UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DECOMPOSITION ALGORITHM IN FIXED CHARGE TIME-SPACE NETWORK FLOW PROBLEMS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

SAMINEH DEHGHAN NAYERI
Norman, Oklahoma
2017

DECOMPOSITION ALGORITHM IN FIXED CHARGE TIME-SPACE NETWORK FLOW PROBLEMS

A THESIS APPROVED FOR THE
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

BY

Dr. Charles D. Nicholson, Chair

Dr. Theodore Trafalis

Dr. Ziho Kang

To my beloved husband, and my loving parents

Acknowledgements

I would like to express my sincere gratitude to my adviser Dr. *Charles Nicholson*, for his guidance, patience, and time through my Master's project. Beside my adviser, I would like to thank Dr. *Theodore Trafalis*, and Dr. *Ziho Kang* to serve on my thesis committee.

I would also like to thank all my friends for their support and help. I will always remember the joyful moments we spent together during my course of Master’s especially my best friend, *Knün*, for her support, encouragement, and helpful comments on my thesis draft.

I would like to express my most profound gratitude to my beloved husband, *Hossein*, for his unconditional love, encouragement, and supports during my Master's study. Certainly, without his help finishing this work would have been much harder for me.

Last but not least, I also would like to express my deepest gratitude to my beloved parents and my sister for their unconditional love, encouragement, and prayers throughout my life.

Table of Contents

Acknowledgements iv

List of Tables vii

List of Figures………………………………………………………………………..ix

Abstract……………………………………………………………………...…xii

Chapter 1: Introduction 1

1.1 Time-Space Fixed Charge Network Flow Problems 1

1.2 Objective and Outline 5

Chapter 2: Literature Review 7

2.1 Fixed charge Network Flow Problems 7

2.1.1 Exact Approaches to Fixed Charge Network Flow Problems 7

2.1.2 Heuristic Approaches in Fixed Charge Network Flow Problems 9

Chapter 3: Decomposition Algorithm in Time-Space Fixed Charge Network Flow Problems 11

3.1 Introduction 11

3.2 Notation 12

3.3 Problem Statement 15

3.4 Methodology 17

3.4.1 Exact Method 17

3.4.2 Decomposition method 19

3.4.3 Decomposition Method with Relaxation 25

3.5 Design of Experiment 27

3.6 Results and Analysis 28

3.6.1 Statistical Information ... 28

3.6.2 Statistical Analysis ... 46

3.7 Conclusions ... 58

Chapter 4: Conclusions and Future Works ... 60

References ... 64

Appendix A: Box plot of the objective value vs. node-period (NP) for three different methods ... 72

Appendix B: Box plot of the solution time vs. node-period (NP) for three different methods ... 73

Appendix C: Box plot of difference vs. node-period (NP) for two decomposition methods ... 74

Appendix D: Normality test of obj value for different TPV in decomposition method. 75

Appendix E: Normality test of obj value for different TPV in decomposition method with relaxation ... 78

Appendix F: Normality test of objective values for exact method ... 81

Appendix G: Mean objective value of each TPV in each TS network problem for both decomposition methods ... 82

List of Tables

Table 3.1 Table of notations for single commodity time-space FCNF problem……….14

Table 3.2 Different levels of requirements, variable cost and fixed cost …..…………16

Table 3.3 Summary table of solution time for all problems ………………………...30

Table 3.4 The average solution time (seconds) for all problems based on method and TPV……………………………………………………………………………….39

Table 3.5 Average solution time ratio (exact average solution time/decomposition average solution time) ………………………………………………………………40

Table 3.6 Average objective values in three methods……………………………………………………………………………...41

Table 3.7 Percentage of average difference between objective value of exact methods and decomposition methods……………………………………………..…….42

Table 3.8 Problems with large negative value for percentage of difference between objective values ……………………………………………………………….…....45

Table 3.9 Revised solved problem after increasing maximum time limit to four hours……………………………………………..……………………....….45

Table 3.10 Normality test of objective values at 0.05 confidence level……………......47

Table 3.11 Summary table of Bartlett's test (homogeneity of variances)…………….…48

Table 3.12 Comparison of normality test results before and after removing outliers…49

Table 3.13 Comparison of homogeneity variances tests before and after removing outliers…………………………………………………………………...….49

Table 3.14 Revised table of average solution time (seconds) for different methods after removing outliers……………………………………………………………....53

Table 3.15 Revised table of average percentage of different between objective value exact methods and decomposition methods after removing outliers……..……………………………………………………………………..54

Table 3.16 Paired t-test for the objective values in decomposition method…………...56

Table 3.17 Paired t-test for the objective values in decomposition method with relaxation ………………………………………………………………………...57

List of Figures

Figure 1.1 (a) A distribution problem [11] and (b) a facility location problem [12] 3

Figure 1.2 A time-space network with Three nodes and Three time-periods (3,3) 4

Figure 1.3 (a) A time-space network presenting a bank problem [17] and (b) a flight scheduling problem with a time-space network [18] …………………………………...4

Figure 3.1 Branch & bound search tree with nodes and leaf nodes……………………18

Figure 3.2 A time-space network with four nodes and five time-periods……………...21

Figure 3.3 (a)-(d) Decomposition algorithm process in a TSN with four nodes and five time-periods (4, 5) 23

Figure 3.4 Flowchart of exact, decomposition and decomposition with relaxation methods…………………………………………………………………………..….26

Figure 3.5 (a)-(c) Pie chart of solved and unsolved problems in three methods………31

Figure 3.6 Box plot of the objective value vs. NP for exact method…………………..32

Figure 3.7 Box plot of the objective value vs. TPV in decomposition method ……….33

Figure 3.8 Box plot of the objective value vs. TPV for decomposition method with relaxation …………………………………………………………………………….34

Figure 3.9 Box plot of the solution time vs. NP for exact method……………………..35

Figure 3.10 Box plot of the solution time vs. TPV for decomposition method ……….36

Figure 3.11 Box plot of the solution time vs. TPV for decomposition method with relaxation…………………………………………………………………………….37

Figure 3.12 Percentage of difference between objective value in exact method and both decomposition methods…………………………………………………………..….44

Figure 3.13 Average percentage of difference decomposition method for different problems and TPV configurations……………………………………………….………..51

Figure 3.14 Average percentage of difference decomposition method with relaxation for different problems and TPV configurations……………………………..………..…52

Abstract

A wide range of network flow problems primarily used in transportation is categorized as time-space fixed charge network flow (FCNF) problems. In this family of networks, each node is associated with a specific time and is replicated across all time-periods. The cost structure in these problems consists of variable and fixed costs where continuous and binary variables are required to formulate the problem as a mixed integer linear programming. FCNF problems are classified as NP-hard problems, therefore, adding another component (i.e., time) to this type of problem results in a complex problem which is time-consuming and CPU and memory intensive. Various exact and heuristic methods have been proposed and implemented to solve FCNF problems.

In this work, a decomposition heuristic is proposed that subdivides the problem into various time epochs to create smaller and more manageable subproblems. These subproblems are solved sequentially to find an overall solution for the original problem. To evaluate the capability and efficiency of the decomposition method vs. exact method, a total of 1600 problems is generated and solved using Gurobi MIP solver, which runs parallel branch & bound algorithm.

Statistical analysis indicates that depending on the problem specification, the average solution time in decomposition methods is improved by up to four orders of magnitude. While statistically, there is a significant difference between the mean objective value of exact method and each TPV configuration in both decomposition methods, however, the average difference (0-2.16% in decomposition and 1.55-7.85% in decomposition method with relaxation) may not be a serious concern for many

practical large-scale problems. This shows great promise for decomposition method to significantly reduce the solution time which has been an outstanding issue in complicated large-scale problems.

Chapter 1: Introduction

1.1 Time-Space Fixed Charge Network Flow Problems

A directed or undirected graph which contains a set of nodes *N* connected by a set of arcs *A* is called a network *G* = (*N*, *A*). Each node is classified, as supply, demand, and transshipment where the associated value to each node *i* (R_i) is a positive, negative or zero value respectively. Let x_{ij} be the decision variable associated with the flow on each arc (*i*, *j*) and c_{ij} be the cost per unit of flow for transferring a single commodity from node *i* to node *j*. When a fixed cost f_{ij} associated with the arc (*i*, *j*), is added to the network, the problem is called a fixed charge network flow (FCNF) and a binary variable y_{ij} is required for each arc. The binary variable is set to be one if there is a flow on the associated arc. Otherwise, it set to be 0.

$$y_{ij} = \begin{cases} 1, & if\ there\ is\ a\ positive\ flow\ on\ arc\ (i,j) \\ 0, & Otherwise \end{cases}$$

The objective of the FCNF problem is to minimize the total cost including the variable and fixed cost in a network flow problem such that satisfying all requirements and restrictions. FCNF problems are a subset of minimum cost network flow (MCNF) problems. The following set of equations explains the general formulation of a single commodity FCNF problem:

$$min\ \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \sum_{(i,j)\in A} f_{ij} y_{ij} \qquad \textbf{1.1}$$

Subject to

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = R_i \qquad \forall\ (i)\epsilon\ N \qquad \textbf{1.2}$$

$$0 \le x_{ij} \le M_{ij} y_{ij} \qquad \forall\ (i,j)\epsilon\ A \qquad \textbf{1.3}$$

$$y_{ij}\epsilon\ \{0,1\} \qquad \forall\ (i,j)\epsilon\ A \qquad \textbf{1.4}$$

Equation **1.1** and **1.2** set the objective function composed of a variable cost and a fixed cost and the flow balance constraints respectively. Equation **1.3** sets the bounds for x_{ij} where M_{ij} denotes the capacity on each arc (i, j). In other words, Equation **1.3** guarantees that if $y_{ij} = 0$ then there is no flow between node *i* and node *j*. Equation **1.4** confirms that y_{ij} is a binary variable.

A large group of real-world problems are categorized as FCNF problems such as transportation [1], [2] , [3], facility location [4], [5], [6] , distribution [7] and network design problems [8], [9], [10]. Figures **1.1 (a)** and (**b)** illustrate a distribution problem (with a plant and a set of warehouses and customers) and a facility location problem where the objective is to find the optimal location for a new airport which minimizes the weighted sum of distances to the potential catchment areas respectively.

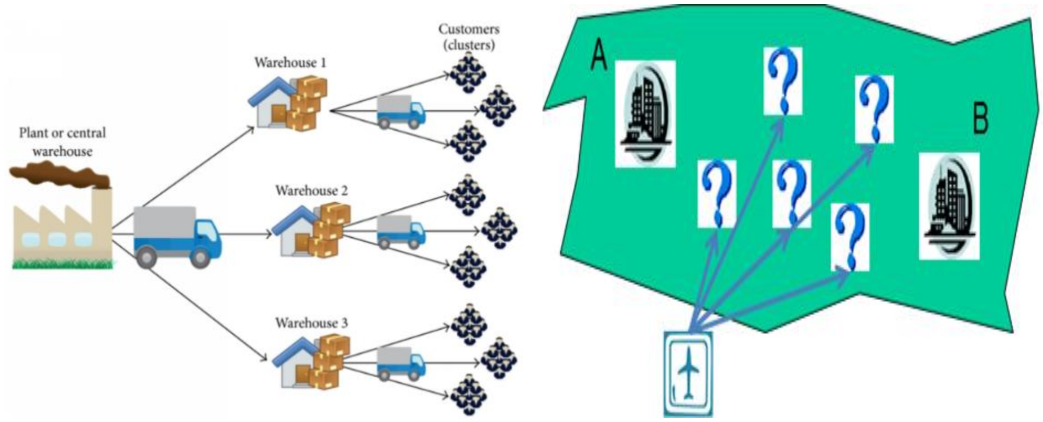


Figure 1.1 (a) A distribution problem [11] **and (b) a facility location problem** [12]

A set of network problems are categorized as time-space network (TSN) problems. In this family of networks, each node i $(i \in N)$ is associated with a certain time-period r ($r \in T$). An additional set of node-time periods are defined such that NTP= $\{(i, r): i \in N \text{ and } r \in T\}$. In this study, the time-space network has some specifications. Firstly, the nodes are replicated in each time-period across the whole network. Secondly, backward arcs are not allowed. This means, each arc only connects node (i, r) to node (j, s) where $r \leq s$. **Figure 1.2** illustrates a simple TSN with three nodes and three time-periods.

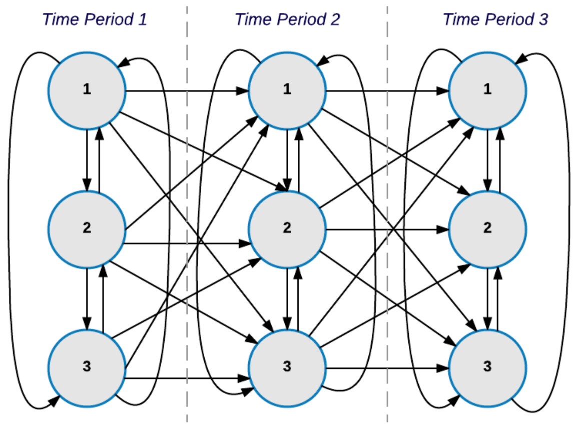


Figure 1.2 A time-space network with three nodes and three time-periods

TSN problems have different applications especially in transportation, scheduling and traffic management problems [13], [14], [15], [16]. Figure **1.3 (a)** and **(b)** illustrate examples of time-space networks. Figure **1.3 (a)** illustrates the application of TSN in the management of cash flow of a large national bank where transferring the flow through TSN is costly. Figure **1.3 (b)** illustrates a flight scheduling problem using TSN. The objective of this problem is to maximize the profit by optimizing the flow of planes while satisfying all the constraints.

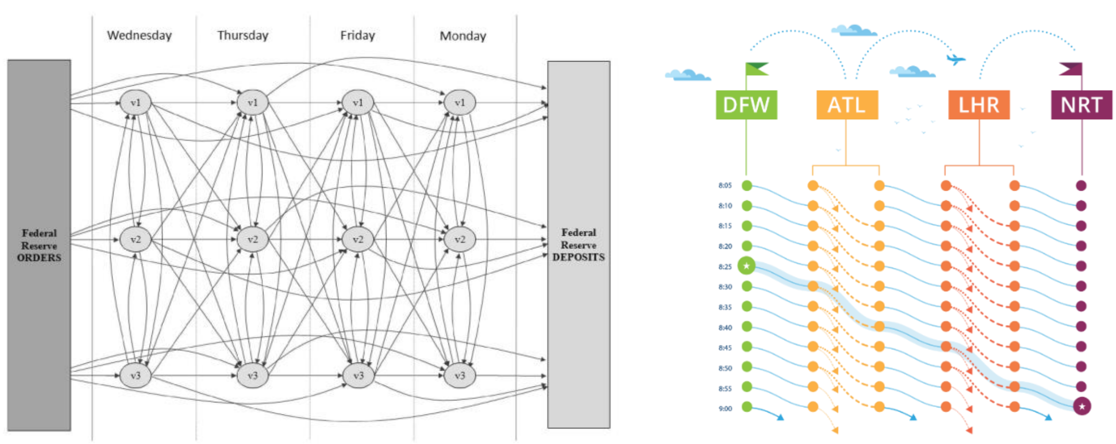


Figure 1.3 (a) A time-space network presenting a bank problem [17] **and (b) a flight scheduling problem with a time-space network** [18]

The fixed charge network flow problems are classified as NP-hard problems [19]. In 1954, Hirsch and Dantzig presented a solution for FCNF problems [20]. Over decades different exact methods were employed to solve FCNF problem to optimality [21], [22], [23], [3]. However, exact methods are time-consuming and computationally expensive (CPU and memory extensive). Therefore, there have been continuous efforts to employ heuristic algorithms to efficiently (e.g., reduced time or CPU and memory demand) solve this type of problems [24], [25], [26].

One of the known heuristic methods to solve FCNF problems is decomposition algorithm. In this algorithm, the problem is broken into a set of sub-problems. These sub-problems are iteratively solved to find an optimal solution for the original problem. The earliest works on decomposition algorithm date back to the seminal work of Dantzig and Wolfe [27] and Benders [28]. Later, this method was further explored and expanded by other researchers to address emerging problems in operation research [29], [30], [31].

1.2 Objective and Outline

The primary goal of this project is to minimize the total cost of transferring a single commodity through a time-space fixed charge network employing the decomposition approach while meeting all requirement and restrictions to find a near optimal solution as fast as possible or within a reasonable amount of time. For this reason, an experiment with some specifications was designed to solve identical problems with exact, decomposition and decomposition with relaxation methods. Then the objective values and solution times of the three methods were analyzed, and the best way that solved the problem the fastest with the lowest gap from the exact method objective value was introduced.

Chapter 2 presents further information about the background of FCNF problems. Employing and examining the decomposition algorithm for single commodity time-space FCNF problems are discussed in **Chapter 3**. Furthermore, **Chapter 4** includes conclusions and future works.

Chapter 2: Literature Review

2.1 Fixed charge Network Flow Problems

The fixed charge network flow problems are classified as NP-hard problems. Over decades different exact methods were employed to solve FCNF problem to optimality. However, exact methods are time-consuming and computationally expensive (CPU and memory extensive). Therefore, there have been continuous efforts to employ heuristic algorithms to efficiently solve this type of problems.

This chapter presents information about the background of FCNF problems, and development of different exact and heuristic approaches.

2.1.1 Exact Approaches to Fixed Charge Network Flow Problems

First time in 1954, Hirsch and Dantzig formulated the FCNF problem [20]. In 1966, Driebeek proposed a method to solve the mixed integer problem in such a way that solved the problem without integrality restrictions and then found the optimum answer which meets the integrality constraints as well [32].

In 1968, Murty presented a solution for FCNF problems by ranking the extreme points [33]. In 1971, a transportation problem was formulated and solved by Gray which was a branch & bound algorithm [21]. He solved the problem by decomposing it into a master integer program and a series of transportation subprograms.

Another exact solution was presented by Steinberg based upon the branch & bound approach which was computationally feasible for large problems [20]. His method added extra features while the amount of required computer storage remained constant for a problem of any given size.

In 1976, Kennington and Unger formulated and solved a transportation problem which included fixed charge as a network using a linear relaxation [34]. Barr et al. in 1981, modified branch & bound algorithm to solve both dense and sparse transportation problems with fixed charge [35].

Later in 1986, Cabot and Erenguc expanded the last two works to improve the branch & bound algorithm by a stronger penalty for branching variable [36].

Palekar et al. developed a conditional penalty for fixed-cost transportation problems which was stronger than other penalties in previous works. They claimed that this method significantly could reduce the branch & bound enumeration and solving time [37]. This work was slightly improved by Lamar and Wallace to ensure there is an optimum solution for the problem.

More improvement occurred in Bell et al. work in 1999 by adding another penalty and employing a new capacity improvement method. It resulted in a noticeable improvement in hard problems solutions [38].

Cruz et al. employed the branch & bound algorithm to solve a large-scale un-capacitated fixed charge network flow problem by using Lagrangian relaxation in place of standard linear relaxation [39].

A branch & cut algorithm was proposed by Ortega and Wolsey to find an optimum solution for un-capacitated fixed charge problems. Their method employs two heuristics (a minimum cost flow and a dynamic slope scaling heuristic) to find feasible solutions as fast as possible. They believed that this method combined with cutting plane algorithm can produce more efficient solutions [40].

2.1.2 Heuristic Approaches in Fixed Charge Network Flow Problems

As mentioned before, the fixed charge network flow problem is classified as an NP-hard problem. Different exact methods which are typically time demanding and memory and CPU extensive were developed to address this problem. Therefore, there is a strong demand to develop more heuristic methods to solve these problems as fast as possible with a reasonable level of accuracy.

Cooper and Drebes, and Denzler each presented heuristic methods to solve the fixed charge problems based on an adjacent extreme point search [24], [41].

Walker proposed a search method as SWIFT algorithm with two phases [42].

Sun and McKeown used tabu search for the fixed charge problem [43]. Also, tabu search has been used by Sun et al. to solve the fixed-charge transportation problem [44]. In both investigations, the researchers believed that to find the best tabu parameters further research is required.

Kim and Pardalos developed a dynamic slope scaling procedure which combined variable and fixed costs as a new coefficient and solves linear programming problem iteratively [45]. A hybrid ant colony optimization algorithm was developed by Monteiro et al. to combine exploration and exploitation [46].

A large group of recent investigations to address fixed charge problems has focused on genetic algorithms. Initially genetic algorithms, which are probabilistic, adaptive algorithms were presented by Holland [47]. These algorithms try to develop a population of candidate solutions, towards a global optimal [48], [49].

Chapter 3: Decomposition Algorithm in Time-Space Fixed Charge Network Flow Problems

3.1 Introduction

As discussed earlier, a set of optimization problems are categorized as time-space networks (TSN). In this family of networks, each node *i* is associated with a certain time-period *r*. TSN problems have wide applications especially in logistics problems such as vehicle-crew scheduling [50], train timetabling [51], and bus scheduling [15]. Generally, TSN problems are complicated large-scale problems. Various exact and heuristic methods have been proposed and implemented to solve TSN problems over the last decades. One of the efficient heuristic algorithms in mathematical programming for solving large-scale time-space network problems is decomposition algorithm.

In this work, a decomposition heuristic is proposed that subdivides the problem into various time epochs to create smaller and more manageable sub-problems. These sub-problems are solved sequentially to find an overall solution for the original problem. The earliest works on decomposition algorithm date back to the seminal work of Dantzig and Wolfe [27] and Benders [28]. Later, this method was further explored and expanded by other researchers to address emerging problems in operation research [29], [30], [31].

Decomposition method was primarily focused on linear programming (LP) problems [52] however, in the last decades this algorithm has been used to solve mixed integer linear programming (MILP) problems [53], [54]. MILP includes problems with integer and continues variables such as transportation and distribution [7], facility location [5], and network design [8] problems as well as other problems.

The focus of this chapter is on single commodity time-space fixed charge network flow problems and is organized as follows: Notations are discussed in Section **3.2**. The problem statement and methodology are described in Sections **3.3** and **3.4**. The rest of this chapter (Sections **3.5-3.7**) details the design of experiment (DOE), data analysis, and concluding remarks.

3.2 Notation

In this work, the time-space network has some specifications as follows:

1- The number of nodes in all time-periods is equal. In other words, nodes are replicated in each time-period across the whole network.

2- Backward arcs are not allowed. This means, no arc is allowed from node (i, r) to node (j, s) when $s < r$ and each arc (i, r, j, s) only connects node (i, r) to node (j, s) in the future time-period $(r < s)$ or in the same time-period $(r = s)$.

