INCORPORATING DECISION MAKING INTO A MULTICRITERIA AGGREGATE PRODUCTION PLANNING MODEL

By

MUHAMMAD ABDULHALIM SHAIKH

Bachelor of Science
Bangladesh University of Engineering
and Technology, Dhaka, Bangladesh
1968

Master of Enginering
Hiroshima University
Hiroshima, Japan
1979

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1984

INCORPORATING DECISION MAKING INTO A

MULTICRITERIA AGGREGATE PRODUCTION

PLANNING MODEL

Thesis Aproved:

Philip M. Wolfe

Thesis Adviser

C. Patrick Koelling

M. P. Terrell

D. W. Grace

Norman N. Durham

Dean of the Graduate College

PREFACE

This research demonstrates that multicriteria aggregate production planning is basically a problem of decision-making as opposed to a problem of finding a conventional optimal solution or nondominated extreme points. The research led to the development of a linear multicriteria aggregate production planning model that incorporates decision-making concepts, mainly, discovering promising alternatives and collecting information about them. It is shown that, theoretically, an infinite number of solutions are available to the operations manager who is responsible for making decisions concerning the aggregate levels of production and work force. After investigating all promising alternatives, the manager chooses the one that is most suitable for his firm. The model together with its solution technique helps the operations manager successively generate better alternatives based on the information obtained about the alternatives in advance.

Also, a technique is developed that helps safely exclude some of the constraints and variables from the model without affecting the accuracy of the solution. The concept of eliminating constraints and variables as well as the technique developed in this research have not been

previously applied to the area of aggregate production planning.

I wish to express my gratefulness to the Creator of the Universe, without whose help this work could not be done.

I wish to express my sincere gratitude to all who assisted me in this work and during my stay at Oklahoma State University. In particular, I am indebted to Dr. Philip M. Wolfe, the Chairman of my Ph. D. Committee for scholarly guidance, invaluable help, and concern for my family.

I also would like to acknowledge the encouragement and help given by Dr. Marvin P. Terrell.

I extend my thanks to Dr. Charles P. Koelling and Dr. Donald W. Grace who served as members of my committee.

Special thanks is given to Dr. James A. Chisman, Professor of Industrial Engineering, who helped provide access to the computer facilities at Clemson University.

I also want to thank Mr. Todd Hartshorn, a graduate student at Clemson University, who helped me in writing this thesis by using SCRIPTI, and Mrs. Cairen Withington for her help in correcting the manuscript.

My parents, particularly my mother, my wife Nahar, my daughter Munni, my son Maasum, and other relatives deserve my deepest appreciation. They have been away from me for a long time and have been very patient and understanding.

TABLE OF CONTENTS

Chapter		Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

General

Aggregate production planning has received a great deal of attention over the last three decades. The problem involved in aggregate production planning may be thought of as that of determining the work force required to produce an aggregate number of units in each period over a specified planning horizon when the demand forecasts for these periods are given.

Considerable work has been done in this area; the work can be broadly classified into two classes: single criterion aggregate production planning and multicriteria aggregate production planning. For the sake of convenience, from now on, multicriteria aggregate production planning will be termed MCAPP. In most cases of single criterion aggregate production planning, the problem is formulated to minimize the total cost over a specified planning horizon. In some cases, however, the problem considered is of maximizing the total profit over the specified planning horizon. In MCAPP more than one criterion are considered in the same problem. These criteria are either to be maximized or minimized depending on the nature of the criteria. For example, if total idle time is considered to be a criterion, it is to be minimized. If, however, the total profit over the specified horizon is considered to be one of the criteria, it needs to be maximized. In MCAPP the problem is to determine the level of the work force and the production quantity together with the available options of production smoothing, such as regular time, overtime, subcontracting, etc., at various periods, so that the specified levels of different criteria are obtained. Since in MCAPP problems, as in other types of multiple criteria decision-making problems, the optimum values for different objective functions are obtained at different points, one objective cannot be improved without sacrificing one or more of the remaining criteria. So, in this case, the problem is one of satisfying the different criteria rather than optimizing a single criterion.

The attainment of various criteria is dependent on the conditions prevailing in the respective production centers. In real world situations, the criteria as well as conditions mentioned above are very complex and difficult to express and solve mathematically. Researchers, however, have simplified the problem, and expressed the conditions as linear constraints and the criteria as linear or nonlinear objective functions.

Another aspect of aggregate production planning is that it may be associated with short cycle products or long cycle products. Short cycle products are those whose production is started and completed in the same period; whereas long cycle products require more than one period for their manufacturing to be completed. The nature of the problems in these two cases is different.

Statement of the Problem

Aggregate production planning models have been developed both in the areas of single criterion and multiple criteria. A survey [24] in the area of aggregate production planning suggests that the decision makers (the operations managers) are interested in achieving many different criteria up to a satisfactory level instead of being inclined towards optimizing one particular criterion. So far, three models [11,13,18] have been developed for handling the MCAPP problem, the nature of the models being more or less similar. But a wide gap exists between what these models suggest and what the decision makers actually do. This is also true for single criterion models. The major shortcomings of these MCAPP models are so similar in nature that they are discussed in a general framework as follows.

Although these models appear to be multiobjective in structure, they lack the essence of a multiobjective problem solution procedure. While solving a multicriteria problem, the already existing alternatives are to investigated; the possibility of creation of completely new alternatives is to be considered; and all these alternatives are to be evaluated in terms of their capability of offering the desired level of different criteria. In other words, the problem of MCAPP, like other multicriteria problems, needs to be solved using a comprehensive decision-making process. This is only possible if the decision maker is exposed to the complete spectrum of the system. But the existing MCAPP models (single criterion models as well) reveal that the earlier researchers did not recognize the requirement of incorporating the concepts of decision-making into building the models and into specifying the solution techniques for solving these models. They apparently considered this as a "one-shot" problem in the sense that the entire mission of reaching a decision is ended when a model is formulated, a solution technique is specified, and "a solution" is obtained from the model.

The existing models do not incorporate all the major requirements of the operational aspects of the firms under consideration. One of these requirements is work force stability. The model builders failed to recognize the fact that the operations managers do not consider the option of frequent hiring and firing of any number (or of the amount dictated by the solution of the model) in any period as feasible production smoothing options to the extent that the models do. Other requirements include the situations prevailing in industries, such as how many periods of overtime elapse before the operations managers consider hiring as a feasible option. The existing models, of course, incorporate the goal of maintaining a certain level of the work force. But these models do not guarantee that the operations managers will be able to maintain the level suggested in the solution because of pressure from the labor unions.

The existing models were developed within the framework of either goal programming or multiobjective programming. Their structures, by nature, dictate that these models have to be solved by specific multiobjective programming techniques, and the solution techniques employed are either goal programming or multiobjective linear programming. When the goal programming technique is used, only one solution point is obtained and most of the goals are not likely to be satisfied, although these goals are explicitly included in the model. In the case of multiobjective linear programming, only a few selected points, called nondominated extreme points, in the solution space formed by the constraints of the problem are investigated. None of the solutions corresponding to the above mentioned selected points may be acceptable to the decision maker because of the lack of attainment of the prespecified goal levels. It is, however, possible that the points near the vicinity of those selected points, or some other points in the existing or created solution space, might give results which are more favored by the decision maker. The existing models do not provide the opportunity to find and analyze these points.

Another problem is that, other than the solution(s) the decision maker gets from these models, the decision maker remains completely ignorant of his system. A model should be capable of providing information with respect to a 'cause and effect' relationship between the system variables and the system responses which, in this case, are the criteria under consideration. The existing models do not have this characteristic. In other words, the models do not help the decision maker to understand the impact on a particular criterion if one of the decision variables is changed from its present value to some other value, provided that such a change seems preferable. Finally, those models which investigate the selected points mentioned above require a lot of computational time for the identification of those points.

Because of the shortcomings of these models, they have become subjects of theoretical interest only and their practical uses are not reported in the literature. The operations managers, not finding any suitable model to apply to their production centers, decide on various options of production smoothing based on their experience. Consequently, further exploration in this area is needed to fill the present vacuum.

