256	5.7742	0.0386	1.9757	0.0106
2	30	381.56	5.42	13.14	7.24	6.42	2.31	5.91	49.38	-2328.50	-1.90	23.47	55.51	0.9259	5.7742	0.0386	1.9734	0.0080
2	31	393.98	5.42	13.14	7.24	6.42	3.31	4.91	51.75	-2219.37	-1.90	22.37	55.95	0.9384	5.7742	0.0386	1.8905	0.0176
3	32	406.73	7.42	13.14	7.24	4.42	3.31	5.91	47.71	-2406.06	-1.87	24.24	54.82	0.9167	5.7742	0.0386	1.8226	0.0258
3	33	419.61	5.42	11.14	7.24	4.42	3.31	5.91	41.81	-2676.63	-1.90	26.96	52.67	0.8940	5.7742	0.0386	1.9947	0.0036
3	34	432.47	5.42	13.14	9.24	4.42	3.31	5.91	48.68	-2360.74	-1.90	23.79	55.01	0.9242	5.7742	0.0386	1.8944	0.0178
3	35	445.44	5.42	13.14	5.24	4.42	3.31	5.91	45.74	-2496.18	-1.90	25.15	53.01	0.9291	5.7742	0.0386	1.8944	0.0178
3	36	458.23	5.42	13.14	7.24	4.42	1.31	5.91	41.67	-2685.63	-1.80	27.05	51.95	0.9078	5.6733	0.0420	1.7089	0.0250
3	37	469.36	5.42	13.14	7.24	4.42	3.31	3.91	45.18	-2523.12	-1.84	25.42	53.23	0.9431	5.7742	0.0386	1.7640	0.0273
3	38	482.11	6.42	13.14	7.24	4.42	3.31	5.91	47.93	-2395.72	-1.87	24.14	54.36	0.9157	5.7742	0.0386	1.8225	0.0242
3	39	493.13	5.42	14.14	7.24	4.42	3.31	5.91	49.14	-2339.34	-1.90	23.58	54.68	0.9320	5.7742	0.0386	1.9187	0.0158
3	40	504.56	5.42	12.14	7.24	4.42	3.31	5.91	44.94	-2533.51	-1.87	25.52	53.83	0.9037	5.7742	0.0386	1.8218	0.0244
3	41	515.78	5.42	13.14	8.24	4.42	3.31	5.91	47.98	-2393.00	-1.90	24.11	54.54	0.9259	5.7742	0.0386	1.8944	0.0178
3	42	528.20	5.42	13.14	6.24	4.42	3.31	5.91	46.42	-2464.90	-1.90	24.84	53.57	0.9271	5.7742	0.0386	1.8944	0.0178
3	43	541.42	5.42	13.14	7.24	3.42	3.31	5.91	44.17	-2568.40	-1.90	25.87	52.99	0.9226	5.7125	0.0396	1.9235	0.0141
3	44	553.86	5.42	13.14	7.24	4.42	2.31	5.91	44.97	-2532.18	-1.87	25.51	52.68	0.9276	5.6816	0.0400	1.8541	0.0177
3	45	568.27	5.42	13.14	7.24	4.42	3.31	4.91	45.75	-2495.43	-1.90	25.14	53.49	0.9393	5.7742	0.0386	1.8926	0.0200

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
4	46	579.48	7.42	13.14	7.24	4.42	1.31	5.91	43.11	-2618.57	-1.82	26.37	53.44	0.9149	5.7742	0.0386	1.7237	0.0253
4	47	591.16	5.42	15.14	7.24	4.42	1.31	5.91	46.13	-2478.65	-1.87	24.97	53.17	0.9472	5.7482	0.0390	1.8316	0.0194
4	48	603.36	5.42	11.14	7.24	4.42	1.31	5.91	35.50	-2970.08	-1.78	29.90	49.97	0.8809	5.7742	0.0386	1.6347	0.0336
4	49	617.75	5.42	13.14	9.24	4.42	1.31	5.91	43.19	-2615.66	-1.80	26.35	53.06	0.9114	5.6733	0.0420	1.7089	0.0250
4	50	630.17	5.42	13.14	5.24	4.42	1.31	5.91	39.91	-2766.67	-1.80	27.86	50.77	0.9062	5.6733	0.0420	1.7089	0.0250
4	51	641.55	5.42	13.14	7.24	4.42	1.31	3.91	39.42	-2786.85	-1.90	28.07	50.29	0.9169	5.6978	0.0398	1.8936	0.0186
4	52	655.61	6.42	13.14	7.24	4.42	1.31	5.91	42.03	-2669.02	-1.79	26.88	52.11	0.9057	5.6886	0.0399	1.6881	0.0288
4	53	666.83	5.42	14.14	7.24	4.42	1.31	5.91	43.93	-2579.38	-1.89	25.98	52.61	0.9170	5.7742	0.0386	1.8472	0.0223
4	54	677.94	5.42	12.14	7.24	4.42	1.31	5.91	38.01	-2852.02	-1.87	28.72	51.18	0.9128	5.7598	0.0389	1.8266	0.0215
4	55	690.47	5.42	13.14	8.24	4.42	1.31	5.91	42.43	-2650.66	-1.80	26.70	52.54	0.9107	5.6733	0.0420	1.7089	0.0250
4	56	702.89	5.42	13.14	6.24	4.42	1.31	5.91	40.79	-2726.15	-1.80	27.46	51.34	0.9055	5.6733	0.0420	1.7089	0.0250
4	57	715.53	5.42	13.14	7.24	5.42	1.31	5.91	44.61	-2547.97	-1.90	25.67	52.80	0.9257	5.7654	0.0388	1.9016	0.0176
4	58	727.95	5.42	13.14	7.24	3.42	1.31	5.91	38.21	-2842.51	-1.90	28.62	50.52	0.9075	5.7302	0.0393	1.8989	0.0173
4	59	742.70	5.42	13.14	7.24	4.42	1.31	4.91	41.44	-2693.71	-1.90	27.13	51.27	0.9439	5.7742	0.0386	1.9640	0.0093
5	60	755.34	7.42	11.14	7.24	4.42	1.31	5.91	37.76	-2864.93	-1.83	28.85	51.04	0.8934	5.7611	0.0388	1.7343	0.0236
5	61	768.13	5.42	11.14	9.24	4.42	1.31	5.91	37.24	-2889.98	-1.78	29.10	51.15	0.8906	5.7742	0.0386	1.6347	0.0336
5	62	780.77	5.42	11.14	5.24	4.42	1.31	5.91	33.64	-3055.70	-1.78	30.76	48.76	0.8769	5.7742	0.0386	1.6347	0.0336
5	63	793.41	5.42	11.14	7.24	6.42	1.31	5.91	41.67	-2683.26	-1.90	27.02	53.08	0.8948	5.7742	0.0386	1.9299	0.0168
5	64	804.78	5.42	11.14	7.24	4.42	1.31	3.91	33.61	-3054.95	-1.86	30.75	49.17	0.8865	5.7742	0.0386	1.7937	0.0240
5	65	816.00	6.42	11.14	7.24	4.42	1.31	5.91	36.44	-2926.81	-1.78	29.47	50.61	0.8852	5.5473	0.0456	1.7171	0.0231
5	66	827.27	5.42	10.14	7.24	4.42	1.31	5.91	31.53	-3150.44	-1.86	31.71	49.44	0.8760	5.6454	0.0407	1.8466	0.0198
5	67	838.47	5.42	11.14	8.24	4.42	1.31	5.91	36.28	-2934.23	-1.78	29.54	50.64	0.8890	5.7742	0.0386	1.6347	0.0336
5	68	850.56	5.42	11.14	6.24	4.42	1.31	5.91	34.72	-3005.92	-1.78	30.26	49.29	0.8753	5.7742	0.0386	1.6347	0.0336
5	69	863.09	5.42	11.14	7.24	5.42	1.31	5.91	38.92	-2812.39	-1.79	28.32	51.66	0.8817	5.7062	0.0399	1.6683	0.0270
5	70	874.64	5.42	11.14	7.24	3.42	1.31	5.91	31.99	-3129.69	-1.85	31.50	48.55	0.8694	5.6270	0.0420	1.8319	0.0206
5	71	888.92	5.42	11.14	7.24	4.42	2.31	5.91	38.26	-2841.02	-1.86	28.60	51.46	0.8819	5.7561	0.0393	1.8054	0.0259
5	72	903.33	5.42	11.14	7.24	4.42	1.31	4.91	34.55	-3012.35	-1.83	30.33	49.17	0.8759	5.7030	0.0397	1.7661	0.0242
6	73	914.66	7.42	10.14	7.24	4.42	1.31	5.91	34.31	-3026.63	-1.71	30.47	50.12	0.8764	5.6377	0.0406	1.5326	0.0323
6	74	926.45	5.42	10.14	9.24	4.42	1.31	5.91	33.41	-3063.82	-1.86	30.84	50.52	0.8793	5.6454	0.0407	1.8466	0.0198
6	75	939.20	5.42	10.14	5.24	4.42	1.31	5.91	29.83	-3228.60	-1.86	32.49	48.13	0.8676	5.6454	0.0407	1.8466	0.0198
6	76	953.59	5.42	10.14	7.24	6.42	1.31	5.91	38.82	-2815.31	-1.85	28.35	52.05	0.8888	5.6961	0.0398	1.8065	0.0237