In this section, the set of notations used for time-space networks formulation are provided. Let $N = \{1, 2, \ldots, n\}$ and $T = \{1, 2, \ldots, t\}$ present the set of nodes and time-periods, respectively. For a time-space network, another set of node- time period can be defined as *NTP*= $\{(i, r): i \in N \; and \;\; r \in T\}$. Each directed arc in a time-space network, links 2 nodes where both of head and tail nodes are defined with two indexes (*e.g.*, *ir* or *js*). Therefore, the arc set *A* includes four-tuple arcs as (i, r, j, s) where both (i, r) and $(\mathrm{j}, \mathrm{s}) \in NTP$. In this mixed integer problem, we have two decision variables as x_{irjs} and y_{irjs}, which are continuous and binary variables, respectively in a single commodity problem. Here, x_{irjs} presents the decision variable associated with the flow on arc (i,r,j,s) and should be positive or zero $x_{irjs} \geq 0$. y_{irjs} is a binary variable and is set to be one if arc $(i,r,j,s) \in$ A has a positive flow, otherwise it set to be zero.

$$y_{irjs} = \begin{cases} 1, & \text{if there is a positive flow on arc } (i,r,j,s) \\ 0, & \text{Otherwise} \end{cases}$$

We also have two parameters c_{irjs} and f_{irjs} to define the cost per unit flow (variable cost) and the fixed cost on each arc (i,r,j,s). The other parameter in this network is R_{ir} which denotes the requirement at node (i,r) where $(i,r) \in NTP$. Each node will be considered as a supply, demand, or transshipment node if the value of R_{ir} is positive, negative or zero, correspondingly. We have also defined three other sets of nodes as *S*, *D* and *TR* to treat each node as a supply, demand or transshipment node:

$S = \{(i,r): (i,r) \in NTP \text{ and } R_{ir} > 0\}$

$D = \{(i,r): (i,r) \in NTP \text{ and } R_{ir} < 0\}$

$TR = \{(i,r): (i,r) \in NTP \text{ and } R_{ir} = 0\}$

The other parameter is *M*. One of the most common applications of the big *M* coefficient is on mixed integer linear programming problems that allow the binary variable y_{irjs} to activate or deactivate the associated constraint. Because small values for *M* will negate the size of feasible region or could even result in an infeasible model this parameter should be sufficiently large. On the other hand, too large values of *M* will not result in a tight linear relaxation in branch and bound algorithm. Different methods have been proposed to set an appropriate value for *M* in a MILP problem [17], [55]. In this work, *M* is set to be equal to the total supply of the network for all arcs (i,r,j,s) [17]. Table **3.1** provides a detailed summary of notations used in this chapter.

Table 3.1 Table of notations for single commodity time-space FCNF problem

	Notation	Description
Sets	N	Set of nodes in the network $N = \{1, 2, \ldots, n\}$
	T	Set of time periods $T = \{1, 2, \ldots, t\}$
	NTP	Set of nodes and time periods (set of all (i, r) nodes) where $i \in N$ and $r \in T$
	A	Set of directed arcs (i, r, j, s) in the network where (i, r) and $(j, s) \in NTP$ and $(i, r, j, s) \neq (j, s, i, r)$
	S	Set of supply nodes in the network
	D	Set of demand nodes in the network
	TR	Set of transshipment nodes in the network
Parameters	c_{irjs}	The cost per unit flow on arc (i, r, j, s)
	f_{irjs}	Fixed cost on arc (i, r, j, s) in the network
	R_{ir}	Requirement value at node (i, r)
	M	The big M coefficient or upper bound (capacity) for each arc
Variables	x_{irjs}	Decision variable associated with the flow on the arc between nodes (i, r) and (j, s)
	y_{irjs}	Binary variable on each arc (i, r, j, s) in the network $y_{irjs} \epsilon \{0, 1\}$
Network	G	$G = (NTP, A)$ is a directed time-space network

3.3 Problem Statement

As mentioned in Chapter **1**, the fixed charge network flow problems are classified as NP-hard problems. Various factors such as the number of variables and constraints and the structure of the problem can change the problem complexity. Therefore, there have been continuous efforts to employ heuristic algorithms to efficiently solve this type of problems.

The primary objective of this chapter is to employ the proposed decomposition method to minimize the total cost of transferring a single commodity through a time-space fixed charge network while meeting all restrictions and requirements.

As described in Section **3.2**, in fixed charge network flow problems a binary variable y_{irjs}, is defined for each arc. This binary variable is required to write the equation sets and the objective function for the problem in hand. The value of y_{irjs} is set to be one if arc (i, r, j, s) ∈ A is used and there is a positive flow between nodes (i, r) and (j, s) otherwise: it is set to be zero.

The problem formulation for time-space fixed charge network flow problems investigated in this chapter, using the notation provided in Table **3.1**, is provided below:

$$min \sum_{(i,r,j,s)\in A} (c_{irjs} x_{irjs} + f_{irjs} y_{irjs}) \qquad \textbf{3.1}$$

Subject to

$$\sum_{(i,r,j,s)\in A} x_{irjs} - \sum_{(j,s,i,r)\in A} x_{jsir} = R_{ir} \qquad \forall\ (i,r) \epsilon\ NP \qquad \textbf{3.2}$$

$$x_{irjs} \leq M y_{irjs} \qquad \forall\ (i,r,j,s) \epsilon\ A \qquad \textbf{3.3}$$

$$x_{irjs} \geq 0 \qquad \forall\ (i,r,j,s) \epsilon\ A \qquad \textbf{3.4}$$

$$y_{irjs} \epsilon\ \{0,1\} \qquad \forall\ (i,r,j,s) \epsilon\ A \qquad \textbf{3.5}$$

Equations **3.1**, **3.2**, **3.4** and **3.5** set the objective function, the flow balance constraint for each node (i, r), continuous and binary variables, respectively. Equation **3.3** is a forcing constraint to impose a 0 or 1 value to binary variable y_{irjs} (1 for a positive flow x_{irjs} on arc (i, r, j, s); otherwise 0). Since all the arcs are un-capacitated, and there is not any natural capacity in this problem, big M coefficient which is an artificial capacity is required. In other words, Equation **3.3** sets the capacity for all directed arcs in the network. Note that *M* should be chosen wisely and here is set to be equal to the total supply of the network.

Kennington and Nicholson [17] have categorized and examined different types of FCNF problems by varying the range of requirements on each node R_{ir}, variable cost C_{irjs} and fixed cost f_{irjs}. In [17], FCNF problems were classified into 27 classes such as LLL, LLM, LLH where the first, second and third letter shows the range of node requirement, variable cost and fixed cost (L=low, M=medium, H=high). Table **3.2** summarizes the range of values for each of these three parameters at each level.

Table 3.2 Different levels of requirements, variable cost and fixed cost

Level	Requirement for Each Node	Variable Cost	Fixed Cost
Low	10-20	0-10	200-600
Medium	100-200	10-100	2,000-6,000
High	1,000-2,000	100-1,000	20,000-60,000

In [17], it is concluded that LML, LHL, LHM, MML, MHL, MHM, HMM, HHM and HHH problems do not take much CPU time and in contrast, MLM, MLH, MMH, and HLH are most time-consuming problems. Additionally, they have reported that among 27 classes of problems those with a low level of variable cost and a high level of fixed cost are the most time demanding problems. For these reasons, HLH problems, as one of the most challenging types of problems, are explored in this work.

3.4 Methodology

Each of the time-space fixed charge network flow problems investigated in this work was solved by three different techniques by Gurobi MIP solver: exact method, decomposition method and decomposition method with relaxation. An overview of each of these three methods is detailed in this section.

3.4.1 Exact Method

In the exact method, each problem was implemented in python based on the formulation of Equations **3-1** to **3-5** and Gurobi Parallel MIP solver was used to solve the generated time-space FCNF problems. Generally, linear programming-based branch and bound algorithm are used to solve mixed integer linear programming problems. The main idea of the branch and bound method, search tree, nodes and leaf nodes are illustrated in Figure **3.1**.

In this method, we start with the original MIP and ignore all the integrality constraints to generate an LP relaxation and will solve this relaxed problem. While it is rare, a solution may be found in this very initial step which satisfies all the integrality constraints even though those were not explicitly imposed. If this occurs the problem is solved, and the solver will terminate the process. However, in most cases, the branching process is required to be continued. Note that branching is only performed on a variable

(X) with non-integer value (N). In each branching step, one of the non-integer variables is removed and substituted with two new constraints: $X \leq [N]\ or\ X \geq [N] + 1$ where these new problems are called nodes. Then, the original MIP problem is solved again with the newly added constraints. If the optimal solution for each of sub-problems can be computed, then the best one is picked as the optimal solution of the original MIP. This process is repeated to generate the search tree with nodes and leaf nodes (nodes that are not branched yet). The optimal solution for the MIP problem is found when all the leaf nodes are branched or disposed.

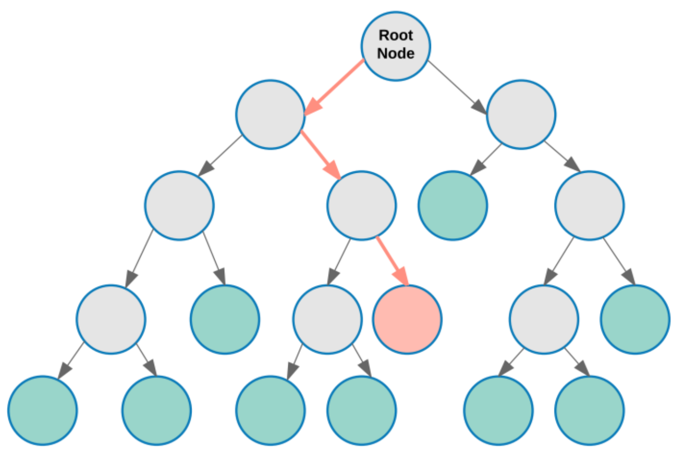


Figure 3.1 Branch & bound search tree with nodes and leaf nodes

The best integer solution found at any point in the search process is called an incumbent. During the search process, if the solver finds an integer feasible solution with a better objective value than the incumbent, the incumbent is updated with this new objective value. The branching process for each node will stop upon reaching to any of the below conditions:

1- The solver finds a feasible solution that satisfies all integrality constraints of the original MIP.

2- The LP relaxation is infeasible, this happens when a feasible integer solution for the original MIP problem does not exist.

3- An optimal solution for LP relaxation is found. However, its objective value is not as good as the incumbent objective value.

Furthermore, in minimization problems, the objective value of the incumbent and the minimum value of all optimal objective values of current leaf nodes are set as the upper and lower bound (best bound) for the optimal solution of the original MIP, respectively. The difference between the current upper and lower bound is known as the gap in Gurobi. Apparently, the original problem is solved to the optimality when this gap is equal to zero [56]. MIP gap in Gurobi is defined as follows:

$$\text{MP Gap} = \frac{|best\ bound - upper\ bound|}{|upper\ bound|} \qquad \textbf{3.6}$$

3.4.2 Decomposition method

The second approach used to address time-space FCNF problems in this work is decomposition algorithm. In this method, the original network is broken into several subproblems based on the consecutive sets of time periods. Then these subproblems will be solved exactly, and the new solutions present the best approximate solution for the original problem. As discussed in Section **3.6**, the decomposition method finds the approximate solution for the original problem much faster than the exact method.

Generally, in NP-hard problems heuristic methods are highly desired. By sacrificing some factors such as optimality, precision, and accuracy to a certain level, heuristic methods can solve most of the time-consuming problems in a shorter time [57]. In this work, the proposed decomposition algorithm can be considered as a greedy heuristic. Despite shortcomings, greedy heuristic algorithms can be desirable in problems where algorithms that seek global optimality are time-demanding and computationally expensive.

To implement the decomposition idea, all possible groups of time-periods should be considered. Note that the minimum number of time-periods in each group is two. A time-period variable, TPV, is defined and based on the number of time-periods in each TSN problem, a range is considered for this variable to decompose the network.

A time-space network with *n* nodes and *t* time-periods is grouped sequentially from two time-periods to half of the time periods ($\frac{t}{2}$) where *t* is an even number and to half of the time-periods +1 ($\frac{t+1}{2}$) where *t* is an odd number, respectively. This step is considered as the base setting of the decomposition approach. When the original problem is decomposed, the decomposition approach will solve the subproblem exactly to find the best answer while the rest of original network is ignored.. Figure **3.2** illustrates a time-space network with four nodes and five time-periods. This TSN problem includes 20 nodes, 124 directed arcs, 124 binary variables, 124 continuous variables and 144 constraints in exact method formulation and additional 40 slack variables in decomposition method formulation.

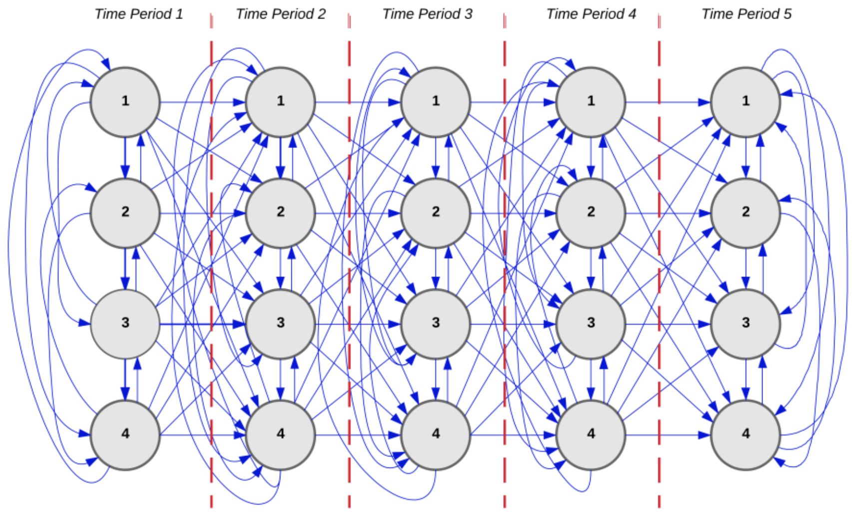
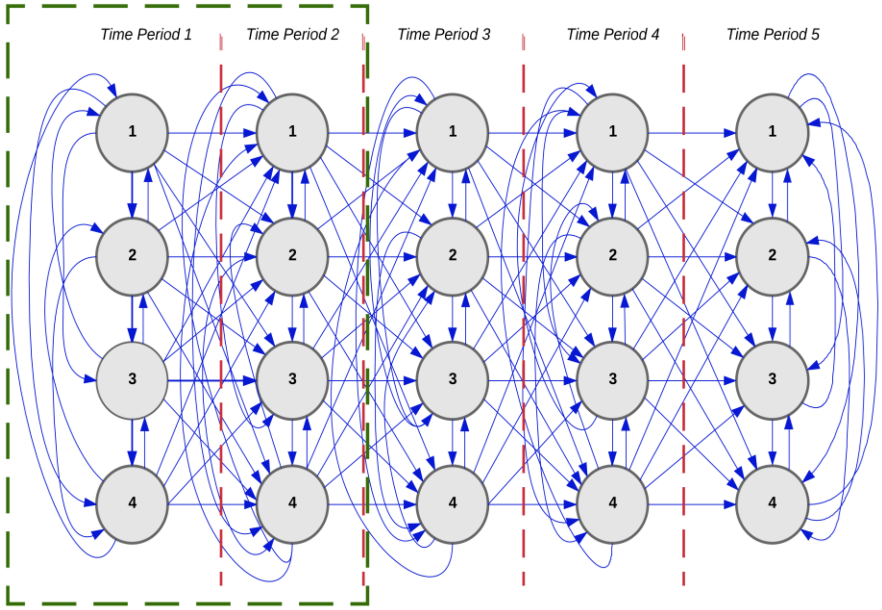
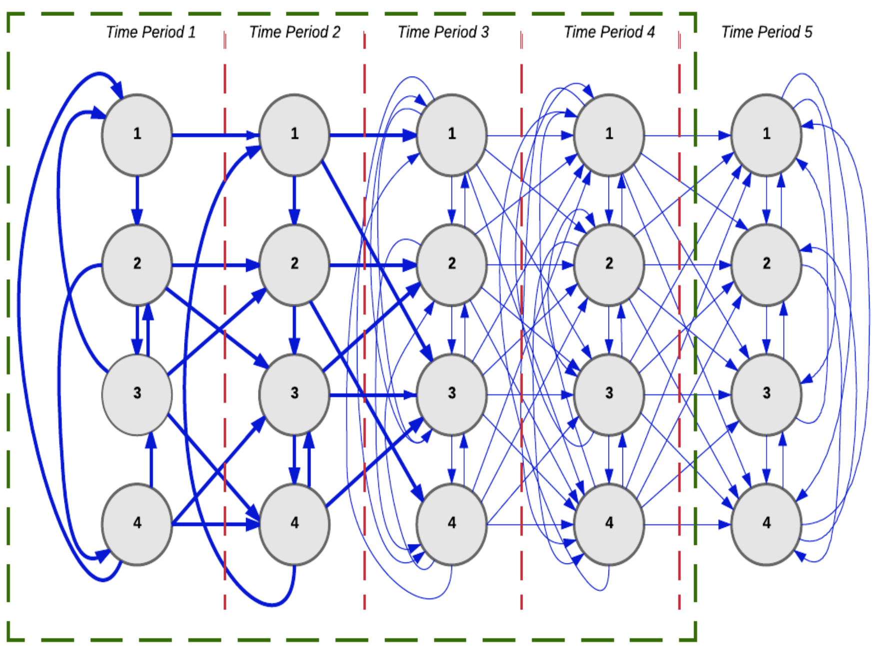


Figure 3.2 A time-space network with four nodes and five time-periods

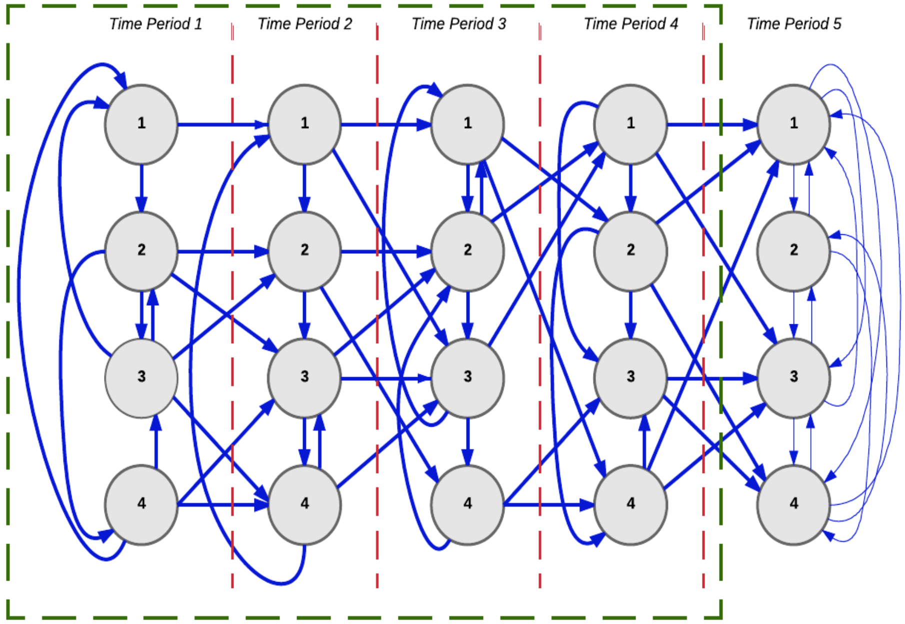
For instance, in a time-space network with four nodes and five time-periods, the decomposition algorithm starts with smallest time period variable (TPV=2), considers the first two time-periods (tp=2) and disregards the rest of network (**Figure 3.3.a**) by zeroing the cost of slack variables for later time-periods. Also, slack variables are required to balance the flow at each node (i,r) in the early time-periods. This subproblem is solved exactly, and the binary variables on arcs with a positive flow are fixed for next step. In next step, the algorithm adds the next two time-periods (tp =4) to make a new subproblem with four consecutive time-periods (**Figure 3.3.b**). This procedure is repeated, and finally the last time-period of the original problem is added **(Figure 3.3.c).** Solving this subproblem proposes a solution to the original MILP problem (**Figure 3.3.d**). The same process is implemented by grouping every three time-periods. The main idea of decomposition algorithm is illustrated in Figures **3.3 (a)**-(**d)**.


Time Period 1
Time Period 2
Time Period 3
Time Period 4
Time Period 5

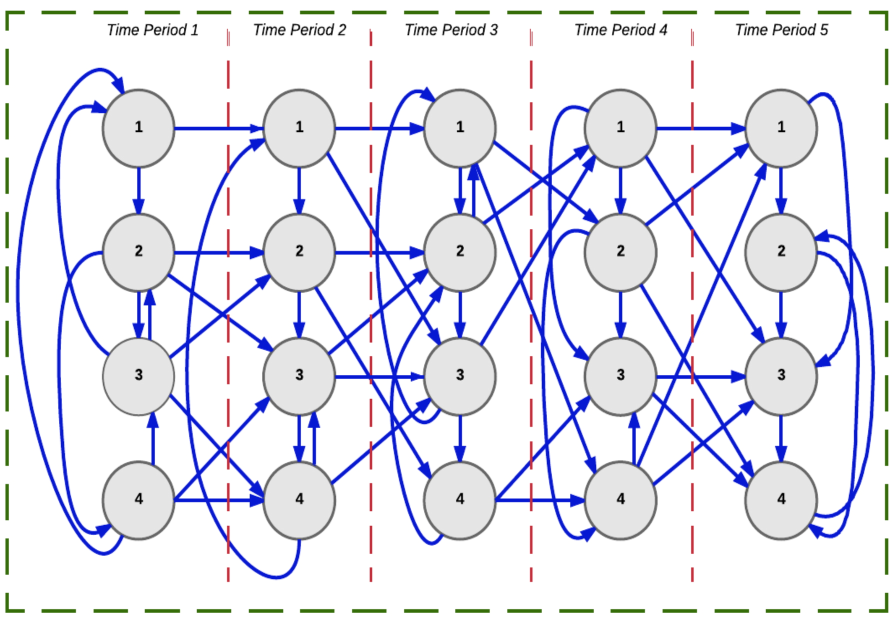
(a)


Time Period 1
Time Period 2
Time Period 3
Time Period 4
Time Period 5

(b)



(c)



(d)

Figure 3.3 (a)-(d) Decomposition algorithm process in a TSN with four nodes and five time-periods (4, 5)

To formulate and implement the decomposition method, further variables and parameters are required namely $S1_{ir}$, $S2_{ir}$, and C_{ir} which are two required slack variables and the coefficient of slack variables at each node (i, r), respectively. The following set of equations explains the mathematical formulation of decomposition method:

$$min \sum_{(i,r,j,s)\in A} (c_{irjs}\, x_{irjs} + f_{irjs} y_{irjs}) + \sum_{(i,r)\in NP} C_{ir}\, (S1_{ir} + S2_{ir}) \quad \textbf{3.7}$$

Subject to

$$\sum_{(i,r,j,s)\in A} x_{irjs} - \sum_{(j,s,i,r)\in A} x_{jsir} + S1_{ir} - S2_{ir} = R_{ir} \quad \forall\ (i,r) \epsilon\ NP \quad \textbf{3.8}$$

$$x_{irjs} \leq M y_{irjs} \quad \forall\ (i,r,j,s) \epsilon\ A \quad \textbf{3.9}$$

$$x_{irjs} \geq 0 \quad \forall\ (i,r,j,s) \epsilon\ A \quad \textbf{3.10}$$

$$y_{irjs} \epsilon\ \{0,1\} \quad \forall\ (i,r,j,s) \epsilon\ A \quad \textbf{3.11}$$

$$S1_{ir}, S2_{ir} \geq 0 \quad \forall\ (i,r) \epsilon\ NP \quad \textbf{3.12}$$

Equations **3.7** and **3.8** set the objective function and the flow balance constraint for each node (i, r) in decomposition method respectively. As you can see, the only difference between the general formulation of FCNF problem and our proposed formulation here is slack variables. The second part of Equation **3.7** includes a critical piece which allows us to solve the early time-periods and ignore the rest of network by zeroing the cost of slack variables for those late time-periods. However, this slack cost for early time-periods is set to be a high value. Equation **3.8** includes two slack variables as $S1_{ir}$ and $S2_{ir}$ which are required to balance the flow at each node (i, r). Equation **3.9** is a forcing constraint to impose a 0 or 1 value to binary variable y_{irjs} (1 for a positive flow x_{irjs} on arc (i, r, j, s); otherwise 0). Also, equations **3.10**, **3.11** and **3.12** set continuous, binary, and slack variables respectively.