Objectives and Limitations of the Research

The objective of this research is to develop an MCAPP model that can be applied to practical problems and that will help the operations managers overcome the problems described above. It is important that the cost effectiveness of a model be considered. As described earlier, the aggregate production planning problems can be very complex because of the nature of the constraints and the objective function(s).

This research led to the development of a new MCAPP model for short cycle products with linear constraints and linear objective functions. This new model will, in turn, lead to the development of a model that can be used to solve a more general and complex MCAPP problem involving a general cost structure. Another objective of this research is to investigate the feasibility of developing a technique that will allow the decision maker to eliminate some of the constraints and variables from the model without sacrificing the quality of the solutions obtained.

If a technique of the sort described above can be developed, the decision-making process involved with the MCAPP problem would be more effective as well as attractive because much of the computational time would be saved due to the reduced problem size. The reader may recall that multi-criteria decision-making problems, in general, require the same problem to be run several times before a decision can be reached.

Summary of Results

The objectives of this research have been met. An MCAPP model for short cycle products has been developed incorporating the concepts of the decision-making process. Also, the features described earlier have been successfully incorporated. The new model is capable of providing the decision maker an insight into the system. It is highly flexible and can easily incorporate the requirements of the operations manager. Also, it utilizes a goal programming technique that is very simple and is easy to understand.

The results obtained from the new model have been compared with those of the Orrbeck model [22], and the Khoshnevis model [17]. The reasons for selecting these two models as the basis of comparison will be described in Chapter VIII. The comparison of the results indicate that the new model is valid. The objective of incorporating decision making concepts, that is, generation of new (promising) alternatives based on the currently available information has also been accomplished. The results show that a little adjustment of a criterion can generate an alternative offering a better solution.

The objective of developing a new technique permitting the decision maker to reduce the number of constraints and variables has also been accomplished. The results of this investigation demonstrate that it is possible to develop such a technique. It has been found that, depending on the initial size of the problem, the savings in the computation time can be as high as 75% while the accuracy of the results can be as high as 100% for the same problem. The investigation, however, is not complete. This technique is new in the area of aggregate production planning. On the basis of the results obtained by using this technique and the experience gathered by the author during the development phase of this technique, it may be said that more than 85% of the work in this area (development of the new technique) has been completed. However, we need to explore more in this area.

Finally, it has been shown that the decision-making concepts developed in this research to solve MCAPP problems can be applied to a general class of linear systems.

Contributions

The major contribution of this research is the development of an MCAPP model that has the following new features:

1. Incorporates decision-making concepts; direct involvement of the decision maker during the solution stage of the model. The decision maker will not be restricted to make decisions on the basis of the reports prepared by the analyst.
2. Permits the decision maker to become completely informed of his system and make decisions on the basis of the information gathered. This feature allows the decision maker to perform experiments by varying the system variables and to observe the resulting change(s) in the system performance. An attempt to incorporate this capability into a model is fundamental to the concept of model building.
3. Provides greater flexibility towards stabilizing the work force level. This is the direct

consequence of the second feature. The decision maker knows, in advance, what might happen if an attempt is made to stabilize the work force.

4. Utilizes a solution technique that is simpler and more effective when compared to those used to solve the existing models.
5. Utilizes the concept of an expert system. A completely new technique is developed for building an aggregate production planning model. The technique allows the formulation of a model with a reduced number of constraints and variables. The impact of this finding is very great because it will allow the inclusion of more relevant constraints and/or variables in the model thereby making the model more realistic. If it is found that no additional constraints and/or variables are needed, the CPU time that is saved can be utilized to find better solutions.
6. Permits the direct involvement of the decision maker in model building; consequently, the concept of dynamic model building can be employed. This feature is a direct result of the previous feature which allows the decision maker to incorporate the situations prevailing in the firm.
7. Develops the concepts and an approach which can be employed to solve a class of linear multicriteria decision making problems.

These results will provide managers with a welcomed decision support aid.

CHAPTER II

AGGREGATE PRODUCTION PLANNING—
AN OVERVIEW

Introduction

In this chapter the nature of the aggregate production planning problem will be briefly described and the earlier work in this area will be reviewed. Also a summary report will be given of the results of a survey on aggregate production planning problems conducted by Shearon [24]. The survey results will aid in realizing the drawbacks of the existing aggregate production planning models.

Aggregate Production Planning Problem

The aggregate production planning problem may be described as one of determining how the firm will respond to fluctuating demand situations on its productive system; specifically, it is the problem of determining aggregate levels of production, inventory, and work force at different periods of the planning horizon. The alternatives that can handle this situation are:

- -- Change in the size of the work force by hiring and firing in response to demand fluctuations.
- -- Change in the production rate by working overtime and undertime, keeping the work force level

constant.

-- Absorption of demand fluctuation through changes in the level of inventory, backlog of orders, or lost sales.

-- Use of subcontracting.

-- Combination of the above mentioned alternatives.

Cost is associated with each of the alternatives. The cost components are:

-- Regular payroll, hiring, and firing.

-- Overtime cost.

-- Inventory carrying cost, lost sales, or backlogging cost.

-- Cost due to subcontracting.

Of the alternatives described above, the last one has been found to be the most effective and is widely employed in practice.

Single Criterion Aggregate Production Planning Models

Quite a number of aggregate production planning models for solving the single criterion problem have been suggested in the literature. Depending on the nature of the methods used to solve the problem, these models can be classified into one of the following groups:

-- Mathematically optimum decision rules [2,12,15,20].

-- Heuristic decision rules [3,16,21].

-- Search decision rules [25].

Discussions about the performance and drawbacks of some of these models can be found in [10,17,19]. The application of these models, however, is limited. The reasons behind the limited applications are:

-- Assumptions are simple and far from the real situations.

-- When much of the reality is incorporated into the model, it becomes too complicated to be handled by the available mathematical tools. This is the reason that firms rely on the judgment of a manager or executive committee for the decision about the aggregate production planning problem [5].

-- The upper management in a firm is interested in more than one criteria rather than a single criterion.

Another probable reason for not applying these single criterion models is that these models are incapable of providing the managers with an insight into their systems.

Some Real Aspects of the Aggregate Production Planning Problem

In this section a summary report of survey results on the aggregate production planning problem is given. The report describes various situations involved with this problem and the managers' usual responses with respect to these situations. We know that one characteristic of a good model is that the model should reflect the static and dynamic aspects of the system. In order to achieve this objective, the model builder needs to have a comprehensive knowledge about the system for which the model is going to be built. The survey results presented here will help in understanding the perspective of how the aggregate production planning problem is handled in production centers.

The survey was conducted by Shearon [24] and was intended to provide both quantitative and qualitative data which could form a basis for model building and analysis in the area of manpower planning. The number of the subjects (plant-level production controllers or plant managers) questioned in the survey was one hundred and the number of the respondents was forty eight. The respondents represented a broad variety of corporations from among the largest in the USA to a few small companies (some with less than two hundred employees). Shearon summarizes his findings by mentioning the following:

> To summarize the survey results, work force levels in a typical firm are planned on a three month horizon and plans are reviewed monthly for the purposes of adjustment. The decision maker usually has sales forecasts which are considered accurate to ±10%. When faced with slackening demand the alternative of work-sharing via thirty two hour work weeks would not normally be selected.
>
> If overtime is required to meet expanded demand, the firm would follow a forty-eight hour week (when required) for nine to twelve weeks before increasing the size of the work force. The expansion of the work week to fifty-six or more hours is not considered feasible by the typical firm.
>
> When the manpower is added, the normal training period before an employee reaches the standard rate is 5 to 6 weeks. Internal promotions of employees to higher job grades require a similar training period. Employees recalled from layoff status are expected to resume the standard production rate within two weeks and workers who are bumped down to a lower labor skill position when there is a general layoff also usually achieve the standard rate in two weeks or less.
>
> The typical firm does not know the cost of laying off an employee, although many indicated qualitatively that this is an expensive item. Of the thirteen firms who supplied a quantitative response, the majority placed their costs within the five hundred to one thousand dollar bracket. The data suggest that the decision makers consider work force reductions to be very expensive and

seek to avoid them by alternative feasible actions.