6	77	966.36	5.42	10.14	7.24	4.42	3.31	5.91	38.92	-2809.78	-1.90	28.29	52.11	0.8712	5.7742	0.0386	1.8973	0.0159
6	78	980.88	5.42	10.14	7.24	4.42	1.31	3.91	30.55	-3195.87	-1.85	32.17	47.40	0.8560	5.4978	0.0446	1.8676	0.0198
6	79	994.06	6.42	10.14	7.24	4.42	1.31	5.91	33.90	-3042.42	-1.82	30.62	50.25	0.8823	5.7742	0.0386	1.7197	0.0269
6	80	1005.53	5.42	10.14	8.24	4.42	1.31	5.91	32.37	-3111.78	-1.86	31.32	50.03	0.8792	5.6454	0.0407	1.8466	0.0198
6	81	1019.81	5.42	10.14	6.24	4.42	1.31	5.91	30.81	-3183.50	-1.86	32.04	48.78	0.8706	5.6454	0.0407	1.8466	0.0198
6	82	1031.14	5.42	10.14	7.24	5.42	1.31	5.91	35.33	-2975.05	-1.90	29.95	50.83	0.9001	5.7742	0.0386	1.9088	0.0203
6	83	1044.33	5.42	10.14	7.24	3.42	1.31	5.91	29.25	-3254.45	-1.90	32.75	47.95	0.8670	5.7742	0.0386	1.9541	0.0105
6	84	1057.41	5.42	10.14	7.24	4.42	2.31	5.91	36.29	-2931.52	-1.87	29.51	50.82	0.8785	5.7219	0.0394	1.8383	0.0271
6	85	1068.75	5.42	10.14	7.24	4.42	1.31	4.91	31.32	-3162.03	-1.78	31.83	48.39	0.8610	5.7742	0.0386	1.6340	0.0350
7	86	1081.39	7.42	10.14	7.24	3.42	1.31	5.91	31.50	-3151.94	-1.86	31.73	48.60	0.8699	5.7742	0.0386	1.7998	0.0268
7	87	1092.84	5.42	12.14	7.24	3.42	1.31	5.91	35.56	-2964.65	-1.89	29.85	50.30	0.8858	5.7742	0.0386	1.8544	0.0203
7	88	1104.30	5.42	10.14	9.24	3.42	1.31	5.91	31.17	-3166.02	-1.90	31.86	49.05	0.8698	5.7742	0.0386	1.9541	0.0105
7	89	1117.05	5.42	10.14	5.24	3.42	1.31	5.91	27.29	-3344.67	-1.90	33.66	46.77	0.8658	5.7742	0.0386	1.9541	0.0105
7	90	1130.06	5.42	10.14	7.24	3.42	3.31	5.91	35.59	-2964.84	-1.81	29.85	50.76	0.8772	5.6927	0.0398	1.7313	0.0304

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
7	91	1142.92	5.42	10.14	7.24	3.42	1.31	3.91	27.07	-3355.53	-1.88	33.77	46.22	0.8427	5.7742	0.0386	1.8303	0.0195
7	92	1155.78	6.42	10.14	7.24	3.42	1.31	5.91	29.67	-3236.42	-1.84	32.57	49.11	0.8706	5.7742	0.0386	1.7583	0.0260
7	93	1167.91	5.42	10.14	8.24	3.42	1.31	5.91	30.11	-3214.90	-1.90	32.35	48.58	0.8717	5.7742	0.0386	1.9541	0.0105
7	94	1179.02	5.42	10.14	6.24	3.42	1.31	5.91	28.15	-3305.14	-1.90	33.26	47.42	0.8675	5.7742	0.0386	1.9541	0.0105
7	95	1191.16	5.42	10.14	7.24	3.42	2.31	5.91	32.61	-3104.23	-1.72	31.24	49.67	0.8689	5.6869	0.0399	1.5428	0.0330
7	96	1203.81	5.42	10.14	7.24	3.42	1.31	4.91	27.59	-3335.61	-1.70	33.57	47.20	0.8553	5.6223	0.0424	1.5249	0.0383
8	97	1214.94	7.42	10.14	7.24	3.42	1.31	3.91	28.82	-3274.85	-1.87	32.96	48.01	0.8787	5.6659	0.0401	1.8527	0.0210
8	98	1226.28	5.42	12.14	7.24	3.42	1.31	3.91	33.37	-3066.59	-1.83	30.87	48.23	0.8720	5.7509	0.0390	1.7415	0.0310
8	99	1240.78	5.42	10.14	9.24	3.42	1.31	3.91	28.82	-3274.60	-1.88	32.96	47.44	0.8494	5.7742	0.0386	1.8303	0.0195
8	100	1253.64	5.42	10.14	5.24	3.42	1.31	3.91	25.45	-3429.90	-1.88	34.52	44.89	0.8382	5.7742	0.0386	1.8303	0.0195
8	101	1266.72	5.42	10.14	7.24	5.42	1.31	3.91	33.39	-3064.09	-1.90	30.85	48.90	0.8754	5.7548	0.0389	1.9051	0.0158
8	102	1278.39	5.42	10.14	7.24	3.42	3.31	3.91	33.55	-3056.85	-1.89	30.77	49.30	0.8891	5.6537	0.0424	1.8971	0.0172
8	103	1292.81	6.42	10.14	7.24	3.42	1.31	3.91	27.14	-3353.91	-1.80	33.75	46.72	0.8553	5.7742	0.0386	1.6762	0.0287
8	104	1307.20	5.42	11.14	7.24	3.42	1.31	3.91	29.59	-3240.55	-1.82	32.61	47.94	0.8779	5.7742	0.0386	1.7229	0.0289
8	105	1320.19	5.42	10.14	8.24	3.42	1.31	3.91	28.01	-3312.29	-1.88	33.34	46.84	0.8459	5.7742	0.0386	1.8303	0.0195
8	106	1331.63	5.42	10.14	6.24	3.42	1.31	3.91	26.35	-3388.53	-1.88	34.11	45.52	0.8382	5.7742	0.0386	1.8303	0.0195
8	107	1344.95	5.42	10.14	7.24	3.42	2.31	3.91	29.90	-3225.78	-1.86	32.47	48.03	0.8622	5.7742	0.0386	1.7857	0.0214
8	108	1358.06	5.42	10.14	7.24	3.42	1.31	2.91	25.43	-3434.09	-1.74	34.56	45.90	0.8623	5.6191	0.0431	1.6124	0.0304
9	109	1372.58	7.42	10.14	7.24	3.42	1.31	2.91	27.75	-3324.93	-1.83	33.47	46.13	0.8476	5.7742	0.0386	1.7409	0.0312
9	110	1384.02	5.42	12.14	7.24	3.42	1.31	2.91	32.59	-3102.42	-1.84	31.23	47.58	0.8865	5.7196	0.0395	1.7792	0.0270
9	111	1396.78	5.42	10.14	9.24	3.42	1.31	2.91	27.57	-3335.51	-1.74	33.57	46.99	0.8677	5.6191	0.0431	1.6124	0.0304
9	112	1409.86	5.42	10.14	5.24	3.42	1.31	2.91	23.55	-3520.55	-1.74	35.43	44.70	0.8599	5.6191	0.0431	1.6124	0.0304
9	113	1421.19	5.42	10.14	7.24	5.42	1.31	2.91	33.12	-3077.81	-1.84	30.98	48.48	0.8576	5.7742	0.0386	1.7532	0.0257
9	114	1433.94	5.42	10.14	7.24	3.42	3.31	2.91	33.23	-3071.23	-1.90	30.92	48.26	0.8511	5.7742	0.0386	1.9010	0.0179
9	115	1445.33	6.42	10.14	7.24	3.42	1.31	2.91	26.85	-3365.14	-1.89	33.86	46.99	0.8476	5.7742	0.0386	1.8478	0.0239
9	116	1456.45	5.42	11.14	7.24	3.42	1.31	2.91	29.36	-3251.27	-1.83	32.73	47.05	0.8886	5.7382	0.0392	1.7440	0.0294
9	117	1467.91	5.42	10.14	8.24	3.42	1.31	2.91	26.65	-3377.85	-1.74	33.99	46.42	0.8637	5.6191	0.0431	1.6124	0.0304
9	118	1479.67	5.42	10.14	6.24	3.42	1.31	2.91	24.47	-3478.25	-1.74	35.00	45.30	0.8597	5.6191	0.0431	1.6124	0.0304
9	119	1492.42	5.42	10.14	7.24	4.42	1.31	2.91	29.87	-3228.22	-1.81	32.49	47.27	0.8727	5.7109	0.0396	1.7219	0.0252
9	120	1505.41	5.42	10.14	7.24	3.42	2.31	2.91	29.84	-3228.14	-1.88	32.49	47.09	0.8600	5.7173	0.0395	1.8462	0.0212
10	121	1518.42	7.42	10.14	5.24	3.42	1.31	2.91	25.82	-3414.11	-1.83	34.36	44.81	0.8384	5.7742	0.0386	1.7409	0.0312
10	122	1532.94	5.42	12.14	5.24	3.42	1.31	2.91	30.67	-3190.79	-1.84	32.12	46.42	0.8864	5.7196	0.0395	1.7792	0.0270