3.4.3 Decomposition Method with Relaxation

The third method studied in this work is decomposition method with relaxation. The basic idea of this approach is practically the same as decomposition method with some changes. As discussed before, in decomposition method after breaking the TSN into two parts, when the algorithm begins to solve the early time-periods, it discards the rest of network by zeroing the cost of slack variables for later node-periods. Therefore, to keep the flow balanced at each node (i,r) in early time-periods, slack variables are required.

In contrast, decomposition method with relaxation considers the rest of network with some changes. It keeps all the binary variables within the early time-periods and only changes binary variables in late time-periods to continuous ones and solves the problem. For this reason, there is no need for slack variables in decomposition method with relaxation. The main motivation for proposing this method is to employ another approach which can solve the problem faster and more accurate than the decomposition method. Figure **3.4** illustrates the flowchart of three employed methods in this work.

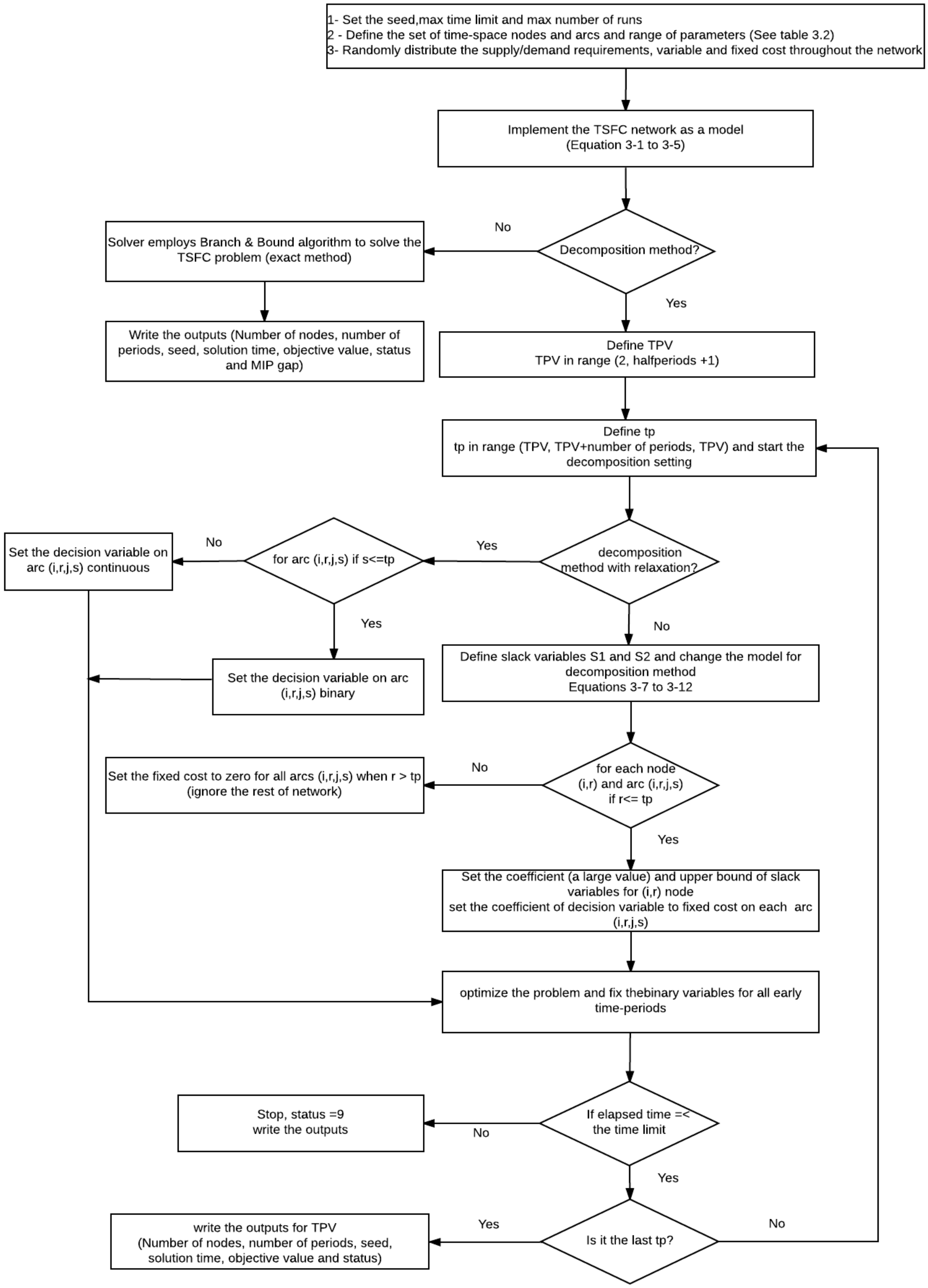


Figure 3.4 Flowchart of exact, decomposition and decomposition with relaxation methods

3.5 Design of Experiment

The primary goal of this experiment is to generate and solve different time-space networks using three methods (Exact, decomposition, and decomposition with relaxation) to evaluate the capability and efficiency of the decomposition methods vs. exact method. For this reason, a total of 1600 problems is solved including 20 runs for each problem (as well as all possible TPVs) within three hours maximum time-limit for each run. The time-space networks considered in this experiment are as follows: (4, 5), (4, 6), (4, 8), (4, 10), (4, 12), (5, 5), (5, 6), (5,8), (5,10), (6,5), (6,6), (6,8)

In this project, the required parameters namely variable cost and fixed cost on each arc (i, r, j, s) and the requirement at each node (i, r) are generated randomly from a uniform probability distribution. To compare the effectiveness of the decomposition method vs. exact method, each problem is solved using identical parameters (same seed for different methods in each run). As discussed in Section **3.3**, the focus of this study is on HLH problems where HLH corresponds to a high (1000-2000), low (0-10) and high (20,000-60,000) range level for the requirement at each node (i, r), variable cost and fixed costs on each arc (i, r, j, s), respectively. Also, the coefficient of the slack variables is set to be 75,000.

In each problem, at first step, all possible arcs are generated and a random cost within the defined range is assigned to each arc. Then supply and demand requirements are distributed randomly throughout the network. This experiment is designed in such a way that 40%, 45% and 15% of time-space nodes are demand, supply, and transshipment nodes, respectively. Also, the problem is considered as a feasible problem if the total demand in each time-period does not exceed the total supply in the same period. The maximum time limit for each problem is set to be three hours, however, later in this chapter for further analysis the maximum time limit is extended to four hours as well.

The numerical computations in this study are performed using Gurobi solver 7.5.1 and Python 3.6.2 on a computer with an Intel core i7- 7700, 3.600 GHz CPU, 16 GB RAM memory, Running 64-bit Windows 10 operating system. The default value for MIP gap in the Gurobi solver is defined as 1e-4. Also, the statistical analysis was performed using R.

3.6 Results and Analysis

3.6.1 Statistical Information

The main purpose of this project is to employ decomposition algorithm to solve the time-space FCNF problems to optimality faster than the exact method while the gap between the objective values is the smallest. For this reason, a total of 1600 problems was solved in about 447 hours to provide enough outputs for statistical analysis and evaluations. Table **3.3** illustrates the summary of solution times for three methods. Note that Gurobi defines some specific values for status that indicates the problem optimality.

The status value for total 119 problems out of 1600 problems is 9. This indicates that the optimization process was terminated while an optimal answer was not found. 95 problems out of 119 unsolved problems occurred in the exact method. Moreover, the majority of 95 unsolved problems occurred in (6, 8), (5, 10), (5, 8) and (6, 6) networks with 20, 19, 16 and 16 unsolved problems respectively. In networks (6,8), (5,10) and (6, 6), total 24 problems out of 1360 problem solved by decomposition methods, left unsolved and the majority of these cases happened in the most complicated network, (6, 8), and in largest TPV (TPV:4). Also, the most complicated and time demanding networks in exact method were (6, 8), (5, 10), (6, 6) and (5, 8) with approximately 60, 57, 51 and 49 hours solution time (for 20 runs each 3 hours maximum time-limit), respectively.

Table 3.3 Summary table of solution time for all problems (hrs)

			Decomposition Method				Decomposition Method with Relaxation				Exact Method		
No	**Node-** Time period	TPV	No. of Problems	Total Problems	Total Unsolved	Total time (Hrs)	No. of Problems	Total Problems	Total Unsolved	Total Time	Total Problems	Total Unsolved	Total Time
1	(4,5)	2	20	40	0	0.0005	20	40	0	0.0002	20	0	0.017412
		3	20		0	0.0013	20		0	0.0005			
2	(4,6)	2	20	40	0	0.0009	20	40	0	0.0005	20	0	0.527215
		3	20		0	0.0026	20		0	0.0021			
3	(5,5)	2	20	40	0	0.0041	20	40	0	0.0006	20	1	5.603729
		3	20		0	0.0467	20		0	0.0054			
4	(5,6)	2	20	40	0	0.0058	20	40	0	0.0016	20	3	17.05932
		3	20		0	0.0801	20		0	0.3187			
5	(6,5)	2	20	40	0	0.1385	20	40	0	0.0024	20	7	27.541715
		3	20		0	1.2094	20		0	0.0387			
6	(6,6)	2	20	40	0	0.04068	20	40	0	0.0116	20	16	51.51966
		3	20		1	5.8561	20		1	3.6059			
7	(4,8)	2	20	60	0	0.001	20	60	0	0.0006	20	0	2.451199
		3	20		0	0.0032	20		0	0.0010			
		4	20		0	0.0117	20		0	0.0091			
8	(5,8)	2	20	60	0	0.0065	20	60	0	0.0024	20	16	48.64328
		3	20		0	0.2312	20		0	0.0109			
		4	20		0	0.9709	20		0	0.4615			
9	(6,8)	2	20	60	0	0.1263	20	60	0	0.0160	20	20	59.68456
		3	20		1	6.7588	20		0	0.2561			
		4	20		8	34.11	20		6	21.9003			
10	(4,10)	2	20	80	0	0.0014	20	80	0	0.0008	20	3	22.34277
		3	20		0	0.0054	20		0	0.0013			
		4	20		0	0.0195	20		0	0.0059			
		5	20		0	0.065	20		0	0.0463			
11	(5,10)	2	20	80	0	0.0095	20	80	0	0.0024	20	19	56.99841
		3	20		0	1.7717	20		0	0.0122			
		4	20		1	4.8795	20		0	0.1297			
		5	20		3	15.807	20		3	12.6908			
12	(4,12)	2	20	100	0	0.0023	20	100	0	0.0013	20	10	38.54187
		3	20		0	0.0088	20		0	0.0033			
		4	20		0	0.0665	20		0	0.0132			
		5	20		0	0.0796	20		0	0.0156			
		6	20		0	2.695	20		0	1.6405			

Figures **3.5 (a)**, **(b)** and **(c)** illustrate the pie chart of solved and unsolved problems in exact, decomposition method and decomposition method with relaxation, respectively.

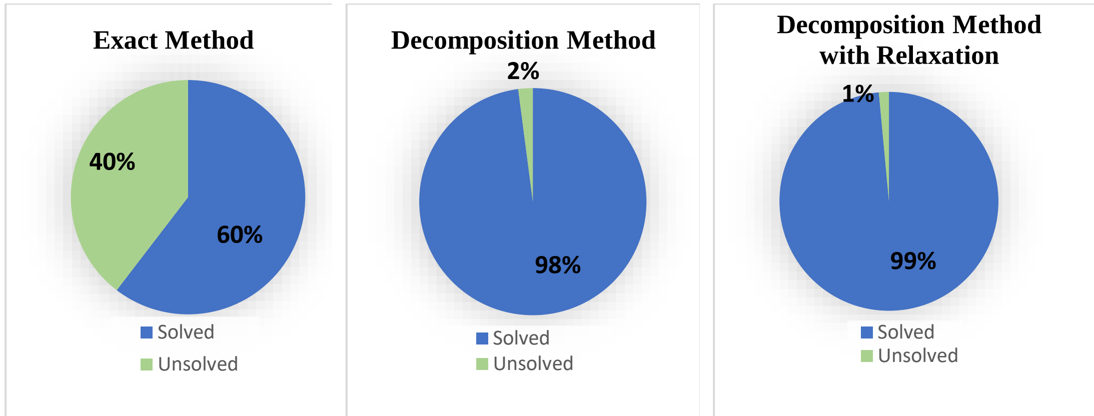


Figure 3.5 (a), (b) and (c) Pie chart of solved and unsolved problems in three methods

Figures **3.6** - **3.8** illustrate the box plot of the objective value vs. node-period and vs. TPV in exact method and both decomposition methods, correspondingly. As shown in Figures **3.7** and **3.8**, larger TPVs compared to smaller ones show a smaller median objective value. When TPV increases, the decomposition approach solves a larger group of time-periods which is more similar to the original time-space network. Therefore, the chance of finding an answer closer to the optimal answer in exact method (a smaller objective value) is higher.

Figures **3.9** - **3.11** illustrate the box plot of the solution time (log scale) vs. node-period and vs. TPV in exact and both decomposition methods, respectively. Clearly, when TPV increases, the median solution time increases monotonically.

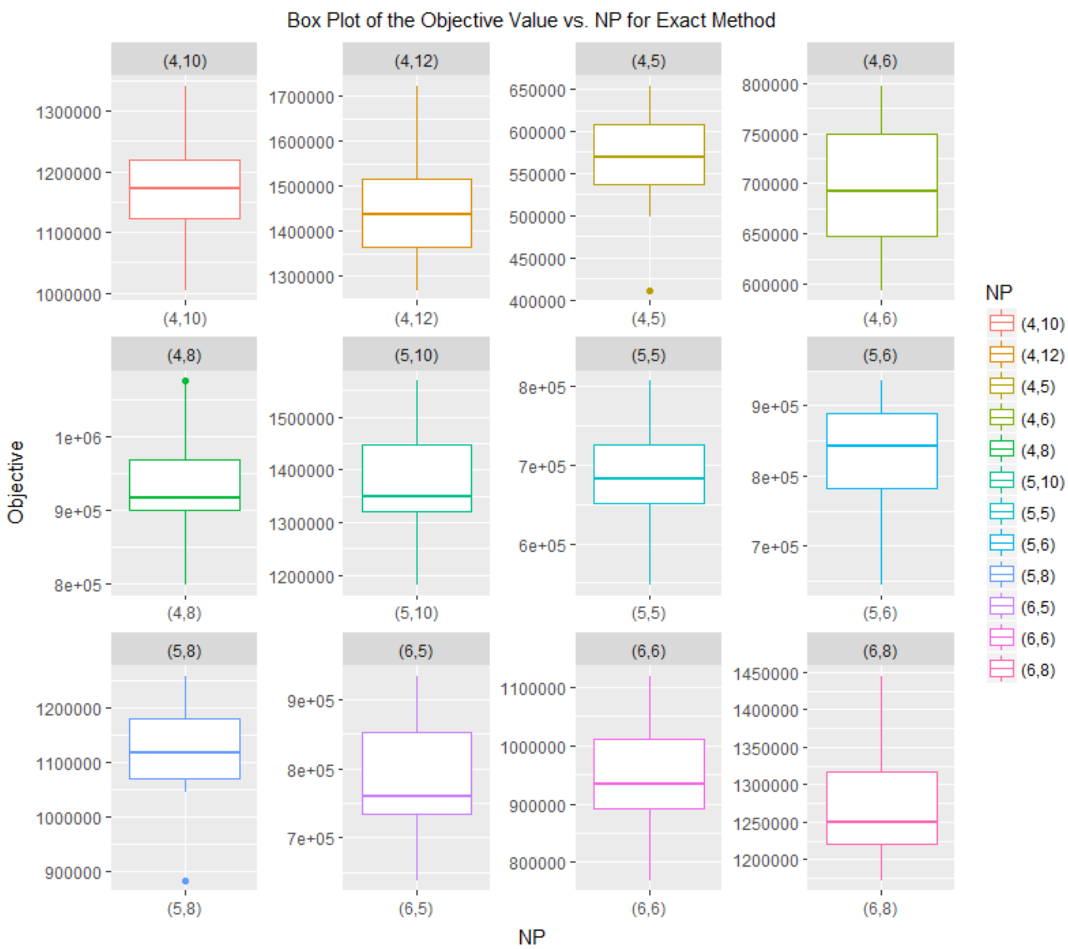


Figure 3.6 Box plot of the objective value vs. NP for exact method

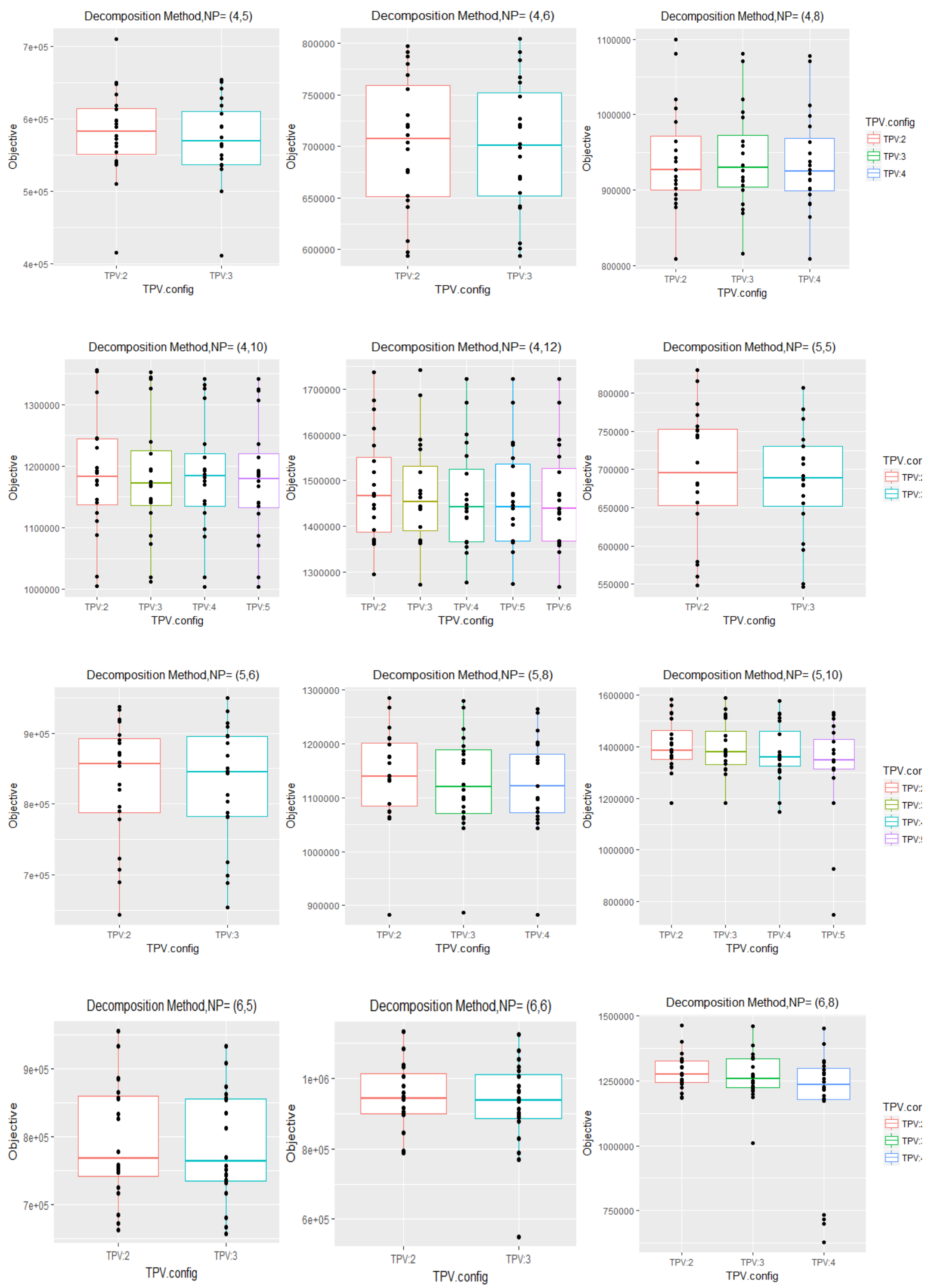


Figure 3.7 Box plot of the objective value vs. TPV in decomposition method

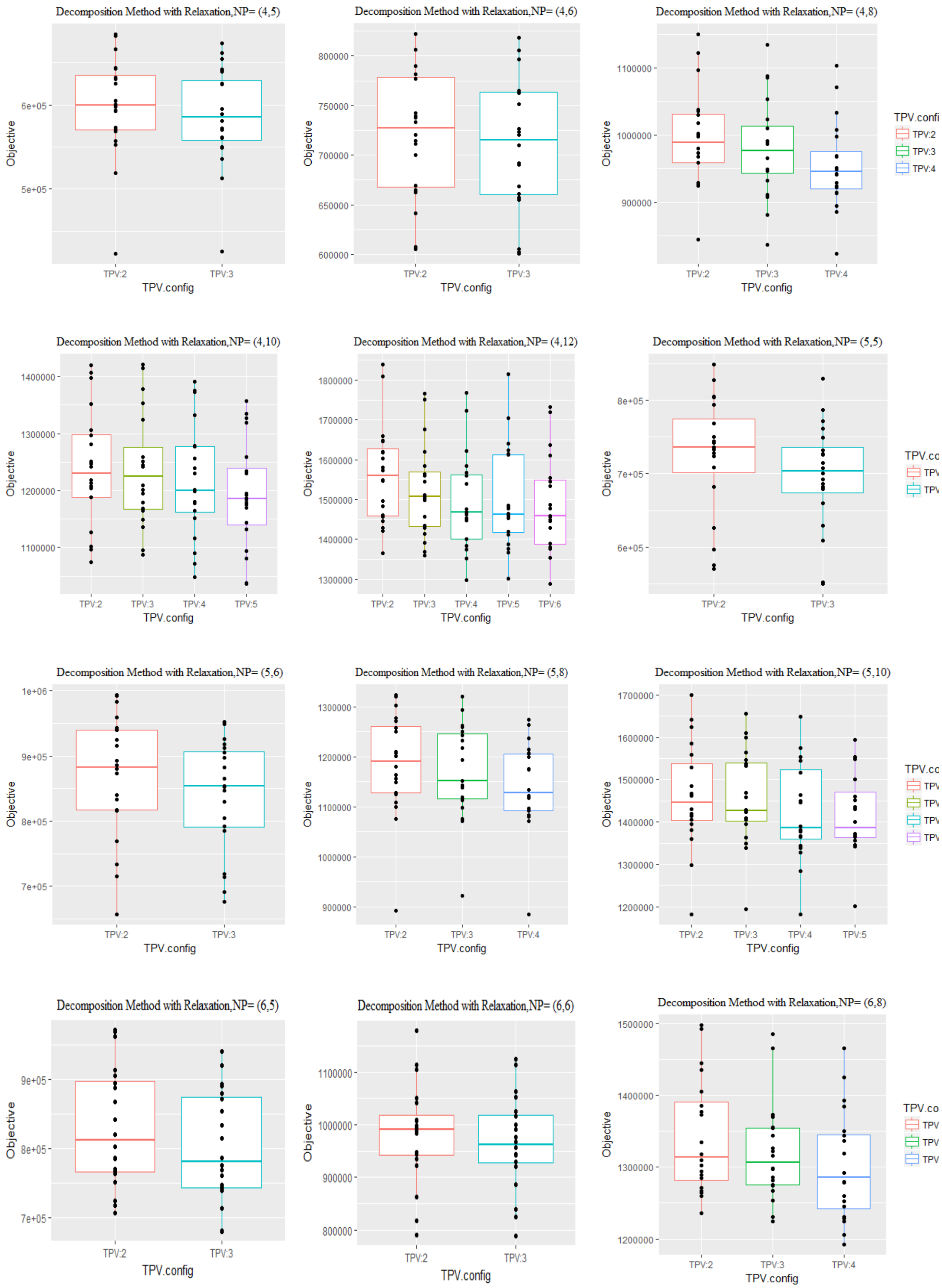


Figure 3.8 Box plot of the objective value vs. TPV for decomposition method with relaxation