Conclusions:

A number of important findings have been drawn from the survey reported in this chapter. The more significant findings were:

A. Plant managers and their staffs make manpower planning decisions. These managers are responsible for operational decisions but not for marketing decisions, hence they must solve problems arising from demand fluctuations without the ability to influence demand through marketing efforts.

B. A majority of firms attempt to maintain a constant size work force whether their demand pattern is uniform, seasonal, cyclical, or constant growth. In more than half of the union agreements the firms were committed to pay supplemental benefits to employees who were layed off. With union pressure to maintain steady employment increasing the economic consequences of altering work force levels, managers will find their options to respond to demand variations increasingly restricted and expensive.

C. The survey provided significant results concerning the ranking of decision alternatives. If demand increases beyond normal capacity, the most likely management reaction is to work overtime for two or three months before adding to the work force. The tenuous nature of demand increases and the pressure to meet customer demands make this policy the most feasible approach. Managers' initial responses to declining demand is to build inventory and reduce the order backlog. These passive actions avoid union/management conflicts but increase the exposure to risks of obsolete inventory or an inadequate backlog to support efficient production.

D. The results concerning the criteria for evaluating operational management provided insight into the factors influencing the choices of management actions

> discussed in the preceding paragraphs. The most important criterion is the ability to meet customer demand schedules. The logical response in an increasing demand situation is to work overtime even though per unit costs are increased. When the situation reverses and demand falls below capacity, the most important criteria of meeting schedules and controlling direct costs are all met satisfactorily while building inventory and reducing the backlog. The relatively low perceived importance of inventory turnover makes this policy the most reasonable for a manager. A particular point to re-emphasize here is that operations managers are evaluated on multiple criteria and the formal structuring of a general decision framework is complicated by the lack of a single measure of performance. (pp. 8 - 11)

Also at one point of the survey report Shearon points out that

> . . . The majority of responses (about three fourths) indicated that potential management union conflicts would not dissuade management from an economic course. (p. 8)

Shearon's survey offers much information about the situations existing in industries with respect to aggregate production planning. The report is long and only the major features are directly quoted here. Anyone who wants to construct a model for the aggregate production planning problem should read the report in order to obtain a good understanding of the problem.

CHAPTER III

MULTICRITERIA DECISION-MAKING FUNDAMENTALS

Introduction

In this chapter some of the concepts of decision-making will be described. MCAPP is essentially a problem of decision-making, and as such we need to have a knowledge of these concepts. The materials presented here can be found in related books. However, placing them here will be appropriate for recognizing the drawbacks of the existing MCAPP models and for developing the basis for the proposed model.

Terminologies

In this section a few of the terms commonly used in multicriteria decision making will be described. For the sake of convenience, from now on, the word 'solution (point)' will be used interchangeably with the word 'alternative.'

Alternatives

Alternatives are the mutually exclusive sets of means engaged towards achieving the stated objectives and prespecified goals or targets [26]. Alternatives are essentially goal-feasible strategies.

Ideal Alternative

When all of the objective functions attain their optimum values simultaneously, this is an ideal alternative. Depending on the situation, the ideal alternative may be feasible, as in Figure 1, or it may be infeasible, as in Figure 2. It may not exist at all, as shown in Figure 3. It is possible that the location of the ideal point will change if some feasible alternatives (solution space) and/or some criteria are added or deleted. In Figure 4, this possibility has been explained by adding the shaded area to the solution space that was shown in Figure 2. As a result of inclusion of the new solution space the ideal point has moved from the location I to I'. This concept of a displaced ideal, as will be noted later, is very important in the decision-making process, because the selection of an alternative is largely dependent on how close the alternative is to the ideal alternative with respect to certain preferred criteria.

Nondominated Solution

The details about the nondominated solutions can be found in [26]. For our purpose, it is sufficient to understand that if someone moves away from a nondominated point, one of the objective functions cannot be improved without worsening one or more of the rest of the objective functions. In Figure 2, all of the points on the line AB, including points A and B are nondominated solutions.