10	123	1545.73	5.42	10.14	5.24	5.42	1.31	2.91	31.28	-3162.53	-1.84	31.84	47.40	0.8635	5.7742	0.0386	1.7532	0.0257
10	124	1556.86	5.42	10.14	5.24	3.42	3.31	2.91	31.45	-3153.11	-1.90	31.74	47.03	0.8495	5.7742	0.0386	1.9010	0.0179
10	125	1568.33	5.42	10.14	5.24	3.42	1.31	4.91	25.71	-3422.08	-1.70	34.44	45.96	0.8471	5.6223	0.0424	1.5249	0.0383
10	126	1582.83	6.42	10.14	5.24	3.42	1.31	2.91	24.75	-3461.83	-1.89	34.84	45.81	0.8487	5.7742	0.0386	1.8478	0.0239
10	127	1595.59	5.42	11.14	5.24	3.42	1.31	2.91	27.72	-3326.55	-1.83	33.48	45.64	0.8768	5.7382	0.0392	1.7440	0.0294
10	128	1608.28	5.42	10.14	5.24	4.42	1.31	2.91	27.91	-3318.42	-1.81	33.40	46.08	0.8699	5.7109	0.0396	1.7219	0.0252
10	129	1619.52	5.42	10.14	5.24	3.42	2.31	2.91	27.75	-3323.87	-1.88	33.46	45.85	0.8570	5.7173	0.0395	1.8462	0.0212
11	130	1634.17	7.42	10.14	6.24	3.42	1.31	2.91	26.63	-3376.50	-1.83	33.98	45.53	0.8441	5.7742	0.0386	1.7409	0.0312
11	131	1647.48	5.42	12.14	6.24	3.42	1.31	2.91	31.58	-3148.48	-1.84	31.70	47.03	0.8861	5.7196	0.0395	1.7792	0.0270
11	132	1658.75	5.42	10.14	6.24	5.42	1.31	2.91	32.28	-3116.46	-1.84	31.37	47.93	0.8594	5.7742	0.0386	1.7532	0.0257
11	133	1672.94	5.42	10.14	6.24	3.42	3.31	2.91	32.21	-3118.23	-1.90	31.39	47.74	0.8547	5.7742	0.0386	1.9010	0.0179
11	134	1685.59	5.42	10.14	6.24	3.42	1.31	4.91	26.69	-3377.00	-1.70	33.98	46.59	0.8517	5.6223	0.0424	1.5249	0.0383
11	135	1696.83	6.42	10.14	6.24	3.42	1.31	2.91	25.87	-3410.25	-1.89	34.32	46.40	0.8472	5.7742	0.0386	1.8478	0.0239

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
11	136	1709.81	5.42	11.14	6.24	3.42	1.31	2.91	28.70	-3281.48	-1.83	33.03	46.30	0.8814	5.7382	0.0392	1.7440	0.0294
11	137	1721.09	5.42	10.14	6.24	4.42	1.31	2.91	28.83	-3276.13	-1.81	32.98	46.73	0.8736	5.7109	0.0396	1.7219	0.0252
11	138	1732.44	5.42	10.14	6.24	3.42	2.31	2.91	28.90	-3271.37	-1.88	32.93	46.45	0.8563	5.7173	0.0395	1.8462	0.0212
12	139	1746.88	8.42	10.14	6.24	3.42	1.31	2.91	27.40	-3341.09	-1.84	33.62	47.04	0.8632	5.5763	0.0430	1.8269	0.0225
12	140	1758.44	6.42	12.14	6.24	3.42	1.31	2.91	32.18	-3124.47	-1.71	31.46	47.20	0.8885	5.5375	0.0457	1.5681	0.0336
12	141	1769.56	6.42	10.14	8.24	3.42	1.31	2.91	27.73	-3324.67	-1.89	33.46	47.60	0.8509	5.7742	0.0386	1.8478	0.0239
12	142	1782.22	6.42	10.14	6.24	5.42	1.31	2.91	32.87	-3091.64	-1.75	31.12	48.34	0.8790	5.7288	0.0393	1.5953	0.0355
12	143	1796.41	6.42	10.14	6.24	3.42	3.31	2.91	32.38	-3112.18	-1.83	31.33	48.75	0.8780	5.7742	0.0386	1.7424	0.0291
12	144	1809.17	6.42	10.14	6.24	3.42	1.31	4.91	27.77	-3325.48	-1.78	33.47	47.34	0.8576	5.3454	0.0512	1.7685	0.0258
12	145	1823.75	6.42	11.14	6.24	3.42	1.31	2.91	29.55	-3240.95	-1.89	32.62	46.76	0.8811	5.7742	0.0386	1.8535	0.0200
12	146	1835.25	6.42	10.14	6.24	4.42	1.31	2.91	28.66	-3281.76	-1.90	33.03	46.90	0.8859	5.7392	0.0392	1.9625	0.0101
12	147	1848.02	6.42	10.14	6.24	3.42	2.31	2.91	29.16	-3261.32	-1.80	32.83	46.74	0.8758	5.5223	0.0455	1.7564	0.0272
12	148	1861.33	6.42	10.14	6.24	3.42	1.31	3.91	26.42	-3386.94	-1.80	34.09	46.06	0.8536	5.7742	0.0386	1.6762	0.0287
13	149	1875.52	8.42	10.14	5.24	3.42	1.31	2.91	26.52	-3381.52	-1.84	34.03	46.35	0.8598	5.5763	0.0430	1.8269	0.0225
13	150	1886.86	6.42	12.14	5.24	3.42	1.31	2.91	31.28	-3165.85	-1.71	31.87	46.54	0.8866	5.5375	0.0457	1.5681	0.0336
13	151	1899.44	6.42	10.14	5.24	5.42	1.31	2.91	31.97	-3133.03	-1.75	31.54	47.70	0.8744	5.7288	0.0393	1.5953	0.0355
13	152	1912.31	6.42	10.14	5.24	3.42	3.31	2.91	31.62	-3147.10	-1.83	31.68	48.11	0.8780	5.7742	0.0386	1.7424	0.0291
13	153	1926.61	6.42	10.14	5.24	3.42	1.31	4.91	27.14	-3354.78	-1.78	33.76	46.60	0.8511	5.3454	0.0512	1.7685	0.0258
13	154	1941.72	6.42	11.14	5.24	3.42	1.31	2.91	28.73	-3278.63	-1.89	33.00	46.12	0.8803	5.7742	0.0386	1.8535	0.0200
13	155	1954.19	6.42	10.14	5.24	4.42	1.31	2.91	27.50	-3335.22	-1.90	33.57	46.36	0.8840	5.7392	0.0392	1.9625	0.0101
13	156	1966.31	6.42	10.14	5.24	3.42	2.31	2.91	28.13	-3308.28	-1.80	33.30	46.18	0.8758	5.5223	0.0455	1.7564	0.0272
13	157	1977.31	6.42	10.14	5.24	3.42	1.31	3.91	25.40	-3433.89	-1.80	34.56	45.48	0.8539	5.7742	0.0386	1.6762	0.0287
14	158	1988.77	8.42	10.14	5.24	3.42	1.31	3.91	27.74	-3325.52	-1.84	33.47	47.43	0.8786	5.7237	0.0394	1.7730	0.0283
14	159	2001.75	6.42	12.14	5.24	3.42	1.31	3.91	32.43	-3109.78	-1.84	31.31	47.47	0.8832	5.7742	0.0386	1.7555	0.0269
14	160	2013.34	6.42	10.14	5.24	5.42	1.31	3.91	32.21	-3118.64	-1.88	31.39	48.69	0.8694	5.6299	0.0418	1.8820	0.0227
14	161	2024.58	6.42	10.14	5.24	3.42	3.31	3.91	32.30	-3115.00	-1.87	31.35	48.84	0.8561	5.7742	0.0386	1.8192	0.0230
14	162	2037.34	6.42	10.14	5.24	3.42	1.31	5.91	27.81	-3322.03	-1.84	33.43	47.92	0.8702	5.7742	0.0386	1.7583	0.0260
14	163	2051.86	7.42	10.14	5.24	3.42	1.31	3.91	27.17	-3351.08	-1.87	33.73	46.61	0.8674	5.6659	0.0401	1.8527	0.0210
14	164	2063.33	6.42	11.14	5.24	3.42	1.31	3.91	28.83	-3275.19	-1.85	32.97	46.70	0.8695	5.7224	0.0394	1.7911	0.0261
14	165	2076.03	6.42	10.14	5.24	4.42	1.31	3.91	29.23	-3257.00	-1.83	32.78	47.43	0.8738	5.6482	0.0405	1.7761	0.0246
14	166	2087.27	6.42	10.14	5.24	3.42	2.31	3.91	29.75	-3240.24	-1.54	32.61	47.24	0.8688	5.7742	0.0386	1.1537	0.0404
15	167	2101.44	5.42	12.14	5.24	3.42	1.31	3.91	31.79	-3139.21	-1.83	31.60	47.01	0.8704	5.7509	0.0390	1.7415	0.0310