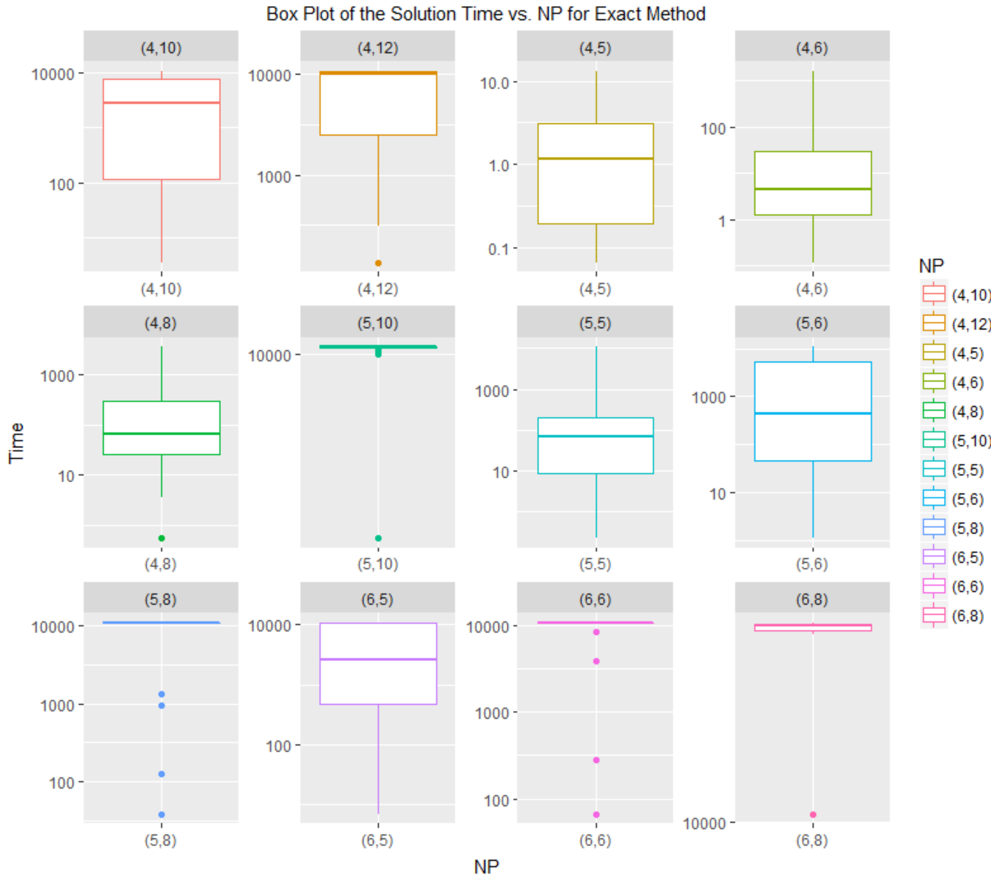


Figure 3.9 Box plot of the solution time vs. NP for exact method

As expected, the median solution time in decomposition method with relaxation is smaller than other methods. In Figure **3.9**, network (5, 10), (5, 8), (6, 6), and (6, 8) illustrate a line in place of a box and whisker for the median solution time. This indicates that most of the problems in these networks (in the exact method) exceeded the three hours maximum time-limit.

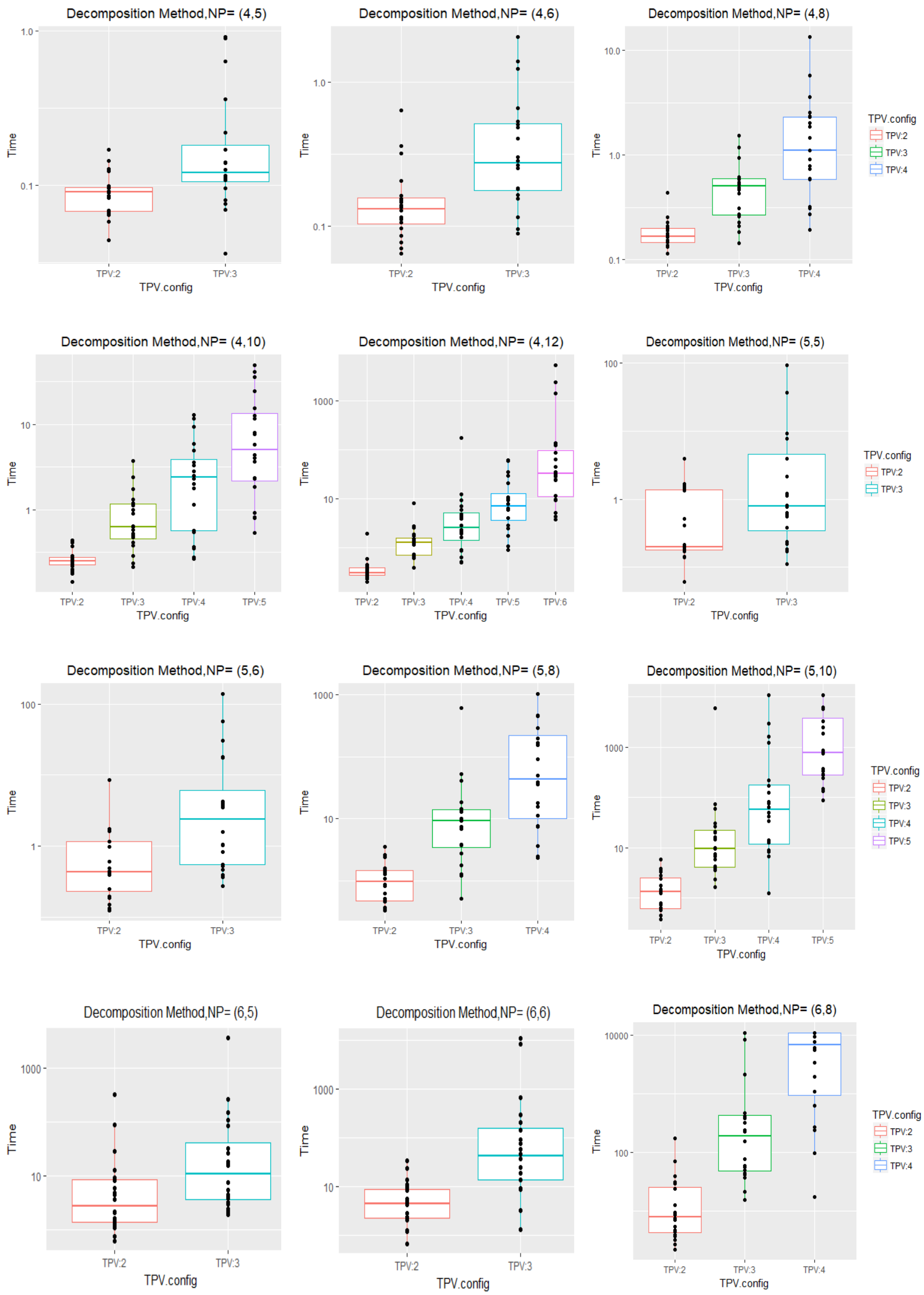


Figure 3.10 Box Plot of the solution time vs. TPV for decomposition method

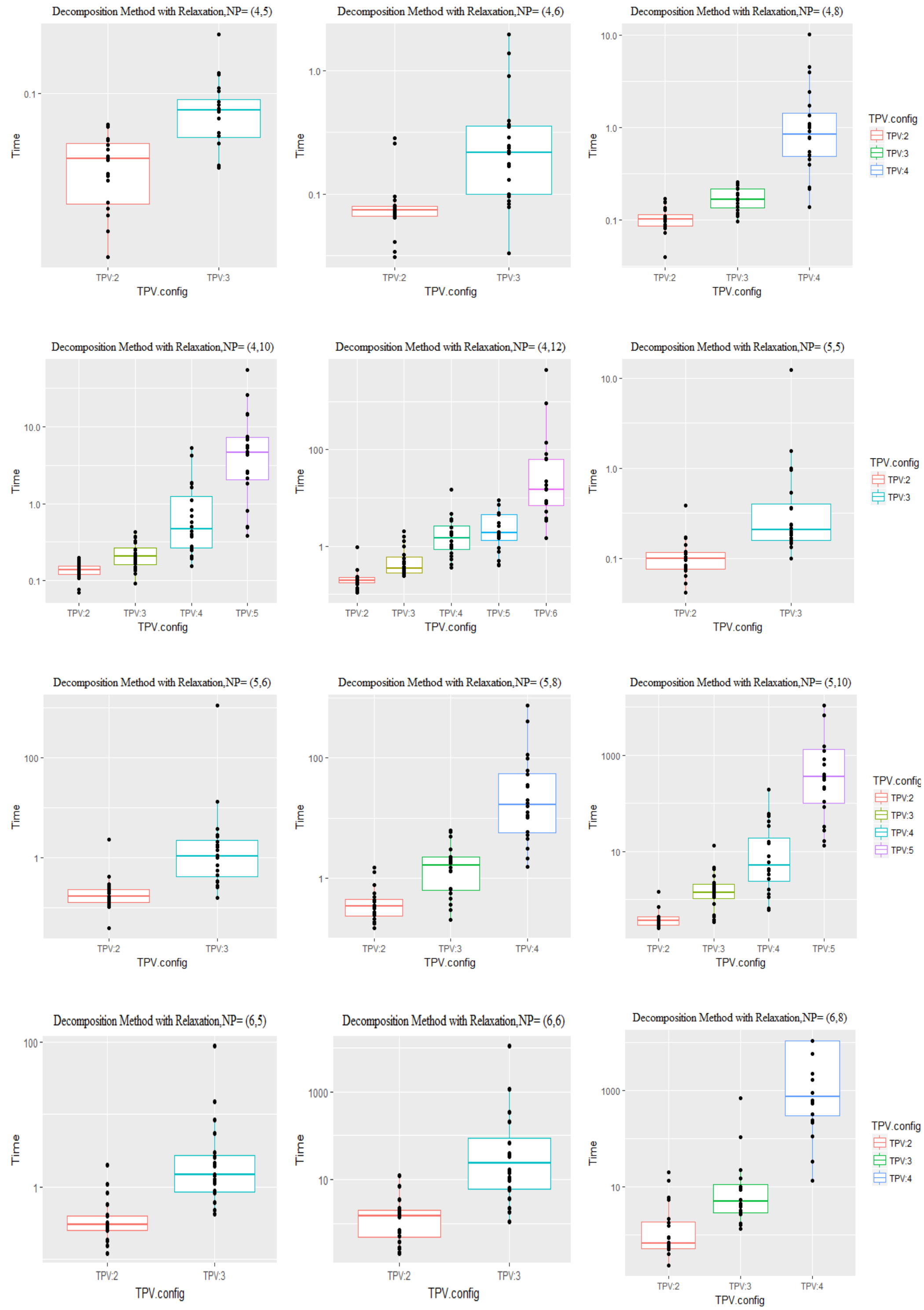


Figure 3.11 Box plot of the solution time vs. TPV for decomposition method with relaxation

A breakdown of average solution time for each of the three methods and for each TPV is listed in Table **3.4**. While TPV increases, the average solution time increases as well. Also, Table **3.4** indicates a huge difference between the average solution time in exact method and decomposition methods. Although, in all problems this difference is evident, however in larger networks this is more noticeable. For example, in the network (4,12) the average solution time in exact method compared to the average solution time in decomposition method with the largest TPV (TPV:6) is increased by 2 orders of magnitude.

Table **3.5** illustrates the ratio of average solution time in the exact method to average solution time in decomposition method and decomposition method with relaxation, respectively. As shown in Table **3.5**, 1.7, 17343.8 and 2.7, 31534.2 are the lowest and highest ratio values in decomposition and decomposition with relaxation methods, correspondingly. This means that, depending on the problem size (i.e., number of nodes and periods) and TPV configuration used, the average solution time could be reduced by up to 4 orders of magnitude (compared to average solution time in Exact method) in decomposition and decomposition with relaxation methods.

Tables **3.6** and **3.7** summarize the average objective value in three methods and percentage of the average difference between the objective value of exact method and decomposition methods, respectively. It is shown in Table **3.6** that, in all problems, the average objective value in decomposition method with relaxation is higher than the decomposition method.

Table 3.4 The average solution time (seconds) for all problems based on method and TPV

Node-Time	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Exact Method		**3.13**	**94.89**	**1008.7**	**3070.7**	**4957.5**	**9193.2**	**441.21**	**8756**	**10743.2**	**10259.7**	**4021.7**	**6937.53**
Decomposition method	2	0.09	0.16	0.69	1.04	24.92	7.32	0.18	1.17	22.72	1.71	0.26	0.4
	3	0.23	0.47	8.4	14.41	217.7	1054.1	0.57	41.6	1216.59	318.9	0.97	1.57
	4							2.1	174.8	6139.85	878.31	3.5	11.96
	5										2845.31	11.69	14.33
	6												485.1
Decomposition method with Relaxation	2	0.042	0.08	0.11	0.28	0.44	2.09	0.1055	0.43	2.87	0.43	0.13	0.22
	3	0.085	0.37	0.96	57.35	6.97	649.06	0.1726	1.96	46.09	2.19	0.22	0.58
	4							1.6391	83.07	3942.05	23.34	1.06	2.37
	5										2284.34	8.32	2.8
	6												295.28

Table 3.5 Average solution time ratio (exact average solution time/decomposition average solution time)

Node-Time Period	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Decomposition method	2	34.8	593.1	1,461.80	2,952.60	198.94	1,255.91	2,451.20	7,483.60	472.9	5,999.80	15,468.00	17,343.80
	3	13.6	201.9	120.1	213.1	22.77	8.72	774.1	210.5	8.8	32.2	4,146.10	4,418.80
	4							210.1	50.1	1.7	11.7	1,149.10	580.1
	5										3.6	344	484.1
	6												14.3
Decomposition method with Relaxation	2	74.5	1,186.10	9,169.70	10,966.70	11,267.07	4,398.68	4,412.10	20,362.30	3,743.30	23,859.80	30,936.10	31,534.20
	3	37	256.5	1,050.70	53.5	711.26	14.16	2,595.40	4,467.20	233.1	4,684.80	18,280.40	11,961.30
	4							270.7	105.4	2.7	439.6	3,794.00	2,927.20
	5										4.5	483.4	2,477.70
	6												23.5

Table 3.6 Average objective value in three methods

Node-Time Period	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Exact Method		5.6872 E+05	6.9644 E+05	6.7769 E+05	8.2208 E+05	7.8006 E+05	9.3780 E+05	9.3434 E+05	1.1211 E+06	1.2688 E+06	1.3812 E+06	1.1754 E+06	1.4564 E+06
Decomposition method	2	5.8117 E+05	7.0287 E+05	6.9658 E+05	8.3059 E+05	7.9526 E+05	9.4906 E+05	9.4216 E+05	1.1398 E+06	1.2868 E+06	1.4036 E+06	1.1920 E+06	1.4837 E+06
	3	5.7043 E+05	6.9971 E+05	6.8416 E+05	8.2596 E+05	7.8741 E+05	9.2583 E+05	9.4131 E+05	1.1293 E+06	1.2673 E+06	1.3963 E+06	1.1832 E+06	1.4733 E+06
	4							9.3693 E+05	1.1261 E+06	1.1568 E+06	1.3751 E+06	1.1831 E+06	1.4623 E+06
	5										1.3259 E+06	1.1780 E+06	1.4626 E+06
	6												1.4623 E+06
Decomposition method with Relaxation	2	5.9807 E+05	7.2176 E+05	7.2498 E+05	8.6801 E+05	8.3055 E+05	9.8177 E+05	9.9607 E+05	1.1849 E+06	1.3424 E+06	1.4645 E+06	1.2409 E+06	1.5622 E+06
	3	5.8748 E+05	7.1102 E+05	6.9730 E+05	8.3862 E+05	8.0178 E+05	9.6606 E+05	9.7850 E+05	1.1687 E+06	1.3202 E+06	1.4559 E+06	1.2362 E+06	1.5224 E+06
	4							9.5417 E+05	1.1422 E+06	1.3015 E+06	1.4255 E+06	1.2168 E+06	1.4956 E+06
	5										1.4182 E+06	1.1933 E+06	1.5004 E+06
	6												1.4851 E+06

Table 3.7 Average Difference (%) Between Objective Value of Exact Method and Decomposition Methods

Node-Time Period	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Decomposition method	2	**2.16**	0.92	0.00	1.03	1.93	1.25	0.83	1.67	1.43	1.62	1.39	1.88
	3	0.29	0.47	0.00	0.47	0.96	-1.26	0.76	0.71	-0.06	1.08	0.64	1.18
	4							0.29	0.44	**-8.75**	-0.42	0.67	0.42
	5										-3.83	0.22	0.44
	6												0.41
Decomposition method with Relaxation	2	5.19	3.68	6.97	5.53	6.55	4.73	6.65	5.64	5.80	5.96	5.58	**7.29**
	3	3.32	2.06	2.95	2.03	2.87	3.09	4.74	4.24	4.09	5.39	5.22	4.57
	4							2.18	1.90	2.57	3.15	3.53	2.70
	5										2.70	**1.55**	2.99
	6												1.97

As shown in table **3.7** the lowest and highest percentage of average difference for objective values in decomposition method and decomposition method with relaxation is -8.75 and 2.16 and 1.55 and 7.29, respectively. Additionally, the percentage of average in problems (6,6), (6,8) and (5,10) is a negative value which indicates that the objective value found by decomposition method is smaller than the exact method ones. The percentage of average difference shown in Table **3.7** is defined as:

$$Difference = \frac{(Obj\ value\ in\ decomposition\ method - obj\ value\ in\ exact\ method)}{obj\ value\ in\ exact\ method} * 100 \qquad \mathbf{3.13}$$

Figure **3.12** visually indicates that the percentage of the average difference between the objective value in exact and decomposition with relaxation methods in all problems is larger than the ones between the exact and decomposition methods. Additionally, few negative outliers are seen in problems (6,8), (5,10) and (6, 6) solved by decomposition method which indicates that the decomposition method found smaller objective values compared to the exact method in the same problem.

There are two possible scenarios that could cause these negative outliers:

1. In networks (6,8), (5,10) and (6, 6) majority of the problems left unsolved in the exact method. In other words, due to the maximum time-limit, the exact method did not solve the problem to optimality and the objective values were not the smallest.
2. The percentage of the average difference for total 61 problems in both decomposition methods, shows a negative value. Also, the optimality status value for 9 problems (Table **3.8**) out of 61 problems is 9 which indicates that the

decomposed network has not been solved to optimality within the maximum time-limit.

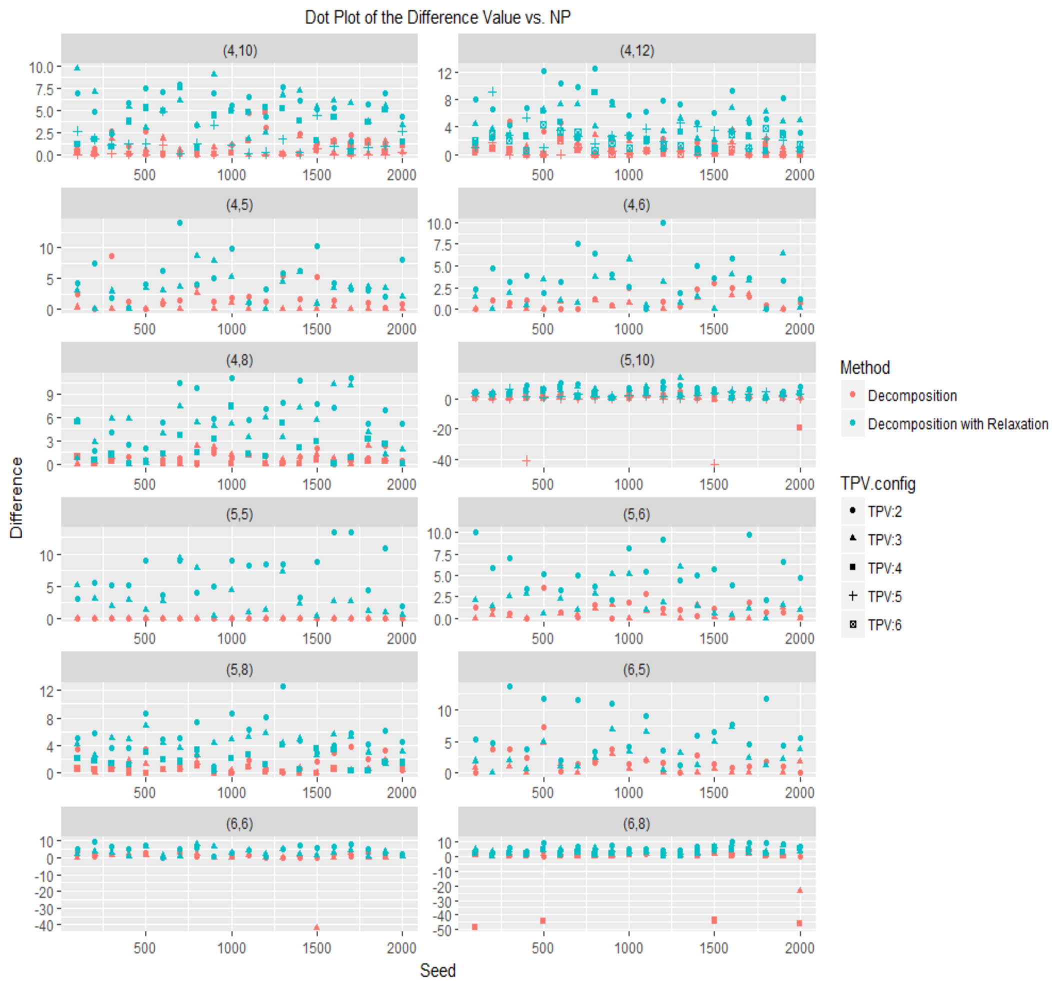


Figure 3.12 Percentage of difference between objective value in exact method and both decomposition methods

For further analysis, one problem out of 9 problems with the large negative percentage of difference in decomposition method (network (5,10) TPV:5) was solved one more time while the maximum time-limit was increased from three to four hours. The result is summarized in Table **3.9**. As shown in Table **3.9**, by increasing the maximum time-limit, the optimality status value changed from 9 to 2 and at the same time, the objective value increased from 926,500,84 to 1,593,191.88. This results in a significant change in the percentage of the difference between the objective value of decomposition and exact method. Moreover, the lowest and highest MIP gap in the exact method for corresponding problems (9 problems) is 5.6% and 12.9% respectively. It shows that these negative outliers are more attributed to the second scenario. Table **3.8** illustrates the problems (outliers) with the large negative percentage of difference.