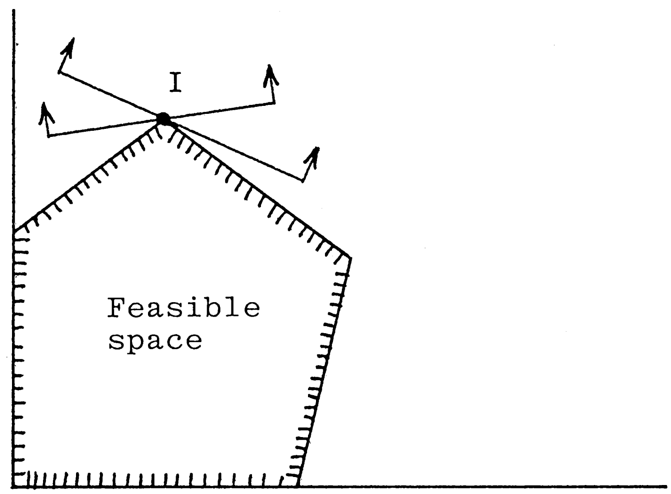


Figure 1. Feasibility of the
Ideal Solution

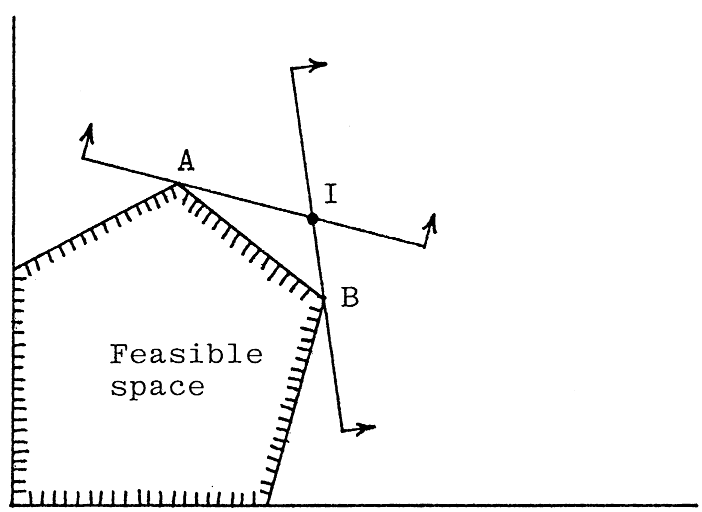


Figure 2. Infeasibility of the
Ideal Solution

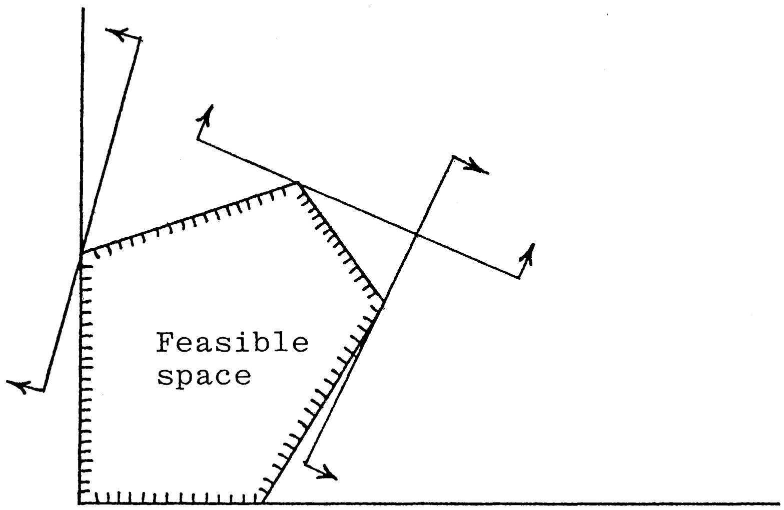


Figure 3. Nonexistence of the
Ideal Solution

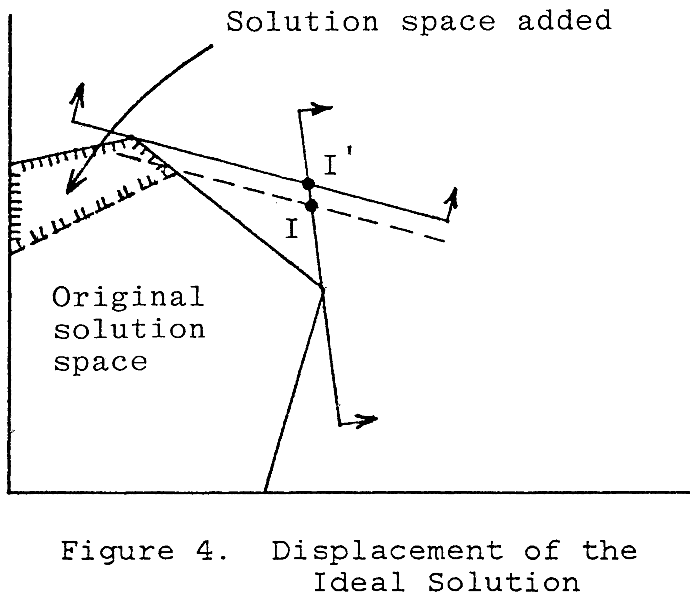


Figure 4. Displacement of the
Ideal Solution

Whereas, in Figure 3 all feasible solutions are nondominated.

Nondominated Extreme Points

The extreme points in the set of nondominated solutions are termed nondominated extreme points. In Figure 2, A and B are nondominated extreme points; whereas, in Figure 3 all extreme points are nondominated extreme points.

It may be noted that a nondominated extreme point does not have to be a point where an objective function attains the optimum value. If at a particular extreme point one or more of the objective functions attains its optimum value, there is no need to investigate this point for its nondominance, because this extreme point is nondominated by nature unless alternate optimal solutions exist. Extreme points other than this type need to be checked for nondominance. But in order to check the nondominant characteristic of each extreme point, we have to construct and solve a separate linear programming problem which consists of the original set of system constraints and some other new constraints. Consequently, those techniques which look for nondominated extreme points to solve multicriteria decision-making problems require a lot of computer time. This is particularly true when the system is large.

Ranking of Importance of Attributes

Ranking of importance of attributes may be preemptive or additive. Preemptive ranking means that only the highest ranked attribute is considered and all the remaining ones are excluded from the analysis; that is, they are assumed not to be important at all. After the highest ranked attribute has been fully analyzed, the analysis is continued with the next highest attribute, and so on. In the case of additive ranking, the weights attached to each attribute have simultaneous effects on the analysis. The weights can be summed and normalized; that is, each weight can be divided by the sum of the weights so that they add to unity.

Decision-Making

The problem of decision-making arises only when someone (the decision maker) has to decide something on the basis of multiple attributes, objectives, criteria, functions, etc. There is no question of decision-making when the decision maker has to decide something on the basis of a single criterion. Decision-making is a dynamic process and consists of three interdependent stages [26]. These stages are predecision, partial decision, and postdecision. Each decision stage itself is composed of a series of partial decisions characterized by their own pre- and postdecision stages.

The Predecision Stage

At this stage, there is a 'sense of conflict' because of unavailability of suitable alternatives and particularly because of infeasibility of the ideal alternative.

Encountering this sense of conflict, the decision maker starts searching new alternatives, gathers information about them, and evaluates them. In the beginning, the information gathering and evaluation process is highly objective and impartial. Also, divergence in the attribute scores of attractiveness is sought, because closeness in the attribute scores of attractiveness makes the decision-making process complicated. However, with the available information, a choice among the alternatives is reached by the decision maker, and when the predecision stage is stabilized, a partial decision is made. Note that accurate collection of information about all possible alternatives is the key to a sound decision-making process.

Partial Decision

At the partial decision phase of the decision-making process, there is a directional adjustment of the decision situation. Such adjustment may consist of the following:

-- Discarding the alternatives that at the moment appear obviously inferior.

-- Reconsidering previously rejected alternatives.

-- Adding or deleting criteria.

It has been found earlier that because of these three actions the ideal is likely to be displaced. In this case, this displacement is expected to be towards the feasible space; because inferior alternatives have been discarded. But since the ideal is still infeasible, the conflict remains.

Another result of a partial decision is cognitive dissonance. There is a tendency to justify the partial decision just made, that is, to reduce the resulting dissonance. At this point the decision maker starts to reevaluate subjectively the attributes under consideration in such a way that the attractiveness of the discarded alternatives is diminished and that of the retained alternatives is increased. But since conflict remains, the decision maker enters a new predecision stage with the current locations of the ideal point and the feasible set of alternatives.

The Final Decision Stage

The predecision and the partial decision stages are continued repeatedly until the decision maker is left with only a few alternatives. During this time, the decision-making process becomes very complicated because the attractiveness of the alternatives converges. The conflict is fully resolved by moving the ideal alternative towards the preferred alternative which was found previously by going through several predecision and partial decision situations. At this stage the magnitude of the post decision dissonance is at its highest level.

The Postdecision Stage

The postdecision stage of the decision-making process is extremely important for an understanding of the decision implementation process. At this stage the decision maker becomes completely biased towards the chosen alternative. He continues to seek new information in favor of the chosen alternative in order to increase his confidence about the decision taken and to reduce postdecision regret and dissonance.

Remarks

It is obvious from what has been described above that decision-making is a very difficult task. It requires both quantitative and subjective evaluation of the alternatives that help achieve different criteria. Not only is information about the currently available alternatives required; but the search for new alternatives is equally important. The new alternatives do not have to be completely different from the existing ones. In real world situations, there are many examples where a slight modification of the existing situation improved the system. This is specially true with the MCAPP problem which involves several options to cope with the fluctuations in demand at different periods. Theoretically, an infinite number of alternatives are available to the operations manager. He needs to choose the one most satisfactory for his firm on the basis of complete information about all of those alternatives. The way the operations manager can get and utilize this information will be described in Chapter VI.

CHAPTER IV

MULTICRITERIA PROBLEM SOLVING TECHNIQUES—AN OVERVIEW

Introduction

In this chapter, a few techniques commonly used in solving multicriteria problems will be discussed. Since these techniques are available in the related books, details will be omitted. The purpose of presenting the following material is to make the reader aware of the limitations of these techniques. This will help in understanding the drawbacks of the existing MCAPP models with respect to the solution techniques used to solve the models. However, since the available multicriteria problem solving techniques are numerous, only the basic categories will be discussed.

Goal Programming Techniques

In goal programming, linear goals and constraints can be written in general notation as follows:

$$\sum_{j=1}^{n} a_{ij}x_j + d_i^- - d_i^+ = b_i \quad i = 1, \ldots, m \tag{4.1}$$

where x_j are n decision variables, d_i^- denote negative deviations or slack variables, d_i^+ denote positive deviations or surplus variables, b_i are m goals or rigid constraining values, and a_{ij} are technological coefficients. The nature of b_i determines whether the corresponding equality is a constraint or a goal. When b_i is a goal, d_i^- and/or d_i^+ appear in the objective function(s). All x_j , d_i^-, and d_i^+ are required to be nonnegative, and the deviations d_i^- ,d_i^+ are always to be minimized.

There are three types of goal programming. The differences in these methods lie in the way the goal deviations are minimized. The three approaches of goal programming are described below.

Preemptive Goal Programming

In this type of goal programming, the objective functions $f_i(d_i^-,d_i^+)$ are minimized one by one. The function with the highest priority is considered first, the function with the next higher priority is considered next, and so on. But a function with a lower priority is not considered if the function with higher priority is deteriorated when the one with lower priority is under the process of improvement. Functions f_i are typically linear functions of d_i^- and d_i^+. The objective functions can be expressed as:

$$P_1 f_1(d_1^-,d_1^+) + \ldots + P_K f_K(d_K^-,d_K^+) \qquad (4.2)$$

where K is the number of goals, and

$$P_1 >>> P_2 >>>, \ldots, >>> P_K \qquad (4.3)$$

<u>Archimedian Goal Programming</u>

In this case the following function is minimized.

$$w_1[f_1(d_1^-, d_1^+)]^p + \ldots + w_K[f_K(d_K^-, d_K^+)]^p \qquad (4.4)$$

All the objective functions are considered simultaneously and their weights w_i are not preemptive. Powers p can take any value, but usually p = 1, 2, or ∞.