15	168	2115.78	5.42	10.14	5.24	5.42	1.31	3.91	31.75	-3139.46	-1.90	31.60	47.59	0.8655	5.7548	0.0389	1.9051	0.0158
15	169	2130.19	5.42	10.14	5.24	3.42	3.31	3.91	32.03	-3126.68	-1.89	31.48	48.01	0.8840	5.6537	0.0424	1.8971	0.0172
15	170	2141.39	5.42	11.14	5.24	3.42	1.31	3.91	27.92	-3317.75	-1.82	33.39	46.62	0.8695	5.7742	0.0386	1.7229	0.0289
15	171	2152.45	5.42	10.14	5.24	4.42	1.31	3.91	28.81	-3275.86	-1.85	32.98	46.15	0.8498	5.4978	0.0446	1.8676	0.0198
15	172	2166.97	5.42	10.14	5.24	3.42	2.31	3.91	28.42	-3293.70	-1.86	33.15	46.68	0.8550	5.7742	0.0386	1.7857	0.0214
16	173	2178.16	7.42	10.14	5.24	3.42	1.31	4.91	28.19	-3305.98	-1.79	33.27	47.22	0.8658	5.7720	0.0387	1.6514	0.0323
16	174	2189.72	5.42	12.14	5.24	3.42	1.31	4.91	32.33	-3116.34	-1.76	31.37	47.65	0.8825	5.7742	0.0386	1.5934	0.0307
16	175	2202.31	5.42	10.14	5.24	5.42	1.31	4.91	32.81	-3090.96	-1.90	31.11	48.93	0.8711	5.7742	0.0386	1.8939	0.0192
16	176	2213.44	5.42	10.14	5.24	3.42	3.31	4.91	32.73	-3094.69	-1.89	31.15	48.98	0.8777	5.7308	0.0393	1.8790	0.0222
16	177	2224.86	5.42	11.14	5.24	3.42	1.31	4.91	29.24	-3254.97	-1.90	32.76	47.04	0.8697	5.7597	0.0389	1.8770	0.0200
16	178	2237.56	5.42	10.14	5.24	4.42	1.31	4.91	29.56	-3242.96	-1.78	32.64	47.10	0.8543	5.7742	0.0386	1.6340	0.0350
16	179	2252.52	5.42	10.14	5.24	3.42	2.31	4.91	29.04	-3264.38	-1.90	32.85	47.36	0.8556	5.7742	0.0386	1.9237	0.0164
17	180	2265.23	7.42	10.14	6.24	3.42	1.31	4.91	29.01	-3268.28	-1.79	32.89	47.82	0.8643	5.7720	0.0387	1.6514	0.0323

Iteration	calls	CPU Time	LOAD	INSPECT	UNLOAD	TRANSPORT1	TRANSPORT2	TRANSPORT3	Z	G1	G2	TPUT	TIS_AVG	TIS_STD	PU_AVG	PU_STD	CU_AVG	CU_STD
17	181	2276.69	5.42	12.14	6.24	3.42	1.31	4.91	33.21	-3075.86	-1.76	30.97	48.27	0.8828	5.7742	0.0386	1.5934	0.0307
17	182	2289.89	5.42	10.14	8.24	3.42	1.31	4.91	28.47	-3295.13	-1.70	33.16	47.81	0.8610	5.6223	0.0424	1.5249	0.0383
17	183	2301.34	5.42	10.14	6.24	5.42	1.31	4.91	33.56	-3056.06	-1.90	30.76	49.63	0.8768	5.7742	0.0386	1.8939	0.0192
17	184	2315.88	5.42	10.14	6.24	3.42	3.31	4.91	33.55	-3056.99	-1.89	30.77	49.63	0.8820	5.7308	0.0393	1.8790	0.0222
17	185	2328.73	5.42	11.14	6.24	3.42	1.31	4.91	30.26	-3208.02	-1.90	32.29	47.66	0.8713	5.7597	0.0389	1.8770	0.0200
17	186	2341.55	5.42	10.14	6.24	4.42	1.31	4.91	30.52	-3198.77	-1.78	32.20	47.69	0.8529	5.7742	0.0386	1.6340	0.0350
17	187	2353.98	5.42	10.14	6.24	3.42	2.31	4.91	29.98	-3221.14	-1.90	32.42	48.00	0.8565	5.7742	0.0386	1.9237	0.0164
18	188	2365.50	7.42	10.14	6.24	3.42	1.31	3.91	28.07	-3309.71	-1.87	33.31	47.27	0.8702	5.6659	0.0401	1.8527	0.0210
18	189	2378.16	5.42	12.14	6.24	3.42	1.31	3.91	32.63	-3100.58	-1.83	31.22	47.60	0.8688	5.7509	0.0390	1.7415	0.0310
18	190	2389.55	5.42	10.14	6.24	5.42	1.31	3.91	32.59	-3100.88	-1.90	31.22	48.29	0.8724	5.7548	0.0389	1.9051	0.0158
18	191	2404.06	5.42	10.14	6.24	3.42	3.31	3.91	32.87	-3088.06	-1.89	31.09	48.64	0.8852	5.6537	0.0424	1.8971	0.0172
18	192	2416.81	5.42	11.14	6.24	3.42	1.31	3.91	28.85	-3274.53	-1.82	32.96	47.29	0.8742	5.7742	0.0386	1.7229	0.0289
18	193	2428.44	5.42	10.14	6.24	4.42	1.31	3.91	29.75	-3232.62	-1.85	32.54	46.75	0.8511	5.4978	0.0446	1.8676	0.0198
18	194	2442.84	5.42	10.14	6.24	3.42	2.31	3.91	29.18	-3258.83	-1.86	32.80	47.38	0.8594	5.7742	0.0386	1.7857	0.0214
19	195	2454.05	8.42	10.14	6.24	3.42	1.31	3.91	28.70	-3281.37	-1.84	33.02	48.11	0.8830	5.7237	0.0394	1.7730	0.0283
19	196	2465.61	6.42	12.14	6.24	3.42	1.31	3.91	33.23	-3073.06	-1.84	30.94	48.20	0.8885	5.7742	0.0386	1.7555	0.0269
19	197	2480.47	6.42	10.14	8.24	3.42	1.31	3.91	28.18	-3306.01	-1.80	33.27	47.29	0.8558	5.7742	0.0386	1.6762	0.0287
19	198	2492.03	6.42	10.14	6.24	5.42	1.31	3.91	33.24	-3071.67	-1.88	30.92	49.30	0.8707	5.6299	0.0418	1.8820	0.0227
19	199	2504.91	6.42	10.14	6.24	3.42	3.31	3.91	33.16	-3075.45	-1.87	30.96	49.48	0.8585	5.7742	0.0386	1.8192	0.0230
19	200	2516.48	6.42	10.14	6.24	3.42	1.31	5.91	28.79	-3276.91	-1.84	32.98	48.51	0.8700	5.7742	0.0386	1.7583	0.0260
19	201	2530.95	6.42	11.14	6.24	3.42	1.31	3.91	29.87	-3227.30	-1.85	32.48	47.29	0.8706	5.7224	0.0394	1.7911	0.0261
19	202	2544.22	6.42	10.14	6.24	4.42	1.31	3.91	30.11	-3216.52	-1.83	32.37	48.04	0.8728	5.6482	0.0405	1.7761	0.0246
19	203	2558.84	6.42	10.14	6.24	3.42	2.31	3.91	30.87	-3188.62	-1.54	32.10	47.77	0.8648	5.7742	0.0386	1.1537	0.0404
20	204	2570.38	8.42	10.14	7.24	3.42	1.31	3.91	29.48	-3245.54	-1.84	32.66	48.77	0.8875	5.7237	0.0394	1.7730	0.0283
20	205	2584.78	6.42	12.14	7.24	3.42	1.31	3.91	34.00	-3037.24	-1.84	30.58	48.90	0.8946	5.7742	0.0386	1.7555	0.0269
20	206	2596.03	6.42	10.14	9.24	3.42	1.31	3.91	28.86	-3274.81	-1.80	32.96	47.93	0.8607	5.7742	0.0386	1.6762	0.0287
20	207	2610.58	6.42	10.14	7.24	5.42	1.31	3.91	34.18	-3028.43	-1.88	30.48	50.00	0.8759	5.6299	0.0418	1.8820	0.0227
20	208	2623.34	6.42	10.14	7.24	3.42	3.31	3.91	34.14	-3030.31	-1.87	30.50	50.06	0.8617	5.7742	0.0386	1.8192	0.0230
20	209	2636.00	6.42	11.14	7.24	3.42	1.31	3.91	30.85	-3182.20	-1.85	32.03	47.90	0.8741	5.7224	0.0394	1.7911	0.0261
20	210	2650.58	6.42	10.14	7.24	4.42	1.31	3.91	30.99	-3176.03	-1.83	31.97	48.64	0.8743	5.6482	0.0405	1.7761	0.0246
20	211	2662.03	6.42	10.14	7.24	3.42	2.31	3.91	31.49	-3160.20	-1.54	31.81	48.42	0.8665	5.7742	0.0386	1.1537	0.0404
20	212	2676.66	6.42	10.14	7.24	3.42	1.31	4.91	28.71	-3282.25	-1.78	33.03	48.01	0.8637	5.3454	0.0512	1.7685	0.0258