Table 3.8 Problems with large negative value for percentage of difference between objective values

Method	Seed	NP	TPV	Time (s)	Objective	Status	Difference %	MIP Gap in Corresponding Exact Method Problem
Decomposition	100	(6,8)	4	10810.77	629170.4	9	-48.72	0.092
Decomposition	2000	(6,8)	4	10821.65	717189.2	9	-45.79	0.129
Decomposition	500	(6,8)	4	10824.07	733205.7	9	-44.54	0.095
Decomposition	1500	(6,8)	4	10814.39	700908.6	9	-43.83	0.102
Decomposition	1500	(5,10)	5	10820.55	749908.9	9	-42.93	0.087
Decomposition	1500	(6,6)	3	10813.47	550626.2	9	-41.76	0.069
Decomposition	400	(5,10)	5	10812.23	926500.8	9	-40.92	0.056
Decomposition	2000	(6,8)	3	10822.48	1010458	9	-23.63	0.129
Decomposition	2000	(5,10)	4	10816.01	1146734	9	-19.25	0.090

Table 3.9 Revised solved problem after increasing the maximum time limit to 4 hours

Maximum time-limit (hrs)	**NP**	**Seed**	**TPV**	**Time (hrs)**	**Objective**	**Status**	**Difference %**
3	(5,10)	400	5	3.0034	926,500.84	9	-40.920
4	(5,10)	400	5	3.1368	1,593,191.88	2	1.5926

Therefore, as the above discussion indicates, the large negative difference (in decomposition method) along with an optimality status value 9 results in an invalid solution for decomposition method. For this reason, to implement an accurate statistical analysis, all affected problems should be removed from the dataset.

3.6.2 Statistical Analysis

As discussed in the previous section, in decomposition method while TPV increases, the average solution time will increase similarly. This is evidence that the decomposition method with the smallest TPV(TPV=2), is the fastest method to solve each network problem compared to other TPVs. However, to verify that, problems can be solved to optimality as well, the dependent t-test which is a type of repeated measures statistical test was implemented.

To check the normality (Table **3.10**) and homogeneity of variances (Table **3.11**) assumptions Shapiro-Wilk and Bartlett's tests were implemented. As shown in Table **3.10**, the p-value in decomposition method for networks (6, 8) TPV:4 and (5, 10) TPV:5 is less than the significance level of 0.05. It indicates that the objective value of these problems is not normally distributed.

Table 3.10 Normality test of objective values at 0.05 confidence level

Node-Time period	TPV	Decomposition Method	Decomposition with Relaxation Method	Exact Method
(4,5)	2	W = 0.9549, p-value = 0.4477	W = 0.92268, p-value = 0.1116	W = 0.93727, p-value = 0.2128
	3	W = 0.93747, p-value = 0.2147	W = 0.93809, p-value = 0.2206	
(4,6)	2	W = 0.94665, p-value = 0.3191	W = 0.95798, p-value = 0.5043	W = 0.95783, p-value = 0.5015
	3	W = 0.96239, p-value = 0.5927	W = 0.94833, p-value = 0.3425	
(5,5)	2	W = 0.94687, p-value = 0.3221	W = 0.9218, p-value = 0.1073	W = 0.95006, p-value = 0.3681
	3	W = 0.96462, p-value = 0.6397	W = 0.96079, p-value = 0.5597	
(5,6)	2	W = 0.9255, p-value = 0.1265	W = 0.94817, p-value = 0.3401	W = 0.93984, p-value = 0.2381
	3	W = 0.9454, p-value = 0.3026	W = 0.92569, p-value = 0.1275	
(6,5)	2	W = 0.95521, p-value = 0.4531	W = 0.93353, p-value = 0.1805	W = 0.96032, p-value = 0.5503
	3	W = 0.9502, p-value = 0.3702	W = 0.94811, p-value = 0.3393	
(6,6)	2	W = 0.97322, p-value = 0.8208	W = 0.96968, p-value = 0.7481	W = 0.98154, p-value = 0.9524
	3	W = 0.92423, p-value = 0.1196	W = 0.97683, p-value = 0.8869	
(4,8)	2	W = 0.9404, p-value = 0.244	W = 0.96092, p-value = 0.5624	W = 0.95235, p-value = 0.4042
	3	W = 0.95866, p-value = 0.5174	W = 0.97686, p-value = 0.8874	
	4	W = 0.95092, p-value = 0.3813	W = 0.95215, p-value = 0.4009	
(5,8)	2	W = 0.93105, p-value = 0.1618	W = 0.92312, p-value = 0.1138	W = 0.94138, p-value = 0.2546
	3	W = 0.94901, p-value = 0.3523	W = 0.9479, p-value = 0.3364	
	4	W = 0.93942, p-value = 0.2338	W = 0.90554, p-value = 0.05243	
(6,8)	2	W = 0.95921, p-value = 0.5282	W = 0.9154, p-value = 0.08085	W = 0.93431, p-value = 0.1868
	3	W = 0.93519, p-value = 0.1942	W = 0.91857, p-value = 0.093	
	4	W = 0.76494, p-value = 0.0002712 ***	W = 0.95807, p-value = 0.5061	
(4,10)	2	W = 0.95136, p-value = 0.3882	W = 0.95783, p-value = 0.5015	W = 0.95248, p-value = 0.4062
	3	W = 0.93906, p-value = 0.2301	W = 0.9433, p-value = 0.2766	
	4	W = 0.94827, p-value = 0.3415	W = 0.96277, p-value = 0.6004	
	5	W = 0.95539, p-value = 0.4563	W = 0.95408, p-value = 0.4333	
(5,10)	2	W = 0.95926, p-value = 0.5292	W = 0.9771, p-value = 0.8914	W = 0.94426, p-value = 0.2882
	3	W = 0.95443, p-value = 0.4394	W = 0.96289, p-value = 0.6031	
	4	W = 0.94471, p-value = 0.2939	W = 0.96394, p-value = 0.6251	
	5	W = 0.78029, p-value = 0.0004456 ***	W = 0.9297, p-value = 0.1524	
(4,12)	2	W = 0.95197, p-value = 0.398	W = 0.94782, p-value = 0.3352	W = 0.93914, p-value = 0.231
	3	W = 0.94389, p-value = 0.2836	W = 0.94236, p-value = 0.2656	
	4	W = 0.94105, p-value = 0.251	W = 0.95539, p-value = 0.4563	
	5	W = 0.94618, p-value = 0.3128	W = 0.91877, p-value = 0.09382	
	6	W = 0.94914, p-value = 0.3542	W = 0.94702, p-value = 0.3241	

Table 3.11 Summary table of Bartlett's test (Homogeneity of Variances)

No.	Node-Time Period	Decomposition Method	Decomposition with Relaxation Method
1	(4,5)	Bartlett's K-squared = 0.054385, df = 1, p-value = 0.8156	Bartlett's K-squared = 0.0059358, df = 1, p-value = 0.9386
2	(4,6)	Bartlett's K-squared = 0.0033415, df = 1, p-value = 0.9539	Bartlett's K-squared = 0.075765, df = 1, p-value = 0.7831
3	(5,5)	Bartlett's K-squared = 0.56673, df = 1, p-value = 0.4516	Bartlett's K-squared = 0.2022, df = 1, p-value = 0.653
4	(5,6)	Bartlett's K-squared = 6.699e-07, df = 1, p-value = 0.9993	Bartlett's K-squared = 0.21316, df = 1, p-value = 0.6443
5	(6,5)	Bartlett's K-squared = 0.061084, df = 1, p-value = 0.8048	Bartlett's K-squared = 0.12279, df = 1, p-value = 0.726
6	(6,6)	Bartlett's K-squared = 2.3531, df = 1, p-value = 0.125	Bartlett's K-squared = 0.068022, df = 1, p-value = 0.7942
7	(4,8)	Bartlett's K-squared = 0.059002, df = 2, p-value = 0.9709	Bartlett's K-squared = 0.33418, df = 2, p-value = 0.8461
8	(5,8)	Bartlett's K-squared = 0.018475, df = 2, p-value = 0.9908	Bartlett's K-squared = 0.4223, df = 2, p-value = 0.8097
9	(6,8)	Bartlett's K-squared = 32.845, df = 2, p-value = 7.376e-08*	Bartlett's K-squared = 0.43458, df = 2, p-value = 0.8047
10	(4,10)	Bartlett's K-squared = 0.1231, df = 3, p-value = 0.9889	Bartlett's K-squared = 0.1318, df = 3, p-value = 0.9878
11	(5,10)	Bartlett's K-squared = 12.006, df = 3, p-value = 0.007364*	Bartlett's K-squared = 1.2059, df = 3, p-value = 0.7516
12	(4,12)	Bartlett's K-squared = 0.042989, df = 4, p-value = 0.9998	Bartlett's K-squared = 0.22502, df = 4, p-value = 0.9941

Table **3.11** illustrates that p-value in the same problems ((6,8) and (5,10)), is less than the significance level of 0.05. Therefore, we conclude that at least one TPV variance in these two network problems is different from the others. Earlier in Table **3.8**, we discussed 9 problems which are noticeable outliers and different two scenarios that could cause this situation. However, Tables **3.10** and **3.11** evidence that some problems (especially the largest TPV) in networks (6, 8) and 5, 10) violated the normality and homogeneity of variances assumptions. This could be concluded that to implement a valid statistical analysis, these problems should be removed from the original dataset.

Table 3.12 Comparison of normality test results before and after removing outliers

Node-Time period	TPV	Before	After
(6,6)	3	W = 0.92423, p-value = 0.1196	W = 0.98613, p-value = 0.9895
(6,8)	3	W = 0.93519, p-value = 0.1942	W = 0.92902, p-value = 0.1661
	4	W = 0.76494, p-value = 0.0002712 *	W = 0.94144, p-value = 0.367
(5,10)	4	W = 0.94471, p-value = 0.2939	W = 0.93328, p-value = 0.199
	5	W = 0.78029, p-value = 0.0004456 *	W = 0.93972, p-value = 0.2866

The normality test results after removing the outliers (in specific problems) are shown in Table **3.12**. As you can see in this table, after removing 9 outliers from effected network problems ((6,8), (5,10)) the corresponding objective values were normally distributed. Also, Table **3.13** proves that after removing the outliers, statistically there is not any significant difference between TPV variances in problems (6,8), (6,6) and (5,10).

Table 3.13 Comparison of homogeneity variances test before and after Removing outliers

Node-Time Period	Before	After
(6,8)	Bartlett's K-squared = 32.845, df = 2, p-value = 7.376e-08***	Bartlett's K-squared = 0.083186, df = 2, p-value = 0.9593
(6,6)	Bartlett's K-squared = 2.3531, df = 1, p-value = 0.125	Bartlett's K-squared = 0.071741, df = 1, p-value = 0.7888
(5,10)	Bartlett's K-squared = 12.006, df = 3, p-value = 0.007364 **	Bartlett's K-squared = 0.23306, df = 3, p-value = 0.9721

Figure **3.13** and **3.14** visually display the percentage of average difference in different problems and different TPV configurations in decomposition method and decomposition method with relaxation, respectively. As shown in Figures **3.13** and **3.14**, in all problems excluding (4,10) and (4,12), while TPV increases the percentage of average difference decreases in both decomposition methods. In other words, in larger TPV configurations, the decomposed network includes a larger portion of original TS network, therefore, in each problem, the objective value found by the largest TPV in decomposition methods is closer to the objective value found by exact method.

Additionally, the lowest and highest percentage of average difference in decomposition method is 0% and ~2.2%, respectively which is noticeably smaller than the corresponding value in decomposition method with relaxation (~1.5% and ~7.3%). Also, Tables **3.14** and **3.15** display the updated version of Tables **3.4** and **3.7**, average solution time and the average percentage of difference tables, after removing the outliers.

As shown in Table **3. 15**, by removing the outliers from problems (6,8), (6,6) and (5,10) solved by decomposition method, the lowest percentage of average difference increased from -8.75 to 0.

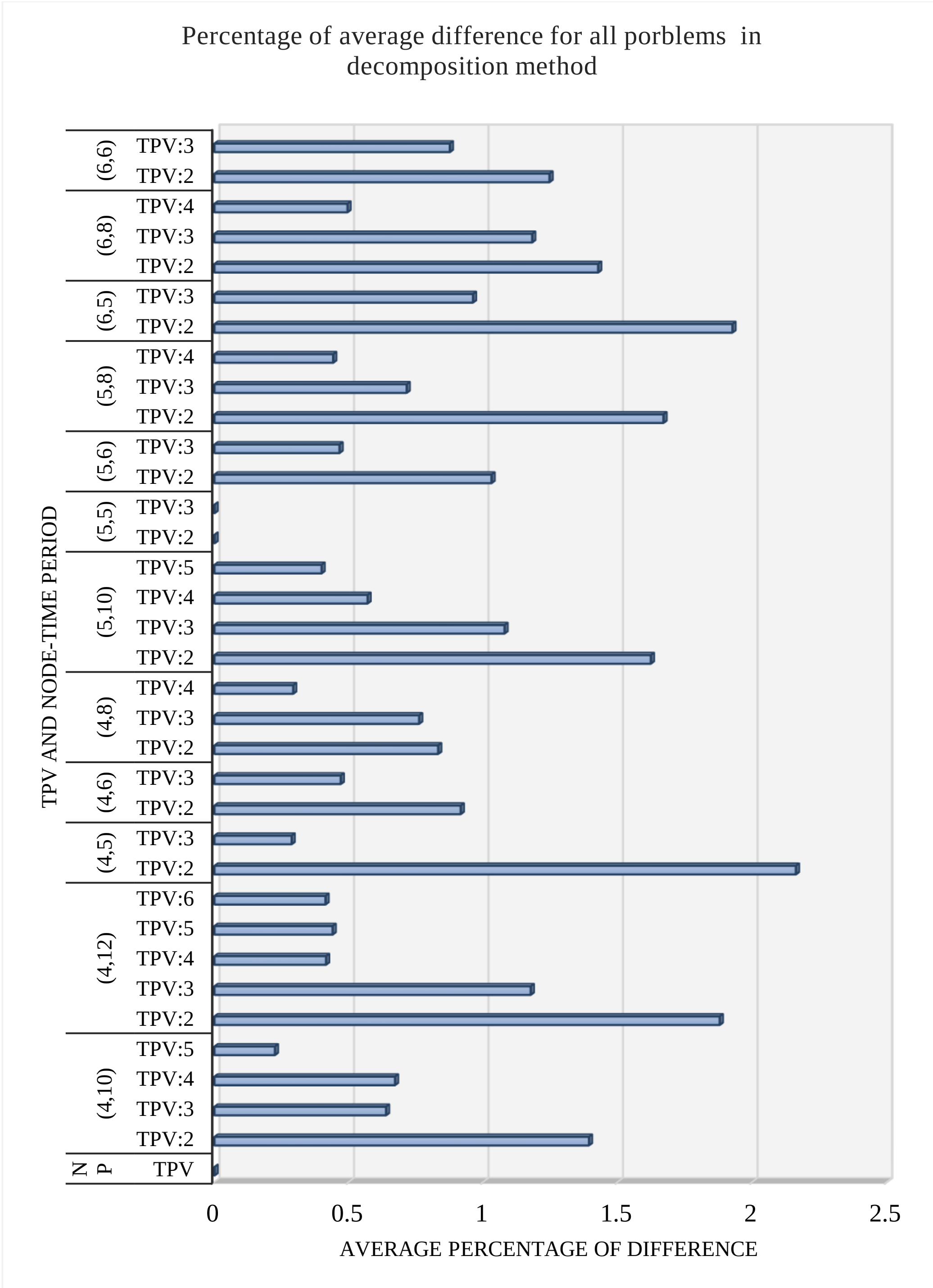


Figure 3.13 Average percentage of difference in decomposition method for different problems and TPV configurations

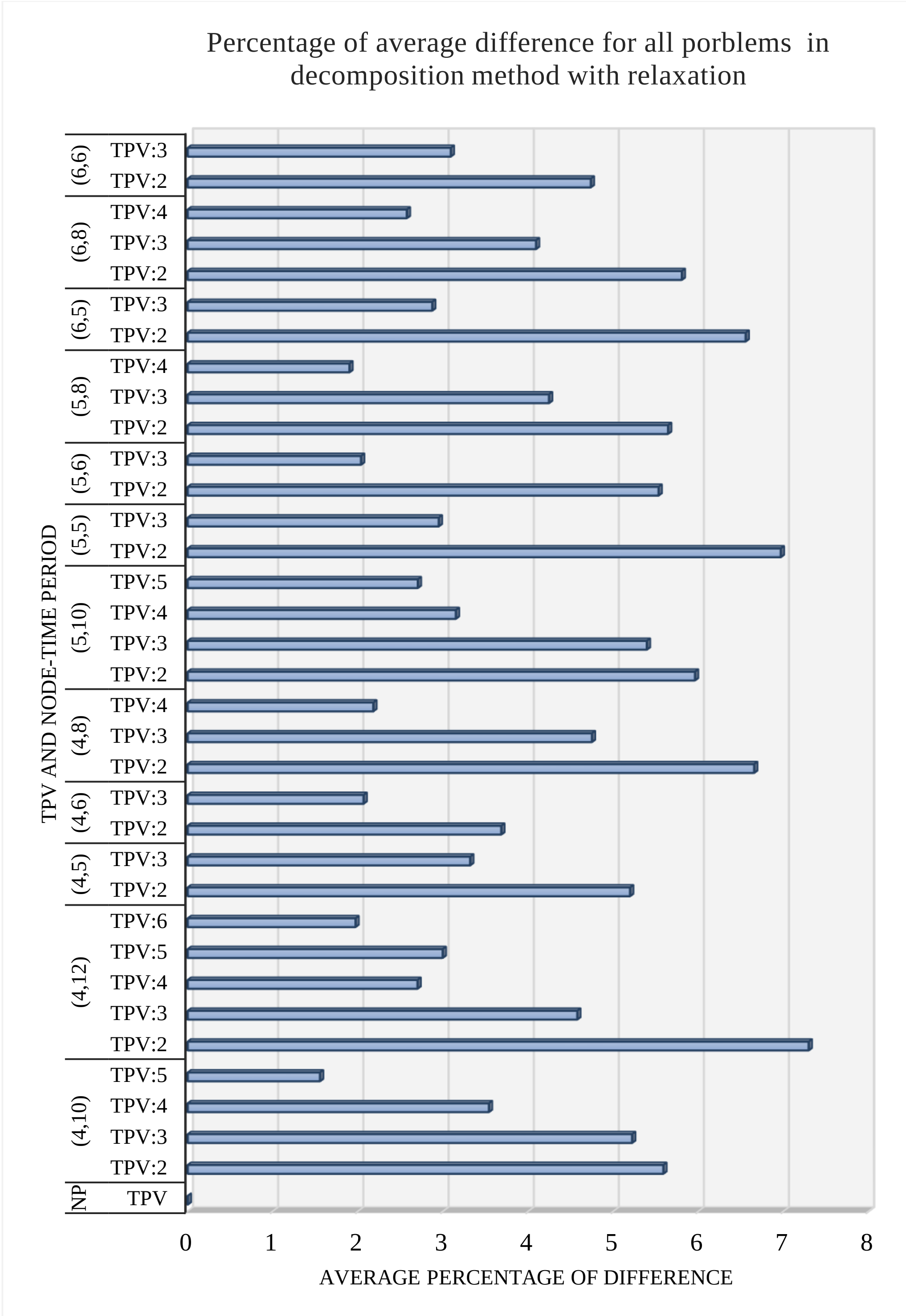


Figure 3.14 Average percentage of difference in decomposition method with relaxation for different problems and TPV configurations

Table 3.14 Revised table of average solution time (seconds) for different methods after removing outliers

Node-Time Period	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Exact Method		**3.13**	**94.89**	**1008.67**	**3070.67**	**4957.51**	**9193.24**	**441.21**	**8755.79**	**10743.22**	**10259.71**	**4021.69**	**6937.53**
Decomposition method	**2**	0.09	0.16	0.69	1.04	24.92	7.32	0.18	1.17	22.72	1.71	0.26	0.4
	3	0.23	0.47	8.4	14.41	217.7	540.44	0.57	41.6	711.02	318.9	0.97	1.57
	4							2.1	174.75	4970.39	355.28	3.5	11.96
	5										1959.63	11.69	14.33
	6												485.1
Decomposition method with Relaxation	**2**	0.042	0.08	0.11	0.28	0.44	2.09	0.1055	0.43	2.87	0.43	0.13	0.22
	3	0.0846	0.37	0.96	57.35	6.97	649.06	0.1726	1.96	46.09	2.19	0.22	0.58
	4							1.6391	83.07	3942.05	23.34	1.06	2.37
	5										2284.34	8.32	2.8
	6												295.28

Table 3.15 Revised table of average percentage of difference between objective value of exact method and decomposition methods after removing the outliers

Node-Time Period	TPV	(4, 5)	(4, 6)	(5, 5)	(5, 6)	(6, 5)	(6, 6)	(4, 8)	(5, 8)	(6, 8)	(5, 10)	(4, 10)	(4, 12)
Decomposition method	2	2.16	0.92	0.00	1.03	1.93	1.25	0.83	1.67	1.43	1.62	1.39	1.88
	3	0.29	0.47	0.00	0.47	0.96	0.87	0.76	0.71	1.18	1.08	0.64	1.18
	4							0.29	0.44	0.49	0.57	0.67	0.42
	5										0.40	0.22	0.44
	6												0.41
Decomposition method with Relaxation	2	5.19	3.68	6.97	5.53	6.55	4.73	6.65	5.64	5.80	5.96	5.58	7.29
	3	3.32	2.06	2.95	2.03	2.87	3.09	4.74	4.24	4.09	5.39	5.22	4.57
	4							2.18	1.90	2.57	3.15	3.53	2.70
	5										2.70	1.55	2.99
	6												1.97

To verify the accuracy and efficiency of proposed decomposition methods to find near-optimal objective values, the dependent t-test which is a type of repeated measures statistical test was implemented. Tables **3.16** and **3.17** indicate the paired t-test results for decomposition method and decomposition method with relaxation, respectively (after removing outliers). Note that the significance level was set to be 0.05. These two tables include information namely node-period, tested-pair, the degree of freedom, lower bound and upper bound of 95% confidence interval, mean of differences (sample estimate), t-value, and p-value.