<u>Multigoal Programming</u>

In this case $[f_1(d_1^-, d_1^+), \ldots, f_K(d_K^-, d_K^+)]$ is minimized in a vector sense; that is, it (the technique) identifies all nondominated solutions with respect to the objective functions $f_i(d_i^-, d_i^+)$, as in multiobjective linear programming. It does not require specification of the criterion weights (preemptive or Archimedian), and there is no need to express the objective function in terms of an aggregate preference or a distance function.

It is worth mentioning here that each of these methods has some disadvantages. In preemptive goal programming, not all of the objective functions are likely to be satisfied. In Archimedian goal programming there is a problem in specifying the values of the weights w_i and the power p. And the task of investigating all nondominated solutions in the case of multigoal programming, as mentioned earlier, is formidable.

Multiobjective Linear Programming

The general form of the problem treated by this technique may be given as:

$$\text{Minimize } [f_1(X),\ f_2(x),\ \ldots,\ f_K(X)]$$
$$\text{subject to} \quad AX \leq b \qquad\qquad (4.5)$$
$$X \geq 0$$

where, $X = [x_1,\ x_2,\ \ldots,\ x_n]$,

A is $m \times n$ matrix,

and, $f_i(X) = c_{i1}x_1 + \ldots + c_{in}x_n \quad i = 1,\ \ldots,\ K$

Multicriteria Simplex Method (MSM)

This is a method to solve the problem stated in (4.5). The method was designed to investigate nondominated extreme points. A nondominated extreme point is found first; then search continues to find other nondominated extreme point(s) until a satisfactory point is found. This principle is used in several multiobjective linear programming tecniques.

Multiparametric Decomposition

When the objective functions $f_1,\ \ldots,\ f_K$ are linear, it is possible to combine them into a multiparametric aggregate function instead of minimizing objective functions as separate parallel entities. The aggregate function is given by

$$f(\lambda, X) = \lambda_1 f_1(X) + \ldots + \lambda_K f_K(X) \qquad (4.6)$$

where, $\lambda_1 + \lambda_2 + \ldots + \lambda_K = 1 \qquad (4.7)$

In linear cases, the nondominated set of X can be found by minimizing $f(\lambda,X)$ for all possible combinations of λ satisfying the above mentioned conditions.

When $f(\lambda,X)$ is minimized over a convex polyhedron X, each nondominated extreme point of X will be associated with a particular subset of λ's such that $f(\lambda,X)$ will reach its minimum at that point. In other words, the set of all parameters can be decomposed into subsets associated with each of the nondominated solutions. This technique offers some advantages when λ is known.

Both MSM and multiparametric decomposition techniques have the disadvantage that all nondominated extreme points have to be investigated although the most satisfactory solution might not be at any one of these extreme points. The multiparametric decomposition technique has an additional problem of specifying the components of λ. Moreover, in Chapter VI it is shown that because of the nature of the MCAPP problem it is not necessary to investigate all nondominated extreme points or nondominated solutions. This fact will also support the claim made later in Chapter VI that the use of the methods (described in this chapter) by the earlier researchers in solving the MCAPP model(s) is inappropriate.

Other variants of goal programming and multiobjective linear programming can be found in [6]. Since the basis of those variants are those described here, the problems associated with those variants are more or less similar to their origins. However, in some cases, the solution techniques are more complex and appear obscure to the decision maker.

In Chapter VI, a goal programming technique known as Method of Satisfactory Goals [6] will be described and this method will be used to solve the proposed model. Consequently, this method will be explained after the proposed model is described.

CHAPTER V

MULTICRITERIA AGGREGATE PRODUCTION PLANNING MODELS—AN OVERVIEW

Introduction

In this chapter the existing MCAPP models will be described and their drawbacks will be pointed out. At present, three MCAPP models are available in the literature [11,13,18]. Some of the drawbacks are common to all of the models; some are specific to a particular model. The author is under the impression that these drawbacks are serious, and have rendered the models unfit for practical applications. Critical analysis of the drawbacks of these models will help the reader realize to what extent these models deviate from reality. The models will be briefly described. An explanation of the major problems associated with these models will follow after the description.

Goodman Model

Goodman's goal programming approach [11] in formulating the aggregate production planning problem was the first in the area of MCAPP. Goodman's model is based on the Holt et al. model [15] which, for the sake of convenience, is given below.

$$\text{Min.} \quad \bar{C} = \sum_{T} [C_1W_t + C_2(W_t - W_{t-1})^2 + C_3(P_t - C_4W_t)^2$$

$$+C_5P_t - C_6W_t + C_7(I_t - C_8)^2] \qquad (5.1)$$

$$\text{subject to} \quad I_t = I_{t-1} + P_t - D_t, \quad t=1, \ldots, T \qquad (5.2)$$

where,

P_t = production rate in period t;

D_t = demand in period t;

W_t = work force level in period t;

I_t = inventory level at the end of period t;

T = planning horizon; and,

C's are known positive constants.

Goodman shows that this model can be transformed into a goal programming model. Goodman's goal programming approach is based upon the idea that each of the quadratic cost terms in (5.1) becomes zero when the expression inside the corresponding parenthesis becomes zero. So, the minimization of each of these quadratic cost terms can be thought of as a goal and formulated as a constraint. The resulting goal constraints can, therefore, be expressed as given below.

$$\begin{aligned} W_t - W_{t-1} + L_t^+ - L_t^- &= 0 \\ P_t - C_4W_t + M_t^+ - M_t^- &= 0 \\ I_t - C_8 + N_t^+ - N_t^- &= 0 \end{aligned} \qquad (5.3)$$

where L, M, and N are slack variables. These slack variables are then included in the objective function by assigning positive coefficients in order to get the effect of penalizing the deviations from the desired goals. The objective function then becomes:

$$\text{Minimize } \bar{\bar{C}} = \sum_{T} [(C_1-C_6)W_t+C_5P_t+C_9L_t^{+}+C_9L_t^{-}$$

$$+C_{10}M_t^{+}+C_{10}M_t^{-}+C_{11}N_t^{+}+C_{11}N_t^{-}] \qquad (5.4)$$

The coefficients C_9, C_{10}, and C_{11} are then selected in such a way as to give adequate cost approximation for the original quadratic terms they represent. Goodman describes a method for computing the coefficients. Finally, Goodman shows that the total cost obtained from his model is about 3% higher than that obtained from the Holt et al. model.

Lawrence and Burbridge Model

This is a multiobjective, multi-item, multi-plant production model. In this model, Lawrence and Burbridge [18] propose to optimize three objective functions subject to a number of constraints. The objective functions are described below.

1. Maximize the total sales revenue of the ath production location to the bth customer location:

$$\text{Maximize } Z_1 = \sum_{i=1}^{m} r_{iab}x_{iab} \qquad (5.5)$$

where,

x_{iab} = amount of the ith item to be produced at the ath production location for transportation and sale at the bth customer location.

r_{iab} = revenue per unit of the ith item produced at the ath production location for the bth customer location, and,

m = number of items.

2. Minimize the sum of the total costs of production at all plant locations and minimize the total cost of transporting the items from all plant locations to all customer locations:

$$\text{Min. } Z_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} x_{ijk} + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{\ell} t_{ijk} x_{ijk} \qquad (5.6)$$

where,

C_{ij} = total unit cost of producing the ith item at the jth location,

t_{ijk} = total unit cost of transporting the ith item from the jth production location to the kth customer location, and,

n, ℓ = total number of production and customer locations, respectively.