21	213	2689.34	8.42	10.14	7.24	3.42	1.31	2.91	28.30	-3299.70	-1.84	33.21	47.66	0.8636	5.5763	0.0430	1.8269	0.0225
21	214	2702.00	6.42	12.14	7.24	3.42	1.31	2.91	33.18	-3078.42	-1.71	30.99	47.75	0.8890	5.5375	0.0457	1.5681	0.0336
21	215	2712.81	6.42	10.14	9.24	3.42	1.31	2.91	28.47	-3290.71	-1.89	33.11	48.30	0.8594	5.7742	0.0386	1.8478	0.0239
21	216	2724.19	6.42	10.14	7.24	5.42	1.31	2.91	33.71	-3053.00	-1.75	30.73	48.94	0.8798	5.7288	0.0393	1.5953	0.0355
21	217	2735.44	6.42	10.14	7.24	3.42	3.31	2.91	33.32	-3068.90	-1.83	30.89	49.33	0.8773	5.7742	0.0386	1.7424	0.0291
21	218	2750.28	6.42	11.14	7.24	3.42	1.31	2.91	30.41	-3201.42	-1.89	32.23	47.40	0.8842	5.7742	0.0386	1.8535	0.0200
21	219	2763.22	6.42	10.14	7.24	4.42	1.31	2.91	29.74	-3231.98	-1.90	32.53	47.41	0.8833	5.7392	0.0392	1.9625	0.0101
21	220	2775.98	6.42	10.14	7.24	3.42	2.31	2.91	29.83	-3230.13	-1.80	32.51	47.40	0.8817	5.5223	0.0455	1.7564	0.0272

Reliability Domain: TS-SO Run Resulting in the BSF (Run 20)

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
1	1	12.45	8.44	3.08	2.61	10.28	23843.71	415.88	0.9756	32.17	3564	3203	3356	10.16	11.68	10.33	0.3929	0.4528	0.4458	0.9756	0.0226
1	2	25.77	10.44	3.08	2.61	10.28	30002.08	494.07	0.9358	35.49	2955	3203	2793	11.88	11.68	11.93	0.4741	0.4528	0.5381	0.9358	0.0277
1	3	37.08	6.44	3.08	2.61	10.28	18052.76	343.94	0.8875	28.90	4419	3203	4166	8.54	11.68	8.68	0.3255	0.4528	0.3657	0.8875	0.0280
1	4	51.13	8.44	1.08	2.61	10.28	16900.81	328.90	0.9273	29.00	3564	3885	4077	10.16	9.98	8.87	0.3929	0.4513	0.3901	0.9273	0.0227
1	5	64.84	8.44	3.08	4.61	10.28	18493.93	349.16	0.9141	34.28	2956	3203	3284	12.10	11.68	10.49	0.4397	0.4528	0.4410	0.9141	0.0278
1	6	79.34	8.44	3.08	2.61	8.28	23980.83	417.47	0.9863	30.57	3564	3754	3399	10.16	10.24	10.18	0.3929	0.3932	0.4492	0.9863	0.0250
1	7	93.50	9.44	3.08	2.61	10.28	26294.17	448.24	0.8610	33.57	3249	3203	3096	10.98	11.68	10.91	0.4335	0.4528	0.4785	0.8610	0.0256
1	8	107.89	7.44	3.08	2.61	10.28	21046.17	381.21	0.9264	30.67	3939	3203	3709	9.43	11.68	9.57	0.3599	0.4528	0.4130	0.9264	0.0280
1	9	121.94	8.44	4.08	2.61	10.28	27557.88	460.07	1.1864	33.70	3564	2920	3119	10.16	12.72	10.82	0.3929	0.4854	0.4718	1.1864	0.0306
1	10	136.44	8.44	2.08	2.61	10.28	20310.92	371.68	0.9465	30.64	3564	3503	3674	10.16	10.85	9.64	0.3929	0.4580	0.4126	0.9465	0.0252
1	11	150.59	8.44	3.08	3.61	10.28	20226.73	372.28	0.8139	33.30	3237	3203	3331	11.15	11.68	10.48	0.4226	0.4528	0.4419	0.8139	0.0291
1	12	164.88	8.44	3.08	1.61	10.28	32164.10	521.36	0.9348	30.94	3958	3203	3374	9.18	11.68	10.08	0.3787	0.4528	0.4524	0.9348	0.0266
1	13	178.92	8.44	3.08	2.61	11.28	24071.63	416.03	1.1911	33.07	3564	2969	3341	10.16	12.58	10.33	0.3929	0.4992	0.4471	1.1911	0.0306
1	14	193.20	8.44	3.08	2.61	9.28	24253.61	418.63	1.1675	31.52	3564	3463	3355	10.16	11.02	10.34	0.3929	0.4288	0.4598	1.1675	0.0310
2	15	203.55	10.44	1.08	2.61	10.28	22802.46	403.85	0.8875	32.19	2955	3885	3378	11.88	9.98	10.33	0.4741	0.4513	0.4738	0.8875	0.0249
2	16	214.97	6.44	1.08	2.61	10.28	11575.62	262.84	0.8391	25.97	4419	3885	5049	8.54	9.98	7.45	0.3255	0.4513	0.3183	0.8391	0.0251
2	17	226.20	8.44	1.08	4.61	10.28	12594.97	275.35	0.8657	31.17	2956	3885	3982	12.10	9.98	9.09	0.4397	0.4513	0.4002	0.8657	0.0240
2	18	238.95	8.44	1.08	2.61	8.28	17145.79	332.23	0.9089	27.31	3564	4661	4173	10.16	8.51	8.64	0.3929	0.3773	0.3892	0.9089	0.0226
2	19	253.02	9.44	1.08	2.61	10.28	19549.84	363.76	0.8126	30.50	3249	3885	3732	10.98	9.98	9.55	0.4335	0.4513	0.4282	0.8126	0.0250
2	20	267.86	7.44	1.08	2.61	10.28	14315.30	296.91	0.8780	27.61	3939	3885	4515	9.43	9.98	8.21	0.3599	0.4513	0.3519	0.8780	0.0259
2	21	282.14	8.44	1.08	3.61	10.28	13822.25	292.10	0.7655	30.09	3237	3885	4020	11.15	9.98	8.96	0.4226	0.4513	0.3955	0.7655	0.0263
2	22	293.59	8.44	1.08	1.61	10.28	24022.18	419.26	0.8865	27.84	3958	3885	4137	9.18	9.98	8.68	0.3787	0.4513	0.3882	0.8865	0.0241
2	23	307.53	8.44	1.08	2.61	11.28	16926.89	328.82	0.9598	29.75	3564	3601	4006	10.16	10.60	9.00	0.3929	0.4807	0.3927	0.9598	0.0251
2	24	321.92	8.44	1.08	2.61	9.28	16881.61	328.10	0.9716	28.07	3564	4231	4156	10.16	9.23	8.68	0.3929	0.4098	0.3858	0.9716	0.0263
3	25	333.05	4.44	1.08	2.61	10.28	6485.17	199.42	0.7755	22.67	5785	3885	6631	6.79	9.98	5.91	0.2425	0.4513	0.2324	0.7755	0.0260
3	26	347.23	6.44	1.08	4.61	10.28	7909.48	219.26	0.6264	28.17	3596	3885	4776	10.33	9.98	7.86	0.3608	0.4513	0.3176	0.6264	0.0272
3	27	358.58	6.44	1.08	2.61	8.28	11704.05	264.69	0.8208	24.30	4419	4661	5185	8.54	8.51	7.24	0.3255	0.3773	0.3157	0.8208	0.0248
3	28	372.78	5.44	1.08	2.61	10.28	8944.08	229.01	0.8888	24.28	5020	3885	5784	7.68	9.98	6.63	0.2831	0.4513	0.2726	0.8888	0.0256
3	29	387.05	6.44	2.08	2.61	10.28	14609.29	300.86	0.8584	27.42	4419	3503	4620	8.54	10.85	8.03	0.3255	0.4580	0.3339	0.8584	0.0247
3	30	399.91	6.44	1.08	3.61	10.28	9066.33	232.01	0.7727	27.02	3959	3885	4923	9.44	9.98	7.60	0.3361	0.4513	0.3124	0.7727	0.0283
3	31	414.09	6.44	1.08	1.61	10.28	16724.20	327.93	0.8285	24.44	5003	3885	5298	7.47	9.98	7.00	0.3044	0.4513	0.3091	0.8285	0.0267
3	32	428.27	6.44	1.08	2.61	11.28	11607.21	262.82	0.8717	26.70	4419	3601	4969	8.54	10.60	7.55	0.3255	0.4807	0.3198	0.8717	0.0270
3	33	442.23	6.44	1.08	2.61	9.28	11737.16	264.32	0.8835	25.17	4419	4231	5083	8.54	9.23	7.39	0.3255	0.4098	0.3168	0.8835	0.0272