Table 3.16 Paired t-test for the objective values in decomposition method

Paired t-test for objective value in decomposition method							
			95% confidence interval		Sample estimate		
NP	**Tested Pair**	**Df**	**lower bound**	**upper bound**	**Mean of the differences**	**t-value**	**p-value**
(4,5)	TPV2-Exact	19	-18690.92	-6207.36	-12449.14	-4.17	0.000514
	TPV3-Exact	19	-3520.56	101.01	-1709.78	-1.98	0.06283
(4,6)	TPV2-Exact	19	-9641.87	-3219.77	-6430.82	-4.19	0.000495
	TPV3-Exact	19	-5145.12	-1401.96	-3273.54	-3.66	0.001662
(5,5)	TPV2-Exact	19	-30882.84	-6898.24	-18890.54	-3.30	0.003791
	TPV3-Exact	19	-10854.20	-2077.25	-6465.72	-3.08	0.006113
(5,6)	TPV2-Exact	19	-12169.33	-4845.43	-8507.38	-4.86	0.000108
	TPV3-Exact	19	-5941.48	-1806.85	-3874.16	-3.92	0.000915
(6,5)	TPV2-Exact	19	-21663.52	-8736.07	-15199.80	-4.92	9.47E-05
	TPV3-Exact	19	-11784.85	-2926.12	-7355.48	-3.48	0.002531
(6,6)	TPV2-Exact	19	-15980.30	-6538.93	-11259.61	-4.99	8.09E-05
	TPV3-Exact	18	-12505.68	-3845.78	-8175.73	-3.97	0.000904
(4,8)	TPV2-Exact	19	-10609.28	-5046.02	-7827.65	-5.89	1.14E-05
	TPV3-Exact	19	-10202.86	-3740.34	-6971.60	-4.52	0.000237
	TPV4-Exact	19	-4283.51	-911.93	-2597.72	-3.23	0.004455
(5,8)	TPV2-Exact	19	-25339.75	-12076.84	-18708.30	-5.90	1.10E-05
	TPV3-Exact	19	-11352.27	-5070.79	-8211.53	-5.47	2.80E-05
	TPV4-Exact	19	-7352.39	-2589.71	-4971.05	-4.37	0.00033
(6,8)	TPV2-Exact	19	-23800.13	-12278.04	-18039.09	-6.55	2.83E-06
	TPV3-Exact	18	-20675.58	-9163.26	-14919.42	-5.45	3.58E-05
	TPV4-Exact	15	-11751.67	-711.74	-6231.71	-2.41	0.02946
(4,10)	TPV2-Exact	19	-23567.74	-9713.18	-16640.46	-5.03	7.47E-05
	TPV3-Exact	19	-11894.04	-3647.88	-7770.96	-3.94	0.000869
	TPV4-Exact	19	-13662.63	-1802.61	-7732.62	-2.73	0.01332
	TPV5-Exact	19	-4785.44	-471.38	-2628.41	-2.55	0.01954
(5,10)	TPV2-Exact	19	-29700.83	-15096.68	-22398.76	-6.42	3.72E-06
	TPV3-Exact	19	-20676.94	-9369.65	-15023.30	-5.56	2.30E-05
	TPV4-Exact	18	-11833.82	-4003.57	-7918.70	-4.25	0.000482
	TPV5-Exact	17	-9560.33	-1387.60	-5473.97	-2.83	0.01164
(4,12)	TPV2-Exact	19	-37234.80	-17488.06	-27361.43	-5.80	1.38E-05
	TPV3-Exact	19	-22386.52	-11543.33	-16964.92	-6.55	2.85E-06
	TPV4-Exact	19	-9700.95	-2254.10	-5977.52	-3.36	0.003288
	TPV5-Exact	19	-9978.21	-2539.06	-6258.63	-3.52	0.00228
	TPV6-Exact	19	-9670.847	-2200.685	-5935.766	-3.326	0.003549

Table 3.17 Paired t-test for the objective values in decomposition method with relaxation

Paired t-test for objective value in decomposition method with relaxation							
			95% confidence interval		Sample estimate		
NP	**Tested Pair**	**Df**	**lower bound**	**upper bound**	**Mean of the differences**	**t-value**	**p-value**
(4,5)	TPV2-Exact	19	-38313.62	-20382.53	-29348.07	-6.85	0.000001544
	TPV3-Exact	19	-25112.76	-12388.73	-18750.74	-6.17	0.000006287
(4,6)	TPV2-Exact	19	-32783.90	-17851.65	-25317.78	-7.10	9.454E-07
	TPV3-Exact	19	-21209.76	-7946.77	-14578.26	-4.60	0.0001949
(5,5)	TPV2-Exact	19	-58323.85	-36260.91	-47292.38	-8.97	2.927E-08
	TPV3-Exact	19	-27098.14	-12120.60	-19609.37	-5.48	0.00002749
(5,6)	TPV2-Exact	19	-55311.15	-36551.23	-45931.19	-10.25	3.536E-09
	TPV3-Exact	19	-22521.38	-10554.83	-16538.11	-5.79	0.00001421
(6,5)	TPV2-Exact	19	-63333.03	-37663.40	-50498.22	-8.23	1.087E-07
	TPV3-Exact	19	-29109.77	-14342.90	-21726.34	-6.16	0.000006419
(6,6)	TPV2-Exact	19	-55211.05	-32718.46	-43964.75	-8.18	1.20E-07
	TPV3-Exact	19	-37210.48	-19308.35	-28259.42	-6.61	0.00000253
(4,8)	TPV2-Exact	19	-74452.43	-49020.70	-61736.57	-10.16	4.06E-09
	TPV3-Exact	19	-56183.62	-32137.54	-44160.58	-7.69	3.012E-07
	TPV4-Exact	19	-28342.15	-11333.34	-19837.75	-4.88	0.0001035
(5,8)	TPV2-Exact	19	-77105.96	-50429.62	-63767.79	-10.01	5.21E-09
	TPV3-Exact	19	-55389.65	-39839.82	-47614.73	-12.82	8.44E-11
	TPV4-Exact	19	-26681.95	-15542.89	-21112.42	-7.93	1.895E-07
(6,8)	TPV2-Exact	19	-87574.48	-59628.95	-73601.72	-11.03	1.07E-09
	TPV3-Exact	19	-61848.04	-41061.10	-51454.57	-10.36	2.96E-09
	TPV4-Exact	19	-42603.45	-22764.11	-32683.78	-6.90	0.000001412
(4,10)	TPV2-Exact	19	-75298.30	-55674.24	-65486.27	-13.97	1.91E-11
	TPV3-Exact	19	-72300.90	-49396.89	-60848.89	-11.12	9.258E-10
	TPV4-Exact	19	-52515.38	-30254.61	-41384.99	-7.78	2.519E-07
	TPV5-Exact	19	-25151.25	-10593.25	-17872.25	-5.14	0.00005833
(5,10)	TPV2-Exact	19	-103275.68	-63204.98	-83240.33	-8.70	4.75E-08
	TPV3-Exact	19	-92743.08	-56475.28	-74609.18	-8.61	5.52E-08
	TPV4-Exact	19	-58330.38	-30267.41	-44298.89	-6.61	0.00000253
	TPV5-Exact	19	-45332.55	-28533.85	-36933.20	-9.20	1.97E-08
(4,12)	TPV2-Exact	19	-123233.74	-88531.37	-105882.60	-12.77	8.97E-11
	TPV3-Exact	19	-77945.38	-54171.71	-66058.55	-11.63	4.38E-10
	TPV4-Exact	19	-53149.34	-25314.79	-39232.07	-5.90	0.00001111
	TPV5-Exact	19	-57812.68	-30243.29	-44027.99	-6.69	0.000002162
	TPV6-Exact	19	-37080.45	-20472.79	-28776.62	-7.2533	6.958E-07

Table **3.16** and **3.17** indicate that the p-value for all problems is less than the significance level of 0.05 except problem (4,5), where the corresponding p-value in comparison of the mean objective value of exact method and decomposition method with TPV:3 is roughly greater than the significance level of 0.05. Therefore, we conclude that statistically there is a significant difference in the mean objective value of each TPV configuration and corresponding objective value in the exact method. In other words, regardless of which TPV configuration was used to solve a problem, the mean objective values for various TPVs are statistically different from the mean objective value in the exact method.

According to the above discussion, while the average solution time for each problem set is significantly shorter when smaller TPV configurations are used, the mean objective values are statistically significant between different TPV configuration and exact method in both decomposition methods.

3.7 Conclusions

In this chapter, a decomposition heuristic is proposed that subdivides the single-commodity time-space FCNF problem into various time epochs to create smaller and more manageable subproblems. These subproblems are solved sequentially to find an overall solution for the original problem. To evaluate the efficiency of the investigated method an experiment included a total of 1600 problem is designed. All required parameters generated randomly from a uniform probability distribution and problems were solved using Gurobi MIP solver, which runs parallel branch & bound algorithm.

Analysis indicates that 112 problems out of 1600 problems are not solved to optimality due to the maximum time-limit of three hours. Also, 95 problems out of 112 unsolved problems occurred in the exact method. Dependent t-test which is a type of repeated measures statistical test is implemented and the outputs indicate that statistically, there is a significant difference between the mean objective value of each TPV configuration and the exact method.

Additionally, depending on the problem specification (i.e., number of nodes and periods) and TPV configuration used, the average solution time could be reduced by up to four orders of magnitude in both decomposition methods compared to the exact method (maximum gap in decomposition method ~ 2.2%).

As shown in Figures **3.13** and **3.14** there is always a trade-off between the accuracy and solution time. While statistically there is a significant difference between the mean objective value of exact method and each TPV configuration, however, the average difference may not be a serious concern for many practical large-scale time-consuming problems. In other words, it still can show a great promise for decomposition method to significantly reduce the solution time which has been an outstanding issue in complicated large-scale problems up to four orders of magnitude, while the average gap between the objective value of exact method and decomposition method is reasonably small (min and max gap is 0.0 and 2.16%)

Chapter 4: Conclusions and Future Works

A wide range of network flow problems primarily used in the area of transportation and logistics is categorized as time-space fixed charge network flow (FCNF) problems. In this family of networks, each node *i* is associated with a specific time *r* and is replicated across all time-periods. There is another specification (in this work) such that each arc only connects each node to another node in the future or the same time-period. The cost structure in fixed charge network flow problems consists of variable and fixed costs where continuous and binary variables are required to formulate the problem in hand as a mixed integer linear programming.

FCNF problems are classified as NP-hard problems, by increasing the size of the problem the solution time and complexity of problem will exponentially increase. Moreover, adding another component (i.e., time) results in a more complex, time-consuming and CPU and memory intensive. Generally, in NP-hard problems heuristic methods are highly desired. By sacrificing some factors such as optimality, precision, and accuracy to a certain level, heuristic methods can solve most of the time-consuming problems much faster.

One of the efficient heuristic algorithms in mathematical programming for solving large-scale problems is decomposition algorithm. In this algorithm, the problem is broken into a set of subproblems where these subproblems are iteratively solved to find an optimal solution for the original problem. The earliest works on decomposition algorithm date back to the seminal work of Dantzig and Wolfe and Benders. Later, this method was further explored and expanded by other researchers to address emerging problems in operation research.

In this study, a heuristic method based on decomposition algorithm is investigated. An experiment which includes a total of 1600 problems (12 time-space networks) is implemented. The primary goal of this experiment is to evaluate the capability and efficiency of the decomposition methods vs. the exact method. The main required parameters namely requirement at each node, variable cost and fixed cost on each arc are generated randomly from a uniform probability distribution. The experiment is designed in such a way that 40%, 45% and 15% of time-space nodes are demand, supply, and transshipment nodes, respectively. Also, the problem is considered as a feasible problem if the total demand in each time-period does not exceed the total supply in the same period.

Kennington and Nicholson [17] have categorized and examined different types of FCNF problems by varying the range of requirements. In [17], FCNF problems were classified into 27 classes such as LLL, LLM, LLH where the first, second and third letter shows the range of node requirement, variable cost and fixed cost (L=low, M=medium, H=high), correspondingly. Their study indicates that among 27 classes of problems those with a low level of variable cost and a high level of fixed cost are the most time-consuming problems. For this reason, HLH problems, as one of the most challenging types of problems, are explored in this work.

Generally, linear programming-based branch and bound algorithm are used to solve mixed integer linear programming problems. In this study, the problems are solved by Gurobi MIP solver, which runs parallel branch & bound algorithm. The investigated decomposition approach subdivides the problem into various time epochs to create smaller and more manageable subproblems. These subproblems are solved sequentially to find an overall solution for the original problem.

Analysis indicates that 112 problems out of 1600 problems were not solved to optimality due to the maximum time-limit of three hours and 95 problems out of 119 unsolved problems occurred in the exact method. The most complicated and time-consuming networks in the exact method are (6, 8), (5, 10), (6, 6) and (5, 8) with approximately 60, 57, 51 and 49 hours solution time (for 20 runs), respectively.

Further analysis indicates that depending on the problem specification (i.e., number of nodes and periods) and TPV configuration used, the average solution time could be reduced by up to four orders of magnitude (compared to average solution time in Exact method) in decomposition methods. Also, in decomposition method, while TPV increases, the average solution time will increase similarly. This is evidence that the decomposition method with the smallest TPV configuration(TPV=2), can be considered as the fastest method to solve each network problem to optimality compared to larger TPV configurations.

To validate the above statement, dependent t-test (paired t-test) which is a type of repeated measures statistical test is implemented. Statistical analysis indicates that there is statistically a significant difference between the mean objective value of exact method and each TPV configuration in both decomposition methods excluding problem (4,5) in decomposition method which roughly does not indicate any significant difference between the mean objective value of the exact method and TPV:3 at 0.05 significance level.

While there is a significant difference between the mean objective value of the exact method and each TPV in decomposition methods, however, the percentage of the average difference between the objective values is reasonably small (average between 0-2.16% in decomposition method).

According to the above discussion, we conclude that despite the statistically significant difference between the mean objective value of exact method and each TPV configuration in decomposition method, still, this approach is considered as a fast and efficient method to solve the time-space FCNF problems with a reasonable level of accuracy. In other words, there is always a trade-off between the accuracy and solution time. Finally, this shows great promise for decomposition method to significantly reduce the solution time which has been an outstanding issue in complicated large-scale problems.

Compared to single commodity FCNF problems the multi-commodity problems are more complicated, time-consuming and memory and CPU intensive. Adding another component (i.e., time) to this type of networks increases the complexity of problem which will increase the solution time remarkably. Therefore, as a future study, the decomposition approach will be expanded to the multi-commodity time-space FCNF problems and the same statistical analysis will be implemented to validate this methodology.

References

[1] G. Angulo and M. Van Vyve, “Fixed-charge transportation problems on trees,” *Oper. Res. Lett.*, vol. 45, no. 3, pp. 275–281, 2017.

[2] K. A. A. D. Raj and C. Rajendran, “A Hybrid Genetic Algorithm for Solving Single-Stage Fixed-Charge Transportation Problems,” *Technol. Oper. Manag.*, vol. 2, no. 1, pp. 1–15, 2011.

[3] R. Roberti, E. Bartolini, A. Mingozzi, R. Roberti, and E. Bartolini, “The Fixed Charge Transportation Problem : An Exact Formulation The Fixed Charge Transportation Problem : An Exact Algorithm Based on a New Integer Programming Formulation,” no. September, 2015.

[4] a. Diabat, T. Aouam, and O. Al-Araidah, “The uncapacitated fixed-charge facility location problem with a multi-echelon inventory system,” *2009 Int. Conf. Comput. Ind. Eng.*, pp. 815–819, 2009.

[5] L. K. Nozick, “The fixed charge facility location problem with coverage restrictions,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 37, no. 4, pp. 281–296, 2001.

[6] M. S. Daskin, W. J. Hopp, and B. Medina, “Forecast horizons and dynamic facility location planning,” *Ann. Oper. Res.*, vol. 40, no. 1, pp. 125–151, 1992.

[7] G. M., V. C.J., and D. K., “Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms,” *Eur. J. Oper. Res.*, vol. 143, no. 1, pp. 1–18, 2002.

[8] A. Fakhri and M. Ghatee, “Application of Benders decomposition method in solution of a fixed-charge multicommodity network design problem avoiding

congestion," *Appl. Math. Model.*, vol. 40, no. 13–14, pp. 6468–6476, 2016.

[9] Y. K. Agarwal and Y. P. Aneja, "Fixed charge multicommodity network design using p-partition facets," *Eur. J. Oper. Res.*, vol. 258, no. 1, pp. 124–135, 2017.

[10] A. M. Costa, "A survey on benders decomposition applied to fixed-charge network design problems," *Comput. Oper. Res.*, vol. 32, no. 6, pp. 1429–1450, 2005.

[11] C. Lagos *et al.*, "A Matheuristic Approach Combining Local Search and Mathematical Programming," *Sci. Program.*, vol. 2016, pp. 1–7, 2016.

[12] F. Dunke, "KIT - IOR - DOL - Facility Location," 2017.

[13] J. Zhang, L. Jia, S. Niu, F. Zhang, L. Tong, and X. Zhou, "A space-time network-based modeling framework for dynamic unmanned aerial vehicle routing in traffic incident monitoring applications," *Sensors (Switzerland)*, vol. 15, no. 6, pp. 13874–13898, 2015.

[14] S. Yan, C. Y. Chen, and C. C. Wu, "Solution methods for the taxi pooling problem," *Transportation (Amst).*, vol. 39, no. 3, pp. 723–748, 2012.

[15] N. Kliewer, T. Mellouli, and L. Suhl, "A time–space network based exact optimization model for multi-depot bus scheduling," 2005.

[16] B. He, R. Song, S. He, and Y. Xu, "High-speed rail train timetabling problem: A time-space network based method with an improved branch-and-price algorithm," *Math. Probl. Eng.*, vol. 2014, 2014.

[17] J. L. Kennington and C. D. Nicholson, "The uncapacitated time-space fixed-charge network flow problem: An empirical investigation of procedures for arc capacity assignment," *INFORMS J. Comput.*, vol. 22, no. 2, pp. 326–337, 2010.

[18] "SkyMAX – Optym." [Online]. Available: https://www.optym.com/sky/skymax/. [Accessed: 28-Oct-2017].

[19] M. R. Garey and D. S. Johnson, *Computers and Intractibility: A guide to the theory of NP-Completeness*. .

[20] W. M. Hirsch and G. B. Dantzig, "The fixed charge problem," *Nav. Res. Logist. Q.*, vol. 15, no. 3, pp. 413–424, 1968.

[21] "Exact Solution of the Fixed-Charge Transportation Problem Author (s): Paul Gray Source : Operations Research , Vol . 19 , No . 6 , Guidelines for the Practice of Operations Published by : INFORMS Stable URL : http://www.jstor.org/stable/169254 REFERENCE," vol. 19, no. 6, pp. 1529–1538, 2017.

[22] U. S. Palekar, M. H. Karwan, and S. Zionts, "A Branch-and-Bound Method for the Fixed Charge Transportation Problem," *Manage. Sci.*, vol. 36, no. 9, pp. 1092–1105, 1990.

[23] B. Alidaee and G. A. Kochenberger, "A Note on a Simple Dynamic Programming Approach to the Single-Sink, Fixed-Charge Transportation Problem," *Transp. Sci.*, vol. 39, no. 1, pp. 140–143, 2005.

[24] L. Cooper and C. Drebes, "An Approximate Solution Method for the Fixed Charge Problem," *Nav. Res. Logist. Q.*, vol. 14, no. 1, pp. 101–113, 1967.

[25] D. B. Khang and O. Fujiwara, "Approximate solutions of capacitated fixed-charge minimum cost network flow problems," *Networks*, vol. 21, no. 6, pp. 689–704, 1991.

[26] M. Sun, J. E. Aronson, P. G. McKeown, and D. Drinka, "A tabu search heuristic

procedure for the fixed charge transportation problem," *Eur. J. Oper. Res.*, vol. 106, no. 2–3, pp. 441–456, 1998.

[27] T. E. Society, "The Decomposition Algorithm for Linear Programs Author (s): George B . Dantzig and Philip Wolfe Published by : The Econometric Society Stable URL : http://www.jstor.org/stable/1911818 Accessed : 07-06-2016 14 : 21 UTC," vol. 29, no. 4, pp. 767–778, 2016.

[28] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math.*, vol. 4, no. 1, pp. 238–252, 1962.

[29] T. J. V. A. N. Roy, "A Cross Decomposition Algorithm for Capacitated Facility Location Author (s): Tony J . van Roy Published by : INFORMS Stable URL : http://www.jstor.org/stable/170679 REFERENCES Linked references are available on JSTOR for this article : You may need to," *Oper. Res.*, vol. 34, no. 1, pp. 145–163, 1986.

[30] H. D. Sherali and B. M. P. Fraticelli, "A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse," *J. Glob. Optim.*, vol. 22, no. 1, pp. 319–342, 2002.

[31] P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser, "Airline Crew Scheduling: A New Formulation and Decomposition Algorithm," *Oper. Res.*, vol. 45, no. 2, pp. 188–200, 1997.

[32] M. Science, "An Algorithm for the Solution of Mixed Integer Programming Problems Author (s): Norman J . Driebeek Source : Management Science , Vol . 12 , No . 7 , Series A , Sciences (Mar ., 1966), pp . 576-587 Published by : INFORMS Stable URL : http://www.jstor.," vol. 12, no. 7, pp. 576–587, 2017.

[33] “Solving the Fixed Charge Problem by Ranking the Extreme Points Author (s): Katta G . Murty Published by : INFORMS Stable URL : http://www.jstor.org/stable/168755,” vol. 16, no. 2, pp. 268–279, 2017.

[34] M. Science, “A New Branch-and-Bound Algorithm for the Fixed-Charge Transportation Problem Author (s): Jeff Kennington and Ed Unger Published by : INFORMS Stable URL : http://www.jstor.org/stable/2629910 REFERENCES Linked references are available on JSTOR for this ar,” vol. 22, no. 10, pp. 1116–1126, 2017.