3. Minimize the total production of the rth item at the sth production location:

$$\text{Minimize } Z_3 = \sum_{k=1}^{\ell} x_{rsk} \qquad (5.7)$$

The constraints are related to:

-- demand of the ith item at the kth customer location;

-- available production capacity of the ith item at the jth production location;

-- maximum weight involved in shipping for all items shipped from the jth production location to all customer locations; and,

-- budget for production and distribution of all items.

Lawrence and Burbridge solve the problem with two products, two production locations, and three customer locations by using the multiobjective linear programming technique to generate 24 efficient extreme point solutions. From these, the decision maker may choose any one that is most satisfactory to his firm.

It is worth noting here that Lawrence and Burbridge solve the problem considering the planning horizon of one period only. In real situations, aggregate production planning problems are usually solved for not less than six periods. When the length of the planning horizon increases, the number of efficient extreme points also increases. The consequence is that the computation time increases considerably.

Hindelang and Hill Model

Hindelang and Hill [13] consider a multi-product and multi-departmental problem, and formulate it into a goal programming model. The goals of the model are as described below.

1. Manpower level and productivity goals (for each department k):

$$T_k^1(L_{k,t-1} - ND_{kt}) + T_k^2\, NI_{kt} + T_k^3\, O_{kt}$$

$$+ D_{ikt}^- - D_{ikt}^+ = \sum_i T_{ik}^4\, P_{ikt} \qquad (5.8)$$

where,

$$L_{k,t-1} + (NI_{kt} - ND_{kt}) = L_{kt}, \qquad (5.9)$$

L_{kt} = labor in department k in period t;

NI_{kt} = net increase in labor in department k in period t;

ND_{kt} = net decrease in labor in department k in period t;

T_k^1 = productivity coefficient for old workers;

T_k^2 = productivity coefficient for newly hired or transferred workers;

T_k^3 = productivity during overtime;

T_{ik}^4 = time required by the ith product in the kth department;

O_k^t = number of overtime hours in department k in period t; and,

P_{ikt} = number of products i to be produced in the kth department in period t.

It appears that the expression (5.8) is incorrect; because according to the way the coefficients have been defined, $T_k^2 NI_{kt}$ indicates number of products, whereas, $\sum_k T_{ik}^4 P_{ikt}$ indicates time.

2. Job rotation and labor force stability goals:

$$\Delta_{1t} + D_{3t}^{-} - D_{3t}^{+} = M_t \tag{5.10}$$

where,

$$\Delta_{1t} = \text{Min}\ (\sum_k NI_{kt}, \sum_k ND_{kt}), \tag{5.11}$$

M_t = desired number of workers rotated among departments in period t.

Also, $$\Delta_{2t} + D_{4t}^{-} - D_{4t}^{+} = Q_t \tag{5.12}$$

where, $$\Delta_{2t} = |\sum_k NI_{kt} - \sum_k ND_{kt}| \tag{5.13}$$

and,

Q_t = maximum desired fluctuation in aggregate work force level in period t.

3. Cost minimization goals:

(i) Cost related to changes in work force size:

$$C^1 \Delta_{3t} + C^2 \Delta_{4t} + C^3 \Delta_{1t} + D_{5t}^- - D_{5t}^+ = WFC_t \qquad (5.14)$$

where,

Δ_{1t} = number of workers transferred;

Δ_{3t} = number of workers hired;

Δ_{4t} = number of workers fired;

C^1, C^2, C^3 = cost coefficients; and

WFC_t = target dollar cost for transfer, hiring, and firing.

Δ_{3t} and Δ_{4t} respectively are given by

$$\Delta_{3t} = \begin{cases} \sum_k NI_{kt} - \sum_k ND_{kt}, & \text{if positive} \\ 0, & \text{otherwise;} \end{cases}$$

$$\Delta_{4t} = \begin{cases} \sum_k ND_{kt} - \sum_k NI_{kt}, & \text{if positive} \\ 0, & \text{otherwise.} \end{cases}$$

(ii) Cost related to production rate:

$$\sum_k \Big[\sum_i (C^4_{ikt} P_{ikt} + C^5_{ik} H_{ikt} + C^6_{it} (SC_{it} + X_{ikt})$$
$$+ C^9_i EI_{ikt} + C^9_{kt} O_{kt} + C^8_{kt} D^-_{ikt} \Big]$$
$$+ D^-_{6t} - D^+_{6t} = PRC_t \qquad (5.15)$$

where,

H_{ikt} = shrinkage of work in process inventory of the ith product in the kth department in period t;

SC_{it} = subcontracted amount of the ith product in period t;

X_{ikt} = work in process inventory of the ith product in the kth department in period t;

EI_{ikt} = ending inventory of the ith product in the kth department in period t;

O_{kt} = overtime in department k in period t;

PRC_t = target level for production rate cost in period t;

and C's are the respective cost coefficients.

Cost related to inventory level:

$$\sum_i (C_i^9 FG_{it} + C_i^{10} R_{it} + C_i^{11} BO_{it})$$

$$+ D_{7t}^- - D_{7t}^+ = IC_t \tag{5.16}$$

where,

FG_{it} = finished good inventory of the ith product in period t;

R_{it} = the amount of finished good inventory shrinkage of the ith product in period t;

BO_{it} = back order of the ith product in period t;

IC_t = target level for inventory cost in period t;

and C's are associated cost coefficients.

Contribution margin goal:

$$\sum_{i} (SP_{it}-VC_{it})S_{it}+ D_{8t}^{-}- D_{8t}^{+} = CM_{t} \qquad (5.17)$$

where,

S_{it} = number of the ith product to be shipped and sold to customers in period t;

SP_{it} = selling price per unit of the ith product in period t;

VC_{it} = standard variable cost per unit associated with the production and sale of the ith product in period t; and

CM_{t} = target dollar contribution in period t.

The major constraints incorporated in the model are:

-- A budget constraint qualifying a strict limitation on cash outflows on a departmental or a plantwide basis.

-- A constraint which shows other limited productive resources required to produce various products.

-- An inventory balance equation for all finished products.

-- Constraints showing the upper and/or lower limits on any decision variables in the model.

Hindelang and Hill do not furnish a numerical example. But, it is evident from the inclusion of a large number of constraints and variables that the model is very large. More will be said about this model in the following section.

Drawbacks of the models

Although the above models are theoretically attractive, they lack many of the features that are important to management. The major drawbacks are described below.

All the existing MCAPP models lack a feature that is considered to be one of the fundamental principles of multicriteria decision making, the ability to generate new alternatives while making a decision. The models are formal and rigid in structure in the sense that each one is solved over 'fixed' solution space formed by the constraints of the problem. The optimum solution given by the model has a fixed location in this fixed space. In most cases, however, the solution needs to be changed because of its impracticality. But if the model is implemented in a strict sense, no solution other than the one already found can be used; because the model is not capable of providing any better solution. But it is possible that near the vicinity of the currently available best (most satisfactory) solution there can be a better feasible solution. This idea can be illustrated using Figure 5. Suppose that A is the most satisfactory solution now available from any one of the existing models. If at this stage, the solution A' can be made feasible, the decision maker will certainly prefer it, because at A' both of the objective functions f_1 and f_2 have values better than at A. In fact, any solution other than A in the doubly shaded region is preferable.

In aggregate production planning, the work force levels in different periods are the variables that can be controlled by the decision maker. A model can be developed which will incorporate these decision variables in such a way that by changing these variables new alternatives (solution space) can be generated and investigated in order to seek better solutions. In fact, the new model developed by this research will identify in advance which alternative(s) can give a better result. The existing MCAPP models (as well as single criterion models) do not provide this insight.