4	34	456.64	2.44	1.08	2.61	10.28	1599.58	140.75	0.5415	19.18	8281	3885	9476	4.92	9.98	4.29	0.1720	0.4513	0.1571	0.5415	0.0296
4	35	467.11	4.44	3.08	2.61	10.28	12485.89	274.51	0.8239	25.36	5785	3203	5483	6.79	11.68	6.90	0.2425	0.4528	0.2737	0.8239	0.0272
4	36	477.98	4.44	1.08	4.61	10.28	3619.54	167.47	0.4422	25.00	4487	3885	5984	8.55	9.98	6.47	0.2977	0.4513	0.2503	0.4422	0.0283
4	37	489.44	4.44	1.08	2.61	8.28	6673.58	202.03	0.7572	21.11	5785	4661	6726	6.79	8.51	5.81	0.2425	0.3773	0.2331	0.7572	0.0249
4	38	502.09	3.44	1.08	2.61	10.28	3914.17	169.52	0.5754	20.93	6854	3885	7811	5.84	9.98	5.12	0.2055	0.4513	0.1934	0.5754	0.0297
4	39	516.28	4.44	2.08	2.61	10.28	9548.67	237.83	0.7948	24.14	5785	3503	5946	6.79	10.85	6.50	0.2425	0.4580	0.2516	0.7948	0.0289
4	40	530.45	4.44	1.08	3.61	10.28	4703.48	177.94	0.6962	23.90	5040	3885	6299	7.72	9.98	6.20	0.2731	0.4513	0.2425	0.6962	0.0283

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
4	41	544.42	4.44	1.08	1.61	10.28	10408.46	249.47	0.7320	21.22	6841	3885	7146	5.76	9.98	5.48	0.2219	0.4513	0.2254	0.7320	0.0273
4	42	558.70	4.44	1.08	2.61	11.28	6541.36	199.72	0.8081	23.38	5785	3601	6534	6.79	10.60	5.99	0.2425	0.4807	0.2357	0.8081	0.0280
4	43	571.56	4.44	1.08	2.61	9.28	6603.15	200.35	0.8199	21.88	5785	4231	6671	6.79	9.23	5.87	0.2425	0.4098	0.2325	0.8199	0.0272
5	44	585.78	2.44	3.08	2.61	10.28	7072.12	209.17	0.5899	21.61	8281	3203	7874	4.92	11.68	5.01	0.1720	0.4528	0.1921	0.5899	0.0308
5	45	599.84	2.44	1.08	4.61	10.28	488.37	126.16	0.5861	21.95	5898	3885	7775	6.79	9.98	5.19	0.2433	0.4513	0.1900	0.5861	0.0309
5	46	614.36	2.44	1.08	2.61	8.28	1602.52	141.02	0.5232	17.62	8281	4661	9697	4.92	8.51	4.19	0.1720	0.3773	0.1541	0.5232	0.0290
5	47	628.56	2.44	2.08	2.61	10.28	4268.09	174.17	0.5608	20.45	8281	3503	8543	4.92	10.85	4.68	0.1720	0.4580	0.1735	0.5608	0.0260
5	48	642.86	2.44	1.08	3.61	10.28	701.56	129.56	0.5303	20.59	6896	3885	8517	5.85	9.98	4.76	0.2091	0.4513	0.1734	0.5303	0.0286
5	49	657.14	2.44	1.08	1.61	10.28	4076.82	170.64	0.6491	17.72	10355	3885	10804	3.96	9.98	3.78	0.1428	0.4513	0.1427	0.6491	0.0274
5	50	670.22	2.44	1.08	2.61	11.28	1682.65	141.38	0.5741	19.87	8281	3601	9334	4.92	10.60	4.35	0.1720	0.4807	0.1575	0.5741	0.0288
5	51	684.30	2.44	1.08	2.61	9.28	1678.60	141.18	0.5859	18.41	8281	4231	9524	4.92	9.23	4.26	0.1720	0.4098	0.1545	0.5859	0.0310
6	52	698.38	2.44	3.08	4.61	10.28	5516.72	188.98	0.6344	24.49	5898	3203	6537	6.79	11.68	6.02	0.2433	0.4528	0.2179	0.6344	0.0318
6	53	712.33	2.44	1.08	6.61	10.28	198.72	126.16	0.2960	24.35	4579	3885	6905	8.56	9.98	5.81	0.3249	0.4513	0.2235	0.2960	0.0276
6	54	722.92	2.44	1.08	4.61	8.28	469.80	126.16	0.5677	20.21	5898	4661	8234	6.79	8.51	4.91	0.2433	0.3773	0.1810	0.5677	0.0296
6	55	733.70	3.44	1.08	4.61	10.28	1992.12	145.52	0.5548	23.53	5087	3885	6782	7.72	9.98	5.84	0.2742	0.4513	0.2184	0.5548	0.0285
6	56	745.14	2.44	2.08	4.61	10.28	2824.77	155.39	0.6053	23.25	5898	3503	7095	6.79	10.85	5.62	0.2433	0.4580	0.1994	0.6053	0.0274
6	57	757.45	2.44	1.08	5.61	10.28	263.45	126.16	0.3609	23.22	5135	3885	7269	7.72	9.98	5.52	0.2908	0.4513	0.2074	0.3609	0.0284
6	58	771.64	2.44	1.08	4.61	11.28	521.06	126.16	0.6186	22.62	5898	3601	7708	6.79	10.60	5.24	0.2433	0.4807	0.1914	0.6186	0.0276
6	59	784.06	2.44	1.08	4.61	9.28	532.62	126.16	0.6304	21.08	5898	4231	7987	6.79	9.23	5.06	0.2433	0.4098	0.1871	0.6304	0.0296
7	60	798.23	4.44	1.08	6.61	10.28	3240.30	161.70	0.5197	27.39	3637	3885	5526	10.40	9.98	7.01	0.3655	0.4513	0.2681	0.5197	0.0284
7	61	812.19	2.44	3.08	6.61	10.28	4790.38	183.47	0.3444	27.00	4579	3203	5780	8.56	11.68	6.76	0.3249	0.4528	0.2478	0.3444	0.0266
7	62	823.20	2.44	1.08	8.61	10.28	256.30	126.16	0.3533	26.45	3748	3885	6396	10.23	9.98	6.25	0.4020	0.4513	0.2421	0.3533	0.0266
7	63	837.20	2.44	1.08	6.61	8.28	180.13	126.16	0.2777	22.49	4579	4661	7424	8.56	8.51	5.42	0.3249	0.3773	0.2072	0.2777	0.0251
7	64	851.58	3.44	1.08	6.61	10.28	1708.65	141.09	0.6221	26.01	4057	3885	6148	9.58	9.98	6.46	0.3507	0.4513	0.2465	0.6221	0.0292
7	65	866.30	2.44	2.08	6.61	10.28	2153.41	150.57	0.3153	25.72	4579	3503	6268	8.56	10.85	6.31	0.3249	0.4580	0.2308	0.3153	0.0250
7	66	880.80	2.44	1.08	7.61	10.28	289.00	126.16	0.3861	25.50	4136	3885	6665	9.49	9.98	6.04	0.3694	0.4513	0.2310	0.3861	0.0271
7	67	894.88	2.44	1.08	6.61	11.28	231.43	126.16	0.3286	25.08	4579	3601	6737	8.56	10.60	5.92	0.3249	0.4807	0.2256	0.3286	0.0262
7	68	909.27	2.44	1.08	6.61	9.28	242.97	126.16	0.3404	23.43	4579	4231	7108	8.56	9.23	5.64	0.3249	0.4098	0.2169	0.3404	0.0265
8	69	921.80	4.44	1.08	6.61	8.28	3237.06	161.90	0.5014	25.48	3637	4661	5923	10.40	8.51	6.57	0.3655	0.3773	0.2569	0.5014	0.0278
8	70	936.00	2.44	3.08	6.61	8.28	4586.58	180.77	0.3551	25.33	4579	3754	6000	8.56	10.24	6.54	0.3249	0.3932	0.2438	0.3551	0.0248
8	71	950.17	2.44	1.08	8.61	8.28	237.70	126.16	0.3350	24.56	3748	4661	6896	10.23	8.51	5.82	0.4020	0.3773	0.2272	0.3350	0.0269
8	72	960.78	2.44	1.08	6.61	6.28	168.17	126.16	0.2661	20.39	4579	5851	8120	8.56	6.85	4.97	0.3249	0.2960	0.1897	0.2661	0.0242
8	73	971.66	3.44	1.08	6.61	8.28	1655.22	140.65	0.6038	24.12	4057	4661	6578	9.58	8.51	6.03	0.3507	0.3773	0.2310	0.6038	0.0282
8	74	983.20	2.44	2.08	6.61	8.28	2144.14	148.83	0.4449	23.90	4579	4168	6590	8.56	9.32	6.02	0.3249	0.3801	0.2206	0.4449	0.0315
8	75	995.63	2.44	1.08	7.61	8.28	270.41	126.16	0.3678	23.64	4136	4661	7129	9.49	8.51	5.65	0.3694	0.3773	0.2188	0.3678	0.0291