[35] N. M. Jun and R. S. Barr, “A New Optimization Method for Large Scale Fixed Charge Transportation Problems Author (s): Richard S . Barr , Fred Glover and Darwin Klingman Published by : INFORMS Stable URL : http://www.jstor.org/stable/170107 REFERENCES Linked references are availab,” vol. 29, no. 3, pp. 448–463, 2017.

[36] M. Science, “Improved Penalties for Fixed Cost Linear Programs Using Lagrangean Relaxation Author (s): A . Victor Cabot and S . Selcuk Erenguc Published by : INFORMS Stable URL : http://www.jstor.org/stable/2631766 REFERENCES Linked references are available on JSTOR,” vol. 32, no. 7, pp. 856–869, 2017.

[37] M. Science, “A Branch-and-Bound Method for the Fixed Charge Transportation Problem Author (s): Udatta S . Palekar , Mark H . Karwan and Stanley Zionts Published by : INFORMS Stable URL : http://www.jstor.org/stable/2632358 REFERENCES Linked references are available ,” vol. 36, no. 9, pp. 1092–1105, 2017.

[38] G. J. Bell, B. W. Lamar, and C. A. Wallace, “Capacity improvement, penalties, and the fixed charge transportation problem,” *Nav. Res. Logist.*, vol. 46, no. 4, pp. 341–355, 1999.

[39] F. R. B. Cruz, J. M. Smith, and G. R. Mateus, “Solving to optimality the uncapacitated fixed-charge network flow problem,” *Comput. Oper. Res.*, vol. 25, no. 1, pp. 67–81, 1998.

[40] F. Ortega and L. Wolsey, “A branch-and-cut algorithm for the single commodity uncapacitated fixed charge network flow problem,” pp. 1–29, 2000.

[41] D. R. Denzler, “an Approximative Algorithm for the Fixed Charge Problem,” *Nav. Res. Logist. Q.*, vol. 16, pp. 411–416, 1969.

[42] M. Science, “A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem Author (s): Warren E . Walker Published by : INFORMS Stable URL : http://www.jstor.org/stable/2629840 PROBLEM *,” vol. 22, no. 5, pp. 587–596, 2017.

[43] M. Sun and P. G. McKeown, “Tabu search applied to the general fixed charge problem,” *Ann. Oper. Res.*, vol. 41, no. 4, pp. 405–420, 1993.

[44] Y.-H. Sun, S. Yang, W. W.-G. Yeh, and P. W. F. Louie, “Modeling Reservoir Evaporation Losses by Generalized Networks,” *J. Water Resour. Plan. Manag.*, vol. 122, no. 3, pp. 222–226, 1996.

[45] D. Kim and P. M. Pardalos, “A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure,” *Oper. Res. Lett.*, vol. 24, no. 4, pp. 195–203, 1999.

[46] M. S. R. Monteiro, D. B. M. M. Fontes, and F. A. C. C. Fontes, “An ant colony

optimization algorithm to solve the minimum cost network flow problem with concave cost functions," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, 2011, p. 139.

[47] J. Holland, "Adaptation in Natural and Artificial Systems," 1975.

[48] Z. Michalewicz, G. Vignaux, and M. Hobbs, "A Nonstandard Genetic Algorithm for the Nonlinear Transportation Problem," *ORSA Journal on Computing*., vol. 3, no. 4, pp. 307-316, 1991.

[49] M. GEN and R. CHENG, "EVOLUTIONARY NETWORK DESIGN: HYBRID GENETIC ALGORITHMS APPROACH," *Int. J. Comput. Intell. Appl.*, vol. 3, no. 4, pp. 357–380, Dec. 2003.

[50] I. Steinzen, V. Gintner, L. Suhl, and N. Kliewer, "A Time-Space Network Approach for the Integrated Vehicle- and Crew-Scheduling Problem with Multiple Depots," *Transp. Sci.*, vol. 44, no. 3, pp. 367–382, 2010.

[51] V. Cacchiani, F. Furini, and M. P. Kidd, "Approaches to a real-world Train Timetabling Problem in a railway node," *Omega (United Kingdom)*, vol. 58, pp. 97–110, 2016.

[52] G. B. Dantzig and P. Wolfe, "Decomposition Principle for Linear Programs Author (s): George B . Dantzig and Philip Wolfe Published by : INFORMS Stable URL : http://www.jstor.org/stable/167547 REFERENCES Linked references are available on JSTOR for this article : You may need to log," vol. 8, no. 1, pp. 101–111, 2016.

[53] W. Zhou, J. Tian, L. Xue, M. Jiang, L. Deng, and J. Qin, "Multi-periodic train timetabling using a period-type-based Lagrangian relaxation decomposition,"

Transp. Res. Part B Methodol., vol. 105, pp. 144–173, 2017.

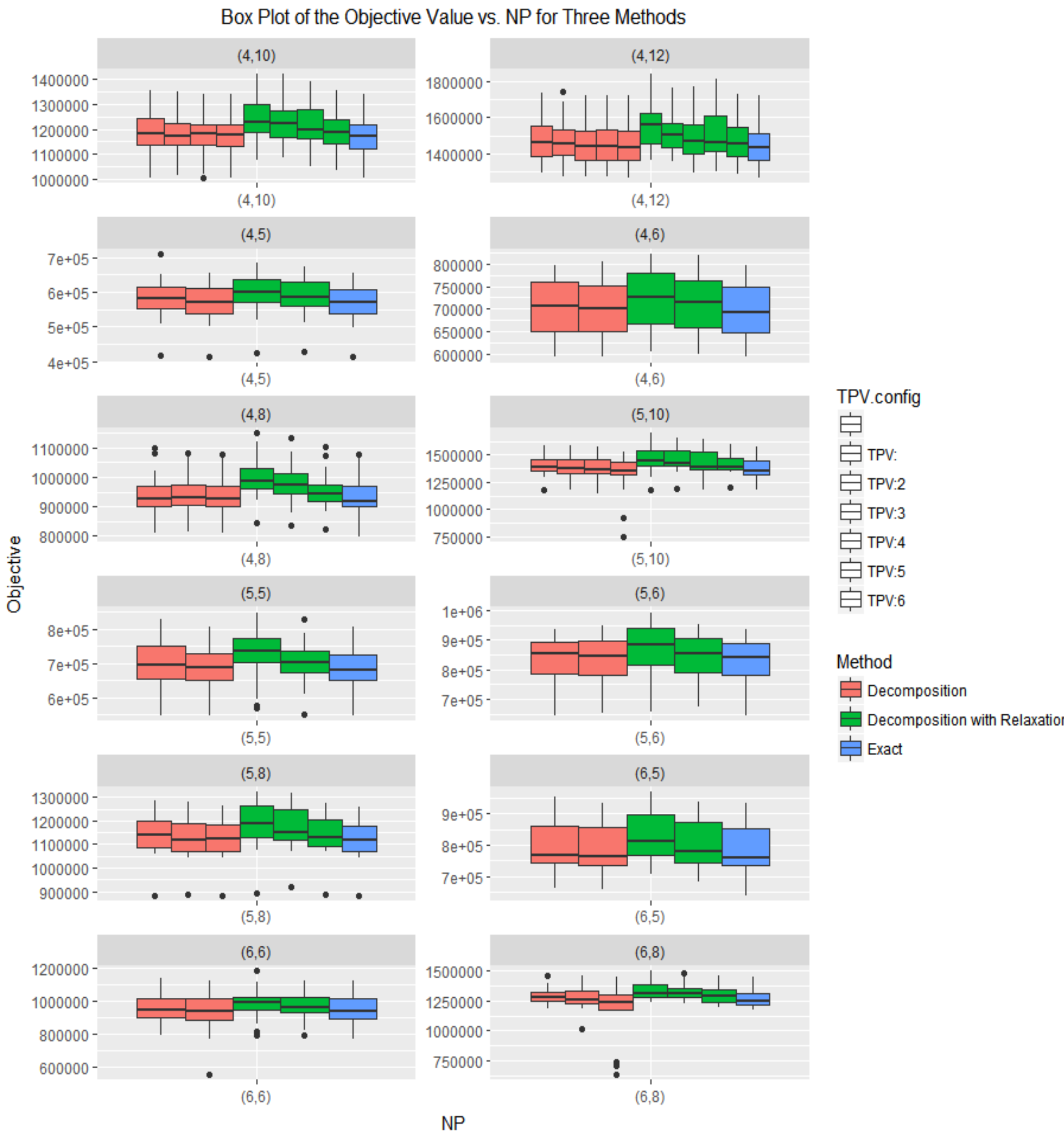
[54] E. Keyvanshokooh, S. M. Ryan, and E. Kabir, “Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated Benders decomposition,” *Eur. J. Oper. Res.*, vol. 249, no. 1, pp. 76–92, 2016.

[55] F. Trespalacios and I. E. Grossmann, “Improved Big-M reformulation for generalized disjunctive programs,” *Comput. Chem. Eng.*, vol. 76, pp. 98–103, 2015.

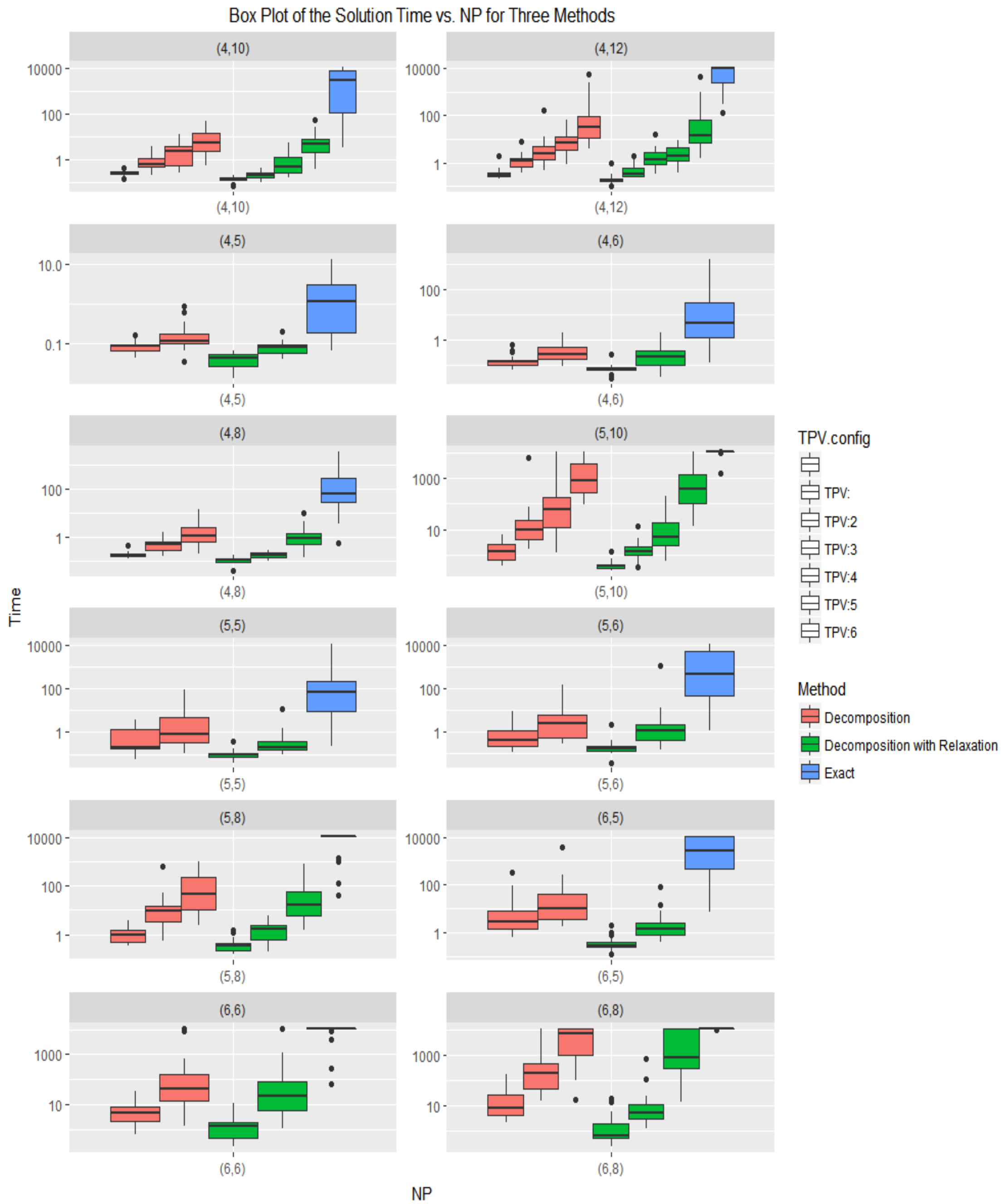
[56] “Mixed-Integer Programming (MIP) Basics | Gurobi.” [Online]. Available: http://www.gurobi.com/resources/getting-started/mip-basics. [Accessed: 03-Nov-2017].

[57] A. C. M. T. Award *et al.*, “stephen Arthur Cook 1982 ACM Turing Award Recipient,” vol. 26, no. June 1963, pp. 1–9, 2013.

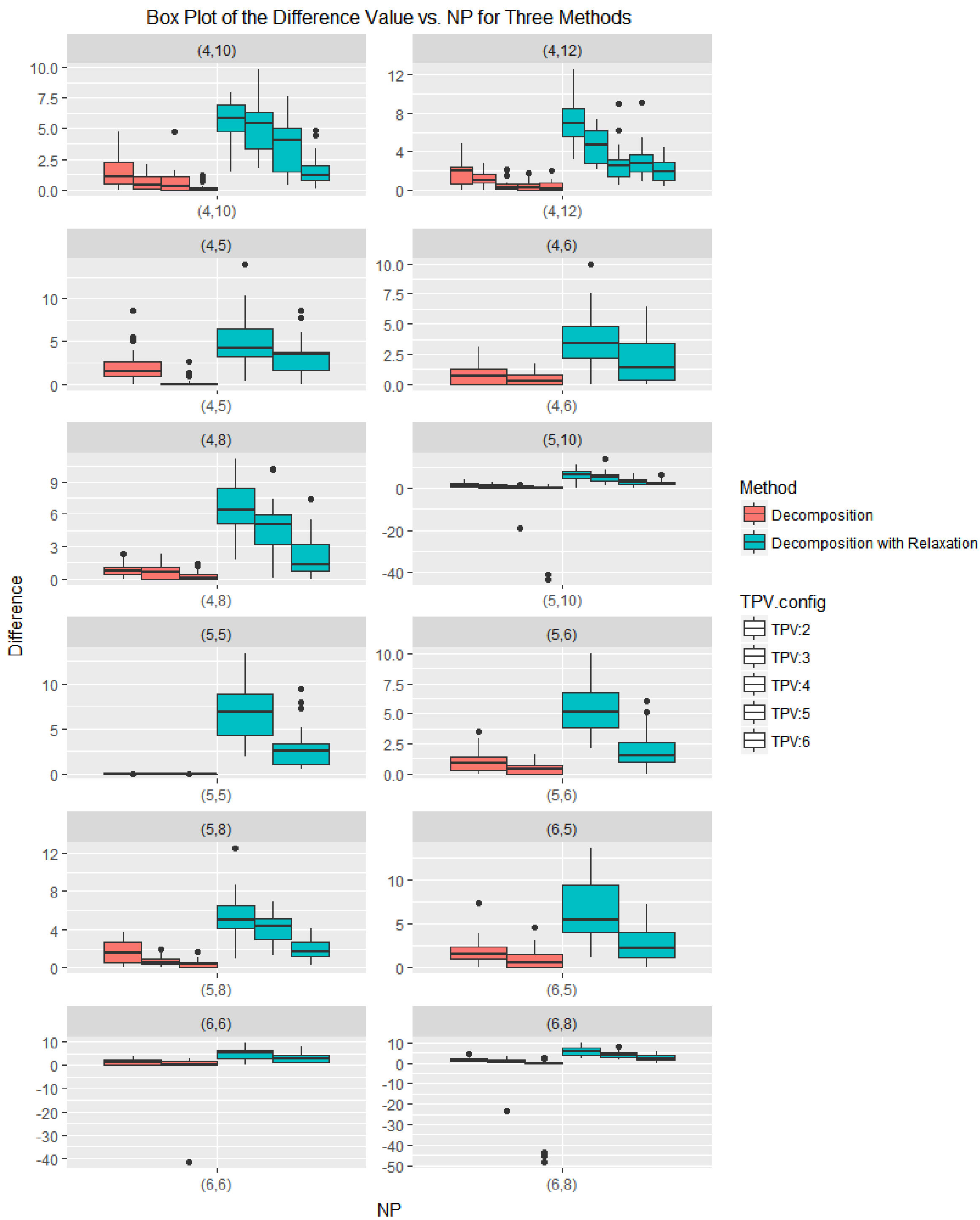
Appendix A: Box plot of the objective value vs. node-period (NP) for three different methods



Appendix B: Box plot of the solution time vs. node-period (NP) for three different methods

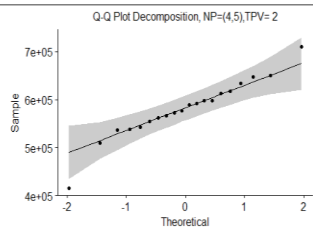


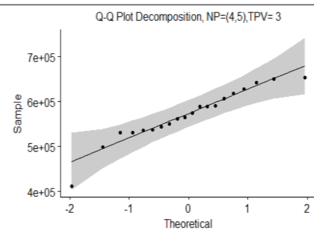
Appendix C: Box plot of difference vs. node-period (NP) for two decomposition methods



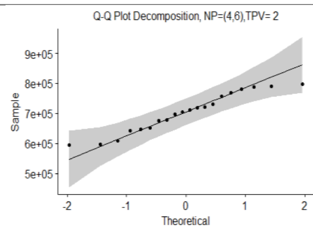
Appendix D: Normality test of obj value for different TPV in decomposition method

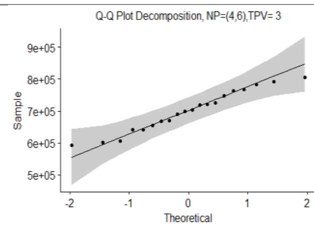
(4,5)



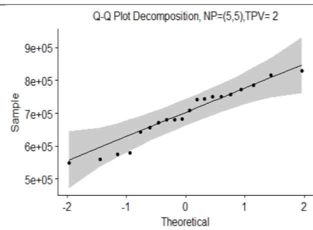


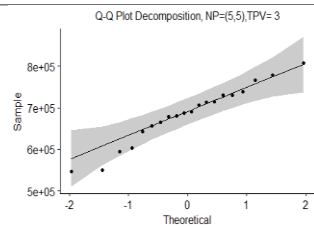
(4,6)



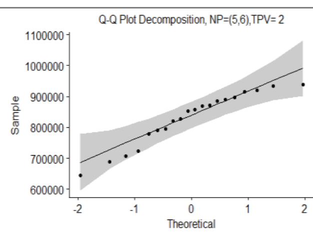


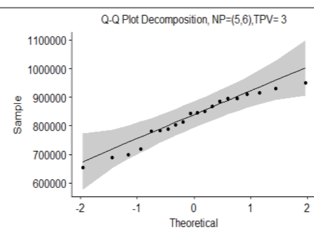
(5,5)



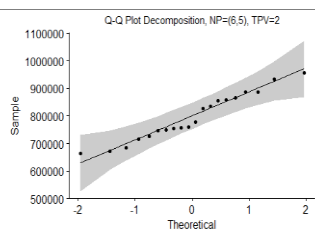


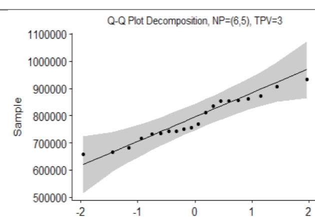
(5,6)



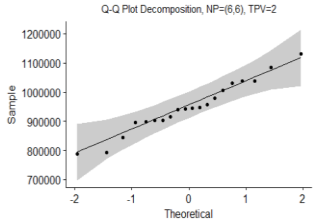


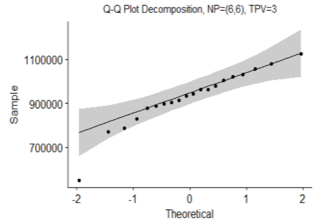
(6,5)



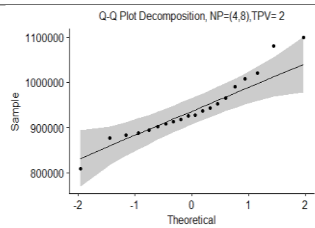


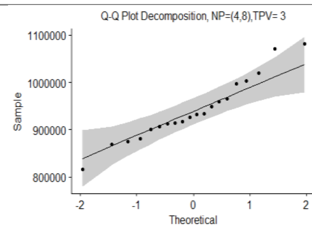
(6,6)

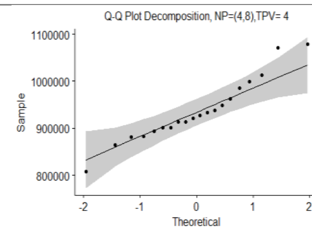




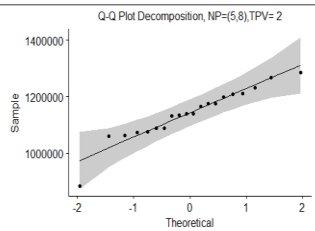
(4,8)

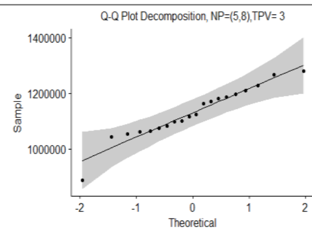


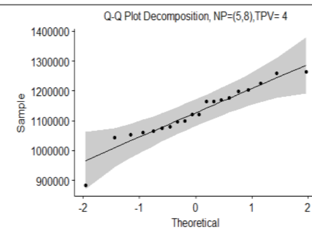




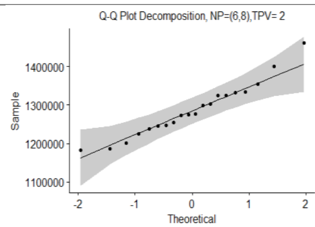
(5,8)

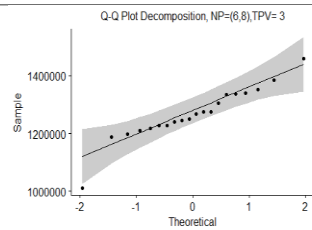


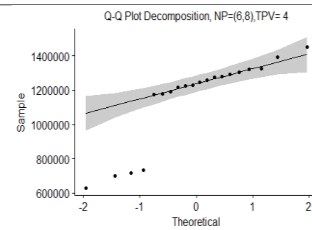




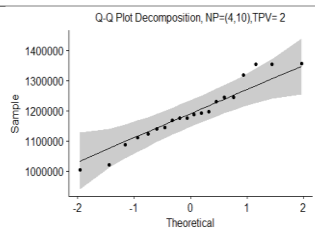
(6,8)