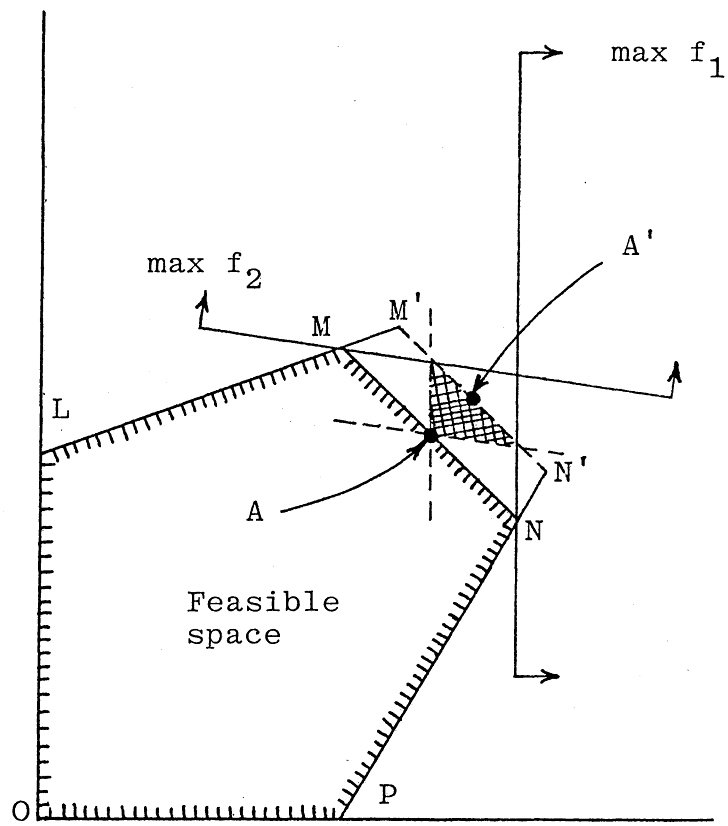


Figure 5. Generation of New Alternatives

As a by-product of inclusion of this capability, that is, the capability of generation of new alternatives, another significant benefit will be obtained. This benefit is the incorporation of greater flexibility toward stabilizing the work force level. As noted earlier, work force stability is considered as one of the major concerns of management, and the option of changing the work force level to cope with the fluctuation of demand is a very expensive one. Stability in the work force level is important not only because of the pressure of labor unions but also for economic reasons. It has been reported that constant work force models (manpower pooling models) can sometimes offer better (economic) results than pure hiring and firing models [7].

In this regard, Goodman as well as Hindelang and Hill did some work, but their models lack two things. First, the models lack realism in the sense that the relationship between the duration of overtime and the timing of hiring the workers, and the relation between the duration of idle time (or inventory build up period) and the timing of firing the workers were not incorporated into these two models. The result is that the immediate past information about the work force level is ignored. But it was noted earlier that the decision regarding the work force level of any period, particularly the first period of the planning horizon, is greatly influenced by the status of idle time, overtime etc., of the immediate past one or more periods. Secondly, the nature of these models and the solution techniques used to solve them is such that hiring and firing in any period can be of any magnitude. However, it has been noted earlier that the decision maker does not consider hiring or firing of any amount in any period as a feasible option for production smoothing.

With respect to transmitting information about the system to the decision maker, the methods used in the existing models have two basic problems. Firstly, they are incapable of providing information about the entire system. Note that in the case of goal programming, the decision maker knows only about one point (the final solution) after the problem is solved. And in the case of multiobjective linear programming, information is available (after the problem is solved) only about the nondominated extreme points which are discrete points in the solution space. The decision maker remains completely ignorant about the rest of the solution space (infinite number of feasible alternatives). Second, the methods do not provide the decision maker with an insight into the system.

Ideally, a method should be capable of providing information about what effect a change in a decision variable will have on a particular criterion. This characteristic offers two benefits. One benefit is that it supports the decision maker, quantitatively, in making a sound decision. When a solution is obtained after using a model and the solution is found not to be favorable for the firm, the decision maker may be willing to change some of the decision variables to make the solution suitable for his firm. At this stage, if the consequence of changing a preferred decision variable is known, the decision maker may or may not change the variable depending on the accurate information. This keeps the decision maker from being in a dilemma over how to change the variables and, finally resorting as usual to a solution based on past experience.

The other benefit of this model which allows insight into the entire system is that at the end of the decision-making process, the decision maker is fully convinced that there can be no solution better than the one he found at the final stage, because the model provided him with enough information about the behavior of the system performance.

In order to avoid confusion, it is worth mentioning here that the generation of new alternatives and the collection of information about them are two completely different concepts. These two concepts are fundamental to solving any multicriteria decision-making problem.

The drawbacks specific to a particular model are described below. Lawrence and Burbridge as well as Hindelang and Hill consider sales or profit as one of their goals. But it has been mentioned earlier, in reference to Shearon's survey, that these two factors are beyond the control of operations managers. Consequently, the inclusion of these factors in a model is not only irrelevant but also a cause of unnecessary computational burden.

Lawrence and Burbridge as well as Hindelang and Hill consider more than one item and many minute details in their models. But this approach is undesirable, because the system becomes large and complicated. In this regard, Bitran et al. [1], in developing their hierarchical production planning system, maintain that they

> . . . favor an aggregate allocation approach at the higher level of the hierarchical system to avoid the massive data manipulation, computational complexities, and forecasting inaccuracies that would be imposed by a detailed allocation model at that level. (p. 234)

Similar arguments regarding this have also been reported in [8].

Computational complexity is a particularly serious problem in multiobjective linear programming when the number of constraints and variables increases causing an increase in the number of extreme points. Each of these extreme points has to be checked for nondominance. It is, therefore, suggested that the MCAPP models should be restricted to aggregate levels of items only.

CHAPTER VI

THE PROPOSED MODEL AND ITS SOLUTION TECHNIQUE

Introduction

In this chapter, the proposed model and its solution technique will be described. As mentioned earlier, one of the objectives of this research is to search for a technique that will help reduce the number of constraints and/or variables to be incorporated in the model while retaining the exactness of the solution compared to the solution obtained from the original model. The search ended with the conclusion that such a technique can be developed.

Based on this result, the author proposes to call the resulting model the 'working model,' and the model with all the constraints and variables as the 'fundamental model.' In this chapter, only the fundamental model will be described. The working model will be described in the next chapter. Also, in this chapter, some of the problems encountered by the author in constructing the model will be mentioned. These problems will provide information regarding the extent to which a linear aggregate production planning model can differ from the actual system.

Some Problems in Formulating a Linear Aggregate Production Planning Model

In aggregate production planning, the problem faced by the model builder in treating the variables representing the productivity of workers at different periods is very acute because the productivity of a worker depends on the number of units he produces. In other words, the productivity depends on the time spent by the worker in the learning process [9,14,17]. If one wants to incorporate this fact into a model, the model will become nonlinear because of the presence of a term which is equal to the product of two variables: 1) the worker level and 2) the worker productivity, the two variables being unknown in advance.

A related problem is how to classify the workers according to their productivity. Theoretically, there are as many classes as the number of workers in the firm because each worker's productivity is, theoretically, different from that of others. For simplicity, however, we may form classes of workers by considering their skills to fall within a certain range. But this will result in other problems.

The manager will have to keep a list of worker classes together with the productivity of each worker, and update this list at the end of each planning period, since the membership of a worker in a class is not necessarily permanent.

This is because, as the production continues, some workers might work overtime during a certain period, whereas, some other workers in the same class may have to remain idle during other period. This will cause a difference in skills among the workers of the same class, resulting in further division of the same class in the next period.

The question is how the manager will keep track of this situation. The problem is further aggravated when new workers are hired and the manager might have no idea about the population of the productivity with which the new workers enter the firm. A little thinking on this issue will reveal that this problem, in its exact form, is practically impossible to solve.

In order to resolve the above mentioned problem, it is required to know to what extent the classification of workers is critical to the manager. Currently, no information is available about this. However, observe that Shearon's survey provides us with the information about how long the new workers, the layed off workers, and the workers who are bumped down take to resume their standard production rate. This period varies from two to six weeks depending on the status of the workers mentioned above. This period is relatively short and managers of most of the firms producing short cycle products will accept a small difference in productivity of the workers rather than accept the troubles of continuously keeping the records of different classes of workers.