8	76	1008.27	2.44	1.08	5.61	8.28	244.87	126.16	0.3426	21.44	5135	4661	7762	7.72	8.51	5.21	0.2908	0.3773	0.1961	0.3426	0.0278
8	77	1020.59	2.44	1.08	6.61	7.28	279.15	126.16	0.3769	21.50	4579	5196	7659	8.56	7.68	5.26	0.3249	0.3395	0.1999	0.3769	0.0269
9	78	1034.00	4.44	1.08	6.61	6.28	3275.74	162.54	0.4898	23.19	3637	5851	6589	10.40	6.85	5.93	0.3655	0.2960	0.2428	0.4898	0.0274
9	79	1048.08	2.44	3.08	6.61	6.28	4884.73	182.57	0.5108	23.59	4579	4481	6231	8.56	8.73	6.30	0.3249	0.3266	0.2328	0.5108	0.0301
9	80	1059.53	2.44	1.08	8.61	6.28	225.72	126.16	0.3234	22.29	3748	5851	7743	10.23	6.85	5.20	0.4020	0.2960	0.2009	0.3234	0.0250

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
9	81	1073.70	2.44	1.08	4.61	6.28	463.29	126.23	0.5561	18.29	5898	5851	8725	6.79	6.85	4.65	0.2433	0.2960	0.1712	0.5561	0.0279
9	82	1087.67	2.44	1.08	6.61	4.28	383.37	126.16	0.4816	18.06	4579	7963	9321	8.56	5.15	4.35	0.3249	0.2104	0.1672	0.4816	0.0329
9	83	1101.63	3.44	1.08	6.61	6.28	1788.42	142.48	0.5921	21.96	4057	5851	7203	9.58	6.85	5.53	0.3507	0.2960	0.2137	0.5921	0.0268
9	84	1112.73	2.44	2.08	6.61	6.28	2103.82	149.50	0.3514	21.97	4579	5073	7070	8.56	7.78	5.62	0.3249	0.3111	0.2010	0.3514	0.0275
9	85	1126.48	2.44	1.08	7.61	6.28	258.43	126.16	0.3562	21.42	4136	5851	7967	9.49	6.85	5.08	0.3694	0.2960	0.1945	0.3562	0.0275
9	86	1140.22	2.44	1.08	5.61	6.28	232.91	126.16	0.3310	19.40	5135	5851	8370	7.72	6.85	4.83	0.2908	0.2960	0.1832	0.3310	0.0251
9	87	1153.98	2.44	1.08	6.61	5.28	292.25	126.16	0.3903	19.30	4579	6729	8533	8.56	6.02	4.72	0.3249	0.2529	0.1794	0.3903	0.0290
10	88	1168.38	4.44	1.08	8.61	6.28	3209.11	162.97	0.3886	25.24	3054	5851	6432	12.30	6.85	6.09	0.4552	0.2960	0.2460	0.3886	0.0268
10	89	1182.33	2.44	3.08	8.61	6.28	4721.95	179.79	0.5680	25.54	3748	4481	5938	10.23	8.73	6.58	0.4020	0.3266	0.2411	0.5680	0.0327
10	90	1196.56	2.44	1.08	8.61	4.28	440.89	126.16	0.5389	19.81	3748	7963	9174	10.23	5.15	4.42	0.4020	0.2104	0.1684	0.5389	0.0307
10	91	1207.13	3.44	1.08	8.61	6.28	1386.45	141.10	0.2995	23.84	3367	5851	6986	11.30	6.85	5.69	0.4337	0.2960	0.2252	0.2995	0.0252
10	92	1218.14	2.44	2.08	8.61	6.28	2080.10	148.48	0.4087	23.95	3748	5073	6679	10.23	7.78	5.93	0.4020	0.3111	0.2197	0.4087	0.0312
10	93	1229.16	2.44	1.08	9.61	6.28	2.54	126.16	0.1000	23.43	3399	5851	7534	11.24	6.85	5.34	0.4593	0.2960	0.2067	0.0981	0.0149
10	94	1241.59	2.44	1.08	8.61	7.28	336.71	126.16	0.4342	23.45	3748	5196	7252	10.23	7.68	5.54	0.4020	0.3395	0.2115	0.4342	0.0293
10	95	1253.91	2.44	1.08	8.61	5.28	349.78	126.16	0.4476	21.14	3748	6729	8233	10.23	6.02	4.89	0.4020	0.2529	0.1879	0.4476	0.0290
11	96	1265.23	4.44	1.08	9.61	6.28	3075.04	162.81	0.2667	25.94	2858	5851	6336	12.95	6.85	6.13	0.4835	0.2960	0.2458	0.2667	0.0238
11	97	1279.19	2.44	3.08	9.61	6.28	4593.58	181.01	0.3428	26.73	3399	4481	5790	11.24	8.73	6.77	0.4593	0.3266	0.2480	0.3428	0.0249
11	98	1293.28	2.44	1.08	9.61	8.28	12.67	126.16	0.1098	25.75	3399	4661	6663	11.24	8.51	6.00	0.4593	0.3773	0.2328	0.1098	0.0172
11	99	1304.19	2.44	1.08	9.61	4.28	215.84	126.16	0.3137	20.89	3399	7963	8995	11.24	5.15	4.50	0.4593	0.2104	0.1718	0.3137	0.0253
11	100	1318.58	3.44	1.08	9.61	6.28	1552.15	141.48	0.4348	24.70	3101	5851	6865	12.08	6.85	5.77	0.4582	0.2960	0.2267	0.4348	0.0265
11	101	1332.55	2.44	2.08	9.61	6.28	1967.01	149.89	0.1835	25.13	3399	5073	6495	11.24	7.78	6.10	0.4593	0.3111	0.2254	0.1835	0.0205
11	102	1346.75	2.44	1.08	9.61	7.28	111.66	126.16	0.2089	24.59	3399	5196	7090	11.24	7.68	5.68	0.4593	0.3395	0.2206	0.2089	0.0220
11	103	1361.03	2.44	1.08	9.61	5.28	124.72	126.16	0.2224	22.19	3399	6729	8186	11.24	6.02	4.93	0.4593	0.2529	0.1909	0.2224	0.0218
12	104	1375.31	4.44	1.08	9.61	8.28	3045.02	162.28	0.2783	28.40	2858	4661	5580	12.95	8.51	6.94	0.4835	0.3773	0.2758	0.2783	0.0240
12	105	1387.75	2.44	3.08	9.61	8.28	4458.22	181.26	0.1872	28.77	3399	3754	5318	11.24	10.24	7.29	0.4593	0.3932	0.2663	0.1872	0.0200
12	106	1402.19	2.44	1.08	9.61	10.28	31.27	126.16	0.1281	27.70	3399	3885	6156	11.24	9.98	6.49	0.4593	0.4513	0.2547	0.1281	0.0181
12	107	1416.06	3.44	1.08	9.61	8.28	1469.39	140.29	0.4464	27.05	3101	4661	6098	12.08	8.51	6.45	0.4582	0.3773	0.2501	0.4464	0.0259
12	108	1430.03	2.44	2.08	9.61	8.28	1918.92	148.10	0.2769	27.13	3399	4168	6009	11.24	9.32	6.58	0.4593	0.3801	0.2404	0.2769	0.0235
12	109	1443.91	2.44	1.08	9.61	9.28	75.51	126.16	0.1725	26.69	3399	4231	6431	11.24	9.23	6.22	0.4593	0.4098	0.2440	0.1725	0.0189
13	110	1455.38	4.44	1.08	9.61	9.28	3075.60	161.88	0.3410	29.42	2858	4231	5327	12.95	9.23	7.24	0.4835	0.4098	0.2817	0.3410	0.0265
13	111	1466.83	2.44	3.08	9.61	9.28	4709.72	182.15	0.3683	29.70	3399	3463	5248	11.24	11.02	7.44	0.4593	0.4288	0.2717	0.3683	0.0258
13	112	1478.30	2.44	1.08	7.61	9.28	333.26	126.16	0.4305	24.64	4136	4231	6810	9.49	9.23	5.92	0.3694	0.4098	0.2288	0.4305	0.0280
13	113	1489.41	2.44	1.08	9.61	11.28	63.98	126.16	0.1607	28.50	3399	3601	5946	11.24	10.60	6.66	0.4593	0.4807	0.2607	0.1607	0.0184
13	114	1501.95	3.44	1.08	9.61	9.28	1545.26	140.45	0.5091	28.08	3101	4231	5811	12.08	9.23	6.77	0.4582	0.4098	0.2636	0.5091	0.0276
13	115	1514.50	2.44	2.08	9.61	9.28	2068.30	150.37	0.2464	28.32	3399	3791	5698	11.24	10.17	6.91	0.4593	0.4256	0.2625	0.2464	0.0220
13	116	1527.92	2.44	1.08	8.61	9.28	300.54	126.16	0.3977	25.49	3748	4231	6624	10.23	9.23	6.03	0.4020	0.4098	0.2326	0.3977	0.0269