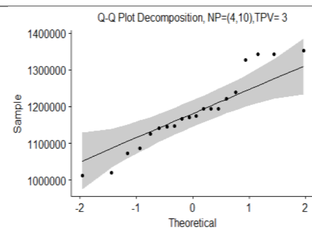


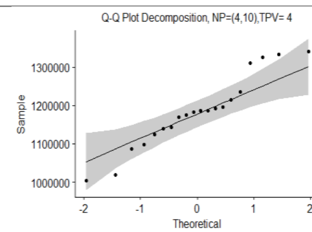


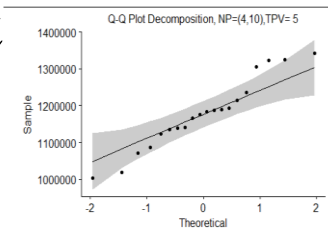


(4,10)

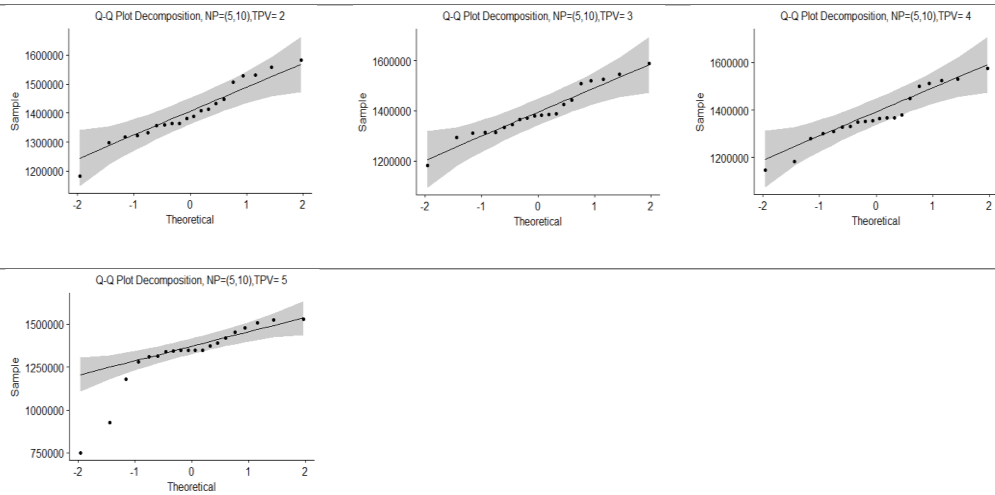






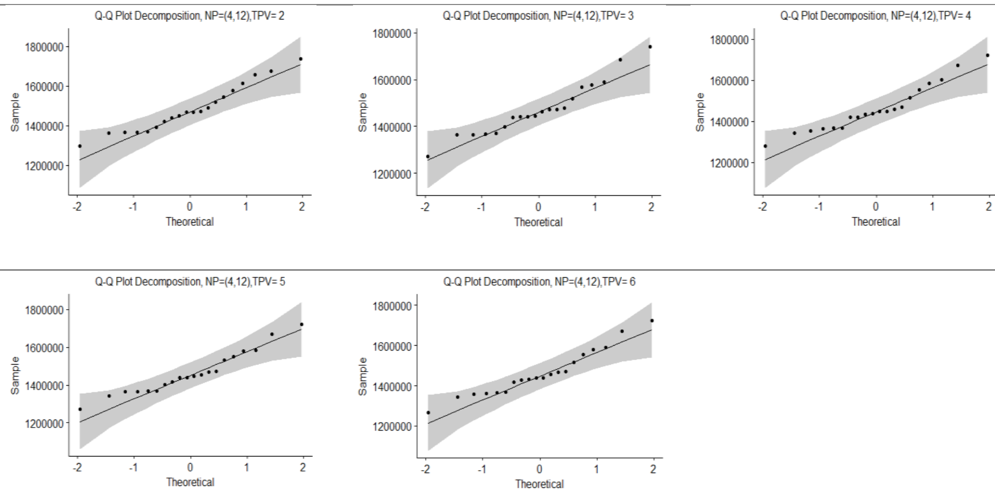


(5,10)



Q-Q Plot Decomposition, NP=(5,10),TPV= 2
Q-Q Plot Decomposition, NP=(5,10),TPV= 3
Q-Q Plot Decomposition, NP=(5,10),TPV= 4
Q-Q Plot Decomposition, NP=(5,10),TPV= 5
Sample
Theoretical

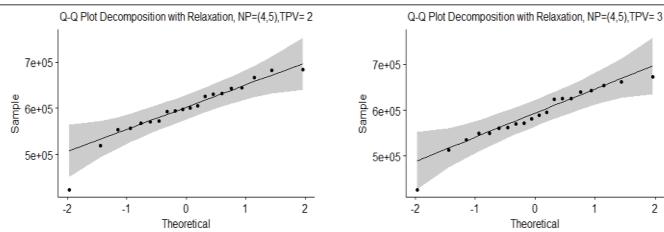
(4,12)



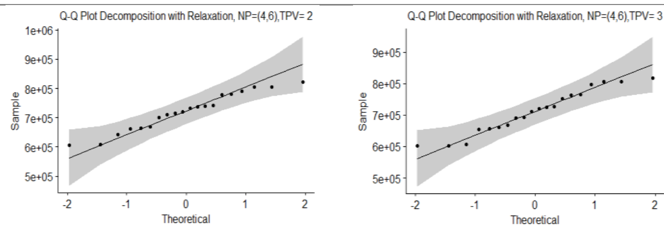
Q-Q Plot Decomposition, NP=(4,12),TPV= 2
Q-Q Plot Decomposition, NP=(4,12),TPV= 3
Q-Q Plot Decomposition, NP=(4,12),TPV= 4
Q-Q Plot Decomposition, NP=(4,12),TPV= 5
Q-Q Plot Decomposition, NP=(4,12),TPV= 6
Sample
Theoretical

Appendix E: Normality test of obj value for different TPV in decomposition method with relaxation

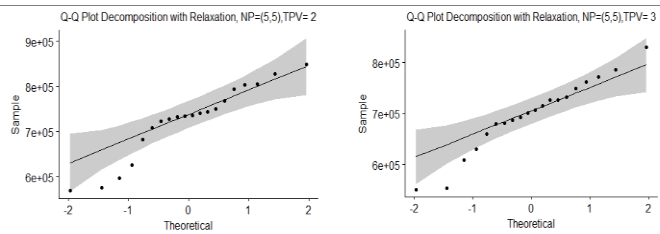
(4,5)



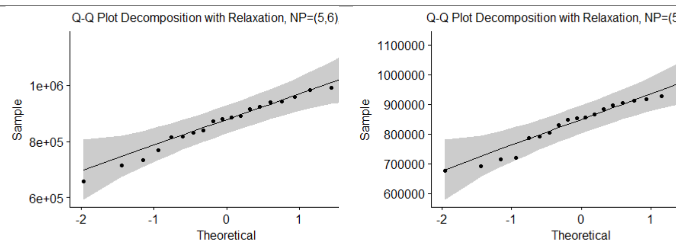
(4,6)



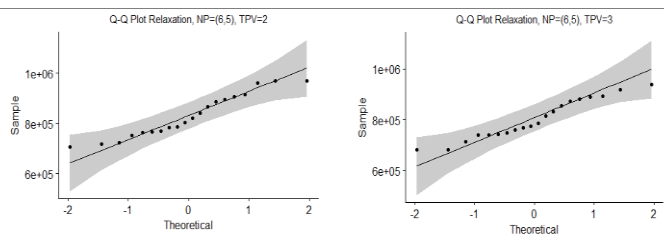
(5,5)

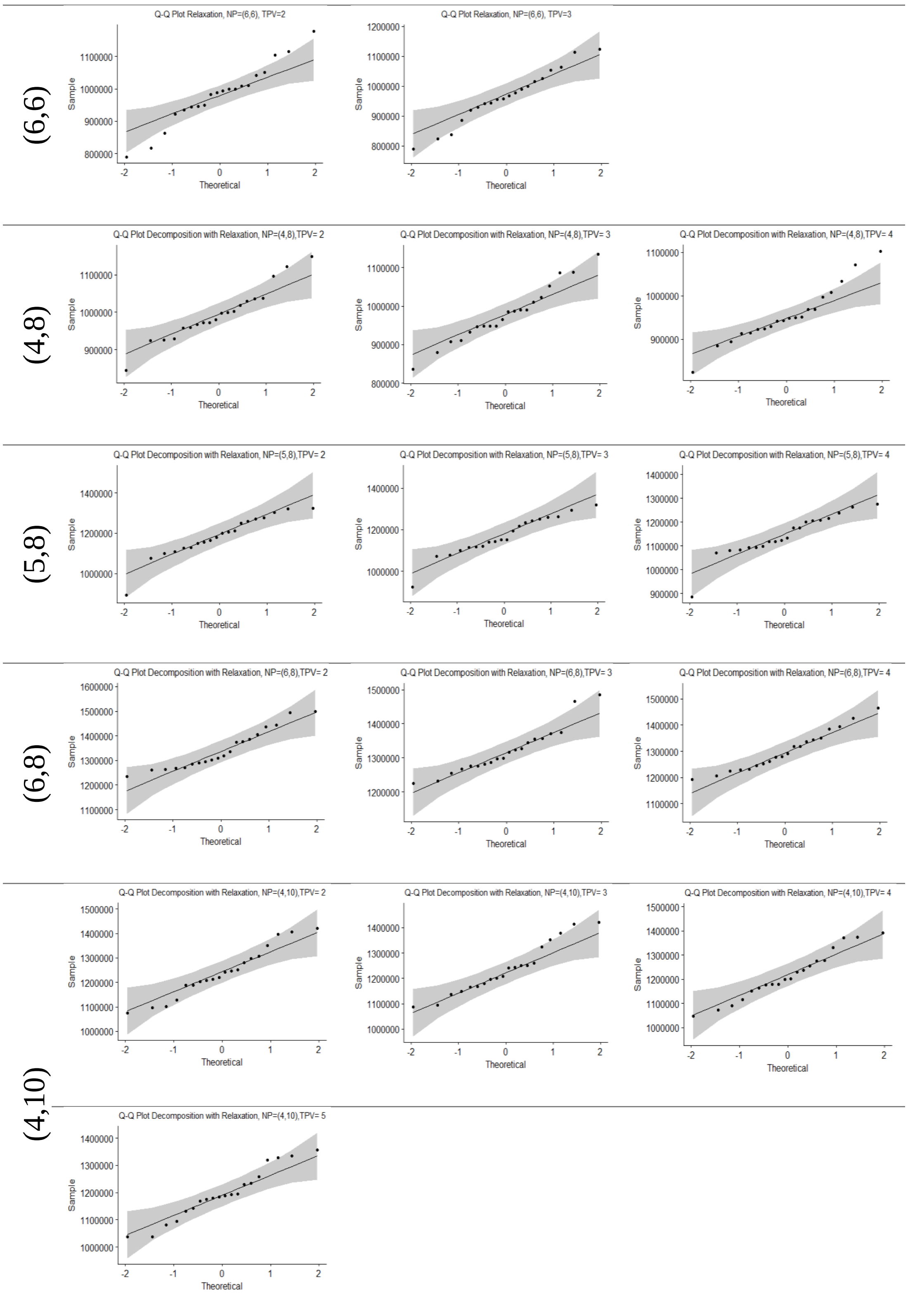


(5,6)

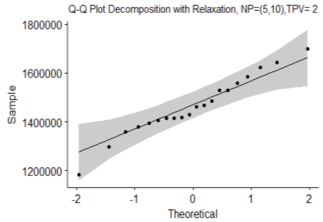


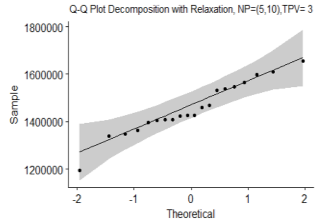
(6,5)

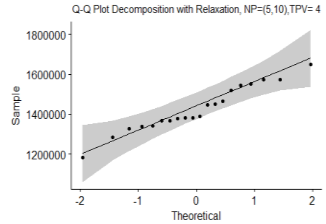


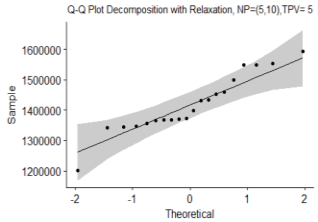

(6,6)
Q-Q Plot Relaxation, NP=(6,6), TPV=2
Q-Q Plot Relaxation, NP=(6,6), TPV=3
(4,8)
Q-Q Plot Decomposition with Relaxation, NP=(4,8),TPV= 2
Q-Q Plot Decomposition with Relaxation, NP=(4,8),TPV= 3
Q-Q Plot Decomposition with Relaxation, NP=(4,8),TPV= 4
(5,8)
Q-Q Plot Decomposition with Relaxation, NP=(5,8),TPV= 2
Q-Q Plot Decomposition with Relaxation, NP=(5,8),TPV= 3
Q-Q Plot Decomposition with Relaxation, NP=(5,8),TPV= 4
(6,8)
Q-Q Plot Decomposition with Relaxation, NP=(6,8),TPV= 2
Q-Q Plot Decomposition with Relaxation, NP=(6,8),TPV= 3
Q-Q Plot Decomposition with Relaxation, NP=(6,8),TPV= 4
(4,10)
Q-Q Plot Decomposition with Relaxation, NP=(4,10),TPV= 2
Q-Q Plot Decomposition with Relaxation, NP=(4,10),TPV= 3
Q-Q Plot Decomposition with Relaxation, NP=(4,10),TPV= 4
Q-Q Plot Decomposition with Relaxation, NP=(4,10),TPV= 5
Sample
Theoretical

(5,10)

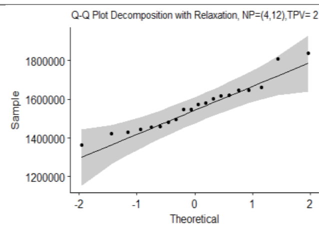

Q-Q Plot Decomposition with Relaxation, NP=(5,10),TPV= 2
Sample
Theoretical

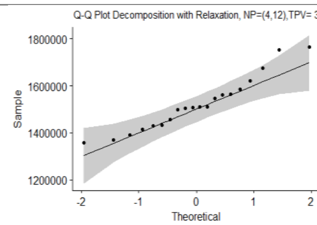

Q-Q Plot Decomposition with Relaxation, NP=(5,10),TPV= 3
Sample
Theoretical

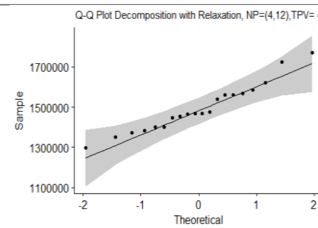

Q-Q Plot Decomposition with Relaxation, NP=(5,10),TPV= 4
Sample
Theoretical

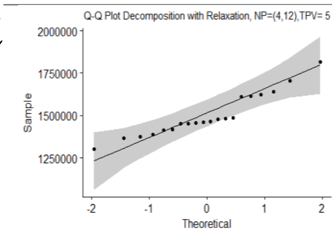

Q-Q Plot Decomposition with Relaxation, NP=(5,10),TPV= 5
Sample
Theoretical

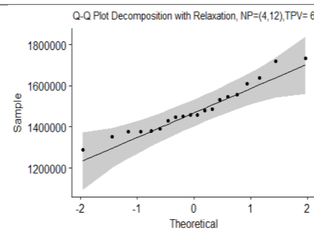
(4,12)


Q-Q Plot Decomposition with Relaxation, NP=(4,12),TPV= 2
Sample
Theoretical


Q-Q Plot Decomposition with Relaxation, NP=(4,12),TPV= 3
Sample
Theoretical

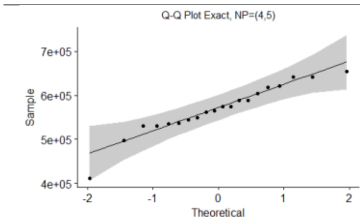

Q-Q Plot Decomposition with Relaxation, NP=(4,12),TPV= 4
Sample
Theoretical


Q-Q Plot Decomposition with Relaxation, NP=(4,12),TPV= 5
Sample
Theoretical

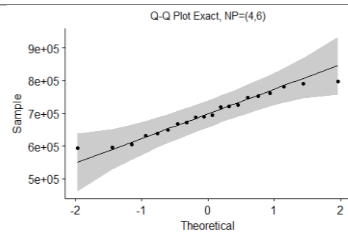

Q-Q Plot Decomposition with Relaxation, NP=(4,12),TPV= 6
Sample
Theoretical

Appendix F: Normality test of objective values for exact method

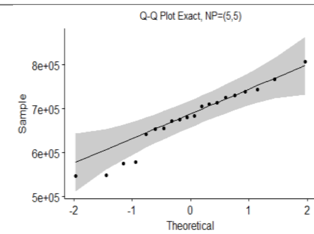
(4,5)



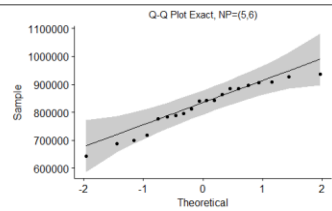
(4,6)



(5,5)



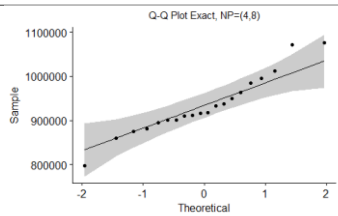
(5,6)



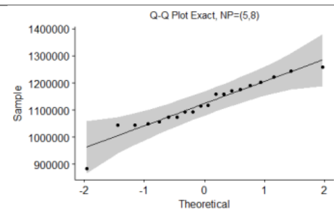
(6,5)

(6,6)

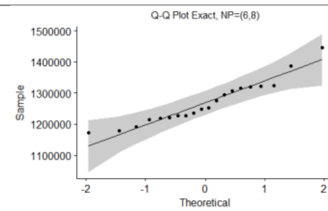
(4,8)



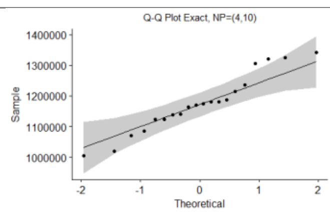
(5,8)



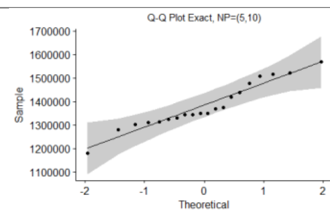
(6,8)



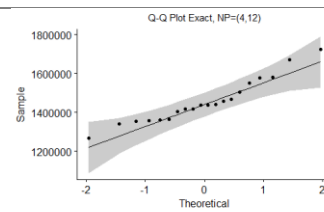
(4,10)



(5,10)



(4,12)



Appendix G: Mean objective value of each TPV in each TS network problem for both decomposition methods

No.	Node-Time Period	Decomposition Method	Decomposition with Relaxation Method
1	(4,5)	TPV.config count mean sd 1 TPV:2 20 581173.6 61103.97 2 TPV:3 20 570434.3 57879.50	TPV.config count mean sd 1 TPV:2 20 598072.6 60201.01 2 TPV:3 20 587475.2 59132.61
2	(4,6)	TPV.config count mean sd 1 TPV:2 20 702869.9 65477.17 2 TPV:3 20 699712.7 64603.36	TPV.config count mean sd 1 TPV:2 20 721756.9 64751.73 2 TPV:3 20 711017.4 69031.00
3	(5,5)	TPV.config count mean sd 1 TPV:2 20 696580.3 84113.54 2 TPV:3 20 684155.5 70580.64	TPV.config count mean sd 1 TPV:2 20 724982.1 79828.69 2 TPV:3 20 697299.1 71900.23
4	(5,6)	TPV.config count mean sd 1 TPV:2 20 830589.5 85746.31 2 TPV:3 20 825956.3 85762.63	TPV.config count mean sd 1 TPV:2 20 868013.3 95368.22 2 TPV:3 20 838620.2 85655.85
5	(6, 5)	TPV.config count mean sd 1 TPV:2 20 795255.2 86204.93 2 TPV:3 20 787410.9 81391.42	TPV.config count mean sd 1 TPV:2 20 830553.6 85708.23 2 TPV:3 20 801781.7 79001.04
6	(6, 6)	TPV.config count mean sd 1 TPV:2 20 949060.5 89171.31 2 TPV:3 20 925828.7 127850.91	TPV.config count mean sd 1 TPV:2 20 981765.7 94456.94 2 TPV:3 20 966060.3 88899.74
7	(4,8)	TPV.config count mean sd 1 TPV:2 20 942164.6 69949.26 2 TPV:3 20 941308.5 66907.76 3 TPV:4 20 936934.6 66413.73	TPV.config count mean sd 1 TPV:2 20 996073.5 71372.61 2 TPV:3 20 978497.5 73226.52 3 TPV:4 20 954174.7 64410.61
8	(5,8)	TPV.config count mean sd 1 TPV:2 20 1139825 90048.96 2 TPV:3 20 1129328 91723.95 3 TPV:4 20 1126088 88892.36	TPV.config count mean sd 1 TPV:2 20 1184885 102214.45 2 TPV:3 20 1168732 94264.28 3 TPV:4 20 1142229 87930.02
9	(6,8)	TPV.config count mean sd 1 TPV:2 20 1286807 70817.64 2 TPV:3 20 1267312 92460.93 3 TPV:4 20 1156794 246907.67	TPV.config count mean sd 1 TPV:2 20 1342369 79692.00 2 TPV:3 20 1320222 68480.49 3 TPV:4 20 1301452 75757.09
10	(4,10)	TPV.config count mean sd 1 TPV:2 20 1192023 101384.93 2 TPV:3 20 1183154 100346.40 3 TPV:4 20 1183115 94861.15 4 TPV:5 20 1178011 95749.04	TPV.config count mean sd 1 TPV:2 20 1240869 101389.75 2 TPV:3 20 1236232 98568.42 3 TPV:4 20 1216768 100042.62 4 TPV:5 20 1193255 93667.24
11	(5,10)	TPV.config count mean sd 1 TPV:2 20 1403642 99540.27 2 TPV:3 20 1396267 100784.45 3 TPV:4 20 1375098 112334.24 4 TPV:5 20 1325881 190004.71	TPV.config count mean sd 1 TPV:2 20 1464484 122777.35 2 TPV:3 20 1455853 110528.99 3 TPV:4 20 1425542 115450.00 4 TPV:5 20 1418177 95677.23

12	(4,12)		TPV.config	count	mean	sd		TPV.config	count	mean	sd
		1	TPV:2	20	1483713	118684.4	1	TPV:2	20	1562234	124099.8
		2	TPV:3	20	1473317	114555.1	2	TPV:3	20	1522410	115490.3
		3	TPV:4	20	1462329	114806.0	3	TPV:4	20	1495584	120892.1
		4	TPV:5	20	1462610	113713.7	4	TPV:5	20	1500380	128026.6
		5	TPV:6	20	1462287	114469.4	5	TPV:6	20	1485128	119567.3