14	117	1542.14	4.44	1.08	9.61	11.28	3001.32	161.09	0.3292	31.22	2858	3601	5021	12.95	10.60	7.67	0.4835	0.4807	0.3025	0.3292	0.0280
14	118	1554.80	2.44	3.08	9.61	11.28	5112.15	186.92	0.3920	31.61	3399	2969	4989	11.24	12.58	7.79	0.4593	0.4992	0.2896	0.3920	0.0268
14	119	1569.20	2.44	1.08	7.61	11.28	321.72	126.16	0.4187	26.32	4136	3601	6426	9.49	10.60	6.23	0.3694	0.4807	0.2372	0.4187	0.0273
14	120	1583.41	3.44	1.08	9.61	11.28	1545.40	140.60	0.4973	29.92	3101	3601	5391	12.08	10.60	7.24	0.4582	0.4807	0.2798	0.4973	0.0273

Iteration	calls	CPU Time	FAIL1	FAIL2	REPAIR1	REPAIR2	Z	G1	G2	G3	BD1_NUM	BD2_NUM	SYSBD_N	BD1_AVG	BD2_AVG	SYSBD_A	BD1_STD	BD2_STD	SYSBD_S	DT_AVG	DT_STD
14	121	1597.61	2.44	2.08	9.61	11.28	2170.19	153.24	0.1205	30.31	3399	3236	5386	11.24	11.76	7.31	0.4593	0.5020	0.2765	0.1205	0.0179
14	122	1611.70	2.44	1.08	8.61	11.28	289.01	126.16	0.3859	27.25	3748	3601	6195	10.23	10.60	6.42	0.4020	0.4807	0.2524	0.3859	0.0270
15	123	1624.16	4.44	1.08	9.61	10.28	2962.42	161.01	0.2967	30.35	2858	3885	5187	12.95	9.98	7.42	0.4835	0.4513	0.2915	0.2967	0.0236
15	124	1638.13	2.44	3.08	9.61	10.28	4672.15	184.09	0.1765	30.59	3399	3203	5052	11.24	11.68	7.67	0.4593	0.4528	0.2839	0.1765	0.0203
15	125	1652.27	3.44	1.08	9.61	10.28	1451.41	139.83	0.4647	29.01	3101	3885	5643	12.08	9.98	6.95	0.4582	0.4513	0.2682	0.4647	0.0261
15	126	1662.73	2.44	2.08	9.61	10.28	1986.16	150.58	0.1473	29.15	3399	3503	5561	11.24	10.85	7.07	0.4593	0.4580	0.2652	0.1473	0.0177
16	127	1674.17	4.44	1.08	8.61	10.28	3139.87	161.71	0.4185	29.62	3054	3885	5296	12.30	9.98	7.35	0.4552	0.4513	0.2935	0.4185	0.0264
16	128	1686.30	2.44	3.08	8.61	10.28	4689.44	181.47	0.4017	29.17	3748	3203	5367	10.23	11.68	7.26	0.4020	0.4528	0.2630	0.4017	0.0276
16	129	1697.70	3.44	1.08	8.61	10.28	1356.80	140.34	0.3295	28.14	3367	3885	5747	11.30	9.98	6.87	0.4337	0.4513	0.2702	0.3295	0.0240
16	130	1710.36	2.44	2.08	8.61	10.28	2092.05	149.07	0.3725	27.86	3748	3503	5802	10.23	10.85	6.78	0.4020	0.4580	0.2556	0.3725	0.0288
17	131	1721.91	4.44	1.08	8.61	8.28	3119.16	161.68	0.4002	27.60	3054	4661	5747	12.30	8.51	6.80	0.4552	0.3773	0.2696	0.4002	0.0259
17	132	1735.98	2.44	3.08	8.61	8.28	4529.89	179.33	0.4124	27.46	3748	3754	5567	10.23	10.24	7.00	0.4020	0.3932	0.2541	0.4124	0.0317
17	133	1751.56	3.44	1.08	8.61	8.28	1307.45	139.95	0.3112	26.14	3367	4661	6238	11.30	8.51	6.34	0.4337	0.3773	0.2494	0.3112	0.0251
17	134	1766.08	2.44	2.08	8.61	8.28	2083.21	147.33	0.5021	25.95	3748	4168	6162	10.23	9.32	6.41	0.4020	0.3801	0.2337	0.5021	0.0309
18	135	1777.95	4.44	1.08	7.61	8.28	2988.58	160.06	0.3984	26.36	3355	4661	5830	11.19	8.51	6.66	0.4040	0.3773	0.2660	0.3984	0.0251
18	136	1792.14	2.44	3.08	7.61	8.28	4706.57	181.14	0.4452	26.59	4136	3754	5707	9.49	10.24	6.87	0.3694	0.3932	0.2564	0.4452	0.0260
18	137	1803.81	3.44	1.08	7.61	8.28	1351.04	139.69	0.3761	25.10	3656	4661	6368	10.41	8.51	6.19	0.3900	0.3773	0.2381	0.3761	0.0245
18	138	1818.11	2.44	2.08	7.61	8.28	2201.26	148.41	0.5350	25.06	4136	4168	6344	9.49	9.32	6.26	0.3694	0.3801	0.2324	0.5350	0.0342
18	139	1832.48	2.44	1.08	7.61	7.28	369.41	126.16	0.4670	22.55	4136	5196	7517	9.49	7.68	5.38	0.3694	0.3395	0.2025	0.4670	0.0262
19	140	1846.89	4.44	1.08	5.61	8.28	3542.57	164.45	0.6050	24.47	4034	4661	6103	9.55	8.51	6.42	0.3340	0.3773	0.2523	0.6050	0.0259
19	141	1860.95	2.44	3.08	5.61	8.28	4885.03	183.72	0.4200	24.26	5135	3754	6230	7.72	10.24	6.30	0.2908	0.3932	0.2302	0.4200	0.0266
19	142	1875.17	2.44	1.08	3.61	8.28	717.15	129.99	0.5120	18.97	6896	4661	8779	5.85	8.51	4.61	0.2091	0.3773	0.1699	0.5120	0.0283
19	143	1889.52	3.44	1.08	5.61	8.28	1577.80	141.45	0.4633	22.88	4542	4661	6864	8.58	8.51	5.79	0.3055	0.3773	0.2239	0.4633	0.0291
19	144	1900.20	2.44	2.08	5.61	8.28	2421.42	151.51	0.5098	22.85	5135	4168	6863	7.72	9.32	5.81	0.2908	0.3801	0.2114	0.5098	0.0321
19	145	1911.59	2.44	1.08	5.61	9.28	307.70	126.16	0.4053	22.31	5135	4231	7516	7.72	9.23	5.36	0.2908	0.4098	0.2007	0.4053	0.0271
19	146	1924.03	2.44	1.08	5.61	7.28	343.88	126.16	0.4418	20.44	5135	5196	8011	7.72	7.68	5.04	0.2908	0.3395	0.1874	0.4418	0.0305
30	147	1938.84	4.44	1.08	6.61	11.28	3192.21	160.69	0.5523	28.11	3637	3601	5447	10.40	10.60	7.11	0.3655	0.4807	0.2702	0.5523	0.0275
30	148	1951.61	2.44	3.08	6.61	11.28	5314.37	187.36	0.5599	28.05	4579	2969	5660	8.56	12.58	6.91	0.3249	0.4992	0.2485	0.5599	0.0325
30	149	1964.58	3.44	1.08	6.61	11.28	1660.80	140.08	0.6547	26.71	4057	3601	6058	9.58	10.60	6.54	0.3507	0.4807	0.2487	0.6547	0.0311
30	150	1978.55	2.44	2.08	6.61	11.28	2261.90	152.28	0.2885	26.72	4579	3236	6167	8.56	11.76	6.40	0.3249	0.5020	0.2365	0.2885	0.0237
30	151	1992.88	2.44	1.08	5.61	11.28	296.14	126.16	0.3935	23.91	5135	3601	7175	7.72	10.60	5.59	0.2908	0.4807	0.2103	0.3935	0.0244
31	152	2005.34	4.44	1.08	6.61	9.28	3332.41	162.31	0.5641	26.49	3637	4231	5670	10.40	9.23	6.86	0.3655	0.4098	0.2661	0.5641	0.0271
31	153	2019.75	2.44	3.08	6.61	9.28	4948.31	183.05	0.5363	26.30	4579	3463	5832	8.56	11.02	6.72	0.3249	0.4288	0.2446	0.5363	0.0307
31	154	2034.06	3.44	1.08	6.61	9.28	1750.86	141.07	0.6665	25.09	4057	4231	6299	9.58	9.23	6.29	0.3507	0.4098	0.2406	0.6665	0.0297
31	155	2048.69	2.44	2.08	6.61	9.28	2192.32	149.82	0.4143	24.90	4579	3791	6434	8.56	10.17	6.17	0.3249	0.4256	0.2261	0.4143	0.0286