The purpose of this research is neither to provide an answer to the question regarding the justification of division of workers, nor to provide a means that will, if such divisions are made, define the nature of the divisions. However, here, provision has been made to work with two classes of workers. The workers may be classified according to their skill. The workers hired at any period may be placed in the least experienced class, and those who are already present are placed in either the most experienced class or in the least experienced class depending on their skills. This classification is maintained throughout the planning horizon. Since aggregate production planning is done on a rolling horizon basis, the manager may, at the end of the first period, transfer some of the workers from the least experienced class to the most experienced class, and for the new rolling horizon these values may be used as initial values.

Another problem is how to handle the overtime variable. The overtime production quantity at any period is given by:

$$\begin{aligned}\text{Overtime production} = \; & \text{Duration of overtime} \\ & \times \text{Overtime workers} \\ & \times \text{Overtime productivity} \qquad (6.1)\end{aligned}$$

Observe that all the three quantities on the right hand side are unknown variables. To the knowledge of the author, there is not a single operations research technique that can solve an aggregate production planning model containing expressions of this nature. Simplification of this expression is needed, and is done by modifying (6.1) in the following way.

$$\text{Overtime production} \leq \text{Maximum overtime duration} \times \text{Overtime workers} \times \text{Overtime productivity} \quad (6.2)$$

In this expression, 'Maximum overtime duration' is a fixed quantity. But the problem of nonlinearity still remains unless a suitable value for overtime productivity is assumed. By making the expression linear, however, the exactness of the system is lost. According to this formulation, in the expression of the total cost, the cost due to overtime will have to be included as a function of production quantity during overtime. But, in reality, the workers are paid on the basis of the duration they work overtime, and not on the basis of the overtime quantity they produce.

These two problems plus the one associated with the nonlinearity of the cost structures, allow us to realize the complexity of aggregate production planning problems.

Assumptions and Objective Functions

The assumptions of the model are:

-- Demand is deterministic and is satisfied by regular time, overtime, and or subcontracting.

-- Inventory holding cost is based on ending inventory of a period.

-- The objective functions as well as the constraints are linear.

-- No backlog or lost sale is allowed.

-- If overtime exits, it is allocated to the workers of the experienced class first, then to the workers of the next higher skill, and so on. A reverse order is maintained in the case of idle time.

The objective functions considered in this model are:

1. Minimize the total regular payroll, hiring, and firing cost.
2. Minimize the total inventory carrying cost.
3. Minimize the total number of idle workers.
4. Minimize the total overtime production cost.

The first, second, and fourth objectives are obvious. The third objective, in fact, indirectly represents idle time. An objective function representing the sum of the total amount of workers either hired or fired could have been included. Since hiring and firing at any period are both positive quantities and loss in productivity takes place whenever there is a hiring or firing, this function would represent the total loss in productivity if this loss is assumed to be directly proportional to the amount of hiring and firing at different periods.

Structure of the Fundamental Model

The following symbols are used in the formulation of the model.

T = planning horizon;

D_t = demand in period t;

I_0 = initial inventory;

I_t = inventory at the end of period t;

W_0 = initial work force;

W^j_{rt} = regular work force of the jth class in period t (maximum limit of the total work force is $\overline{W}_{rt}$);

W_{ht} = work force hired in period t;

W_{ft} = work force fired in period t;

W^j_{it} = work force that will remain idle in period t (jth class);

P_{rt} = total regular time production in period t;

P^j_{ot} = overtime production in period t by the jth class of workers;

P_{st} = amount to be subcontracted in period t;

e = efficiency of the workers during overtime;

f = fraction of the regular time allowed for overtime;

d^j_{rt} = regular time productivity per worker of the jth class per month in period t;

d^j_{ot} = overtime productivity per worker of the jth class per month in period t;

C^j_{rt} = average regular payroll per worker of the jth class per month in period t;

C_{ht} = hiring cost per worker in period t;

C_{ft} = firing cost per worker in period t;

C_{ot} = cost per unit produced during overtime in period t;

C_{st} = unit cost of products subcontracted in period t;

C_{ct} = inventory carrying cost per unit per period.

K = number of objective functions considered.

The subscript t is attached to the cost coefficient terms for the purpose of generalization.

<u>Objective Functions</u>

Minimize the total regular payroll, hiring, and firing cost:

$$\text{Min.} \quad f_1 = \sum_{t=1}^{T} \left[\sum_{j=1}^{2} C_{rt}^{j} W_{rt}^{j} + C_{ht} W_{ht} + C_{ft} W_{ft} \right] \tag{6.3}$$

Minimize the total inventory carrying cost:

$$\text{Min.} \quad f_2 = \sum_{t=1}^{T} C_{ct} I_t \tag{6.4}$$

Minimize the total number of idle workers:

$$\text{Min.} \quad f_3 = \sum_{t=1}^{T} (W_{it}^{1} + W_{it}^{2}) \tag{6.5}$$

Minimize the total overtime production cost:

$$\text{Min.} \quad f_4 = \sum_{t=1}^{T} (P_{ot}^{1} + P_{ot}^{2}) C_{ot} \tag{6.6}$$

<u>Constraints</u>

$$\sum_{t=1}^{T} (P_{rt} + \sum_{j=1}^{2} P_{ot}^{j} + P_{st}) + I_0 = \sum_{t=1}^{T} D_t \tag{6.7}$$

$$P_{rt} = \sum_{j=1}^{2} (W_{rt}^{j} - W_{it}^{j}) \, d_{rt}^{j} \qquad t=1, \ldots, T \tag{6.8}$$

$$\left. \begin{array}{l} W_{rt}^{2} \\ \\ W_{rt}^{1} \end{array} \right\} = f(W_{ht}, W_{ft}, V_t) \qquad t=1, \ldots, T \tag{6.9}$$

$$I_t = I_{t-1} + (P_{rt} + \sum_{j=1}^{2} P_{ot}^{j} + P_{st}) - D_t \qquad t=1, \ldots, T \tag{6.10}$$

$$W^1_{rt} + W^2_{rt} \le \overline{W}_{rt} \qquad t=1, \ldots, T \qquad (6.11)$$

$$P^j_{ot} \le f_{max} W^j_{rt} d^j_{ot} \qquad j=1,2;\ t=1, \ldots, T \qquad (6.12)$$

$$W_{ht} \le \overline{W}_{ht} \qquad t=1, \ldots, T \qquad (6.13)$$

$$W_{ft} \le \overline{W}_{ft} \qquad t=1, \ldots, T \qquad (6.14)$$

$$W^1_{it} + W^2_{it} \le \overline{W}_{it} \qquad t=1, \ldots, T \qquad (6.15)$$

$$f_k \le \overline{f}_k \qquad k=1, \ldots, 4 \qquad (6.16)$$

In the above expressions all the variables are nonnegative.

Then the problem may be stated as

$$\begin{aligned} &\text{Minimize} \quad (6.3) \sim (6.6) \\ &\text{subject to } (6.7) \sim (6.16) \end{aligned} \qquad (6.17)$$

Equation (6.7) balances the total production and demand throughout the planning horizon. Equation (6.8) expresses the relation among the number of workers, their productivity, and the number of products produced by them during the regular time. Equation (6.9) expresses the relation between the number of workers in different groups of the current period and that of the previous period. More about this equation will be presented in the next paragraph. Equation (6.10) balances the ending inventory of the current period with the production of the current period, demand of the current period, and the ending inventory of the previous period. The maximum limit of the regular work force is given by (6.11). Equation (6.12) indicates the upper limit for overtime production. Equations (6.13), (6.14), and (6.15) specify the upper limits of workers hired, workers fired, and workers kept idle, respectively. Equation (6.16) specifies the upper limit of the kth objective function.

Equation (6.9) requires explanation. When a model with a single class of workers is considered, the structure of this equation is given by

$$W_t = W_{t-1} + W_{ht} - W_{ft} \tag{6.18}$$

But when there is more than one class, it is necessary to ascertain if people are to be fired from more than one class. When the number of workers to be fired is greater than that in the least experienced class, it will be necessary to fire some people from the next higher skilled class with the condition that the least experienced class of workers will be fired first. In the case of two classes of workers this condition can be met by the following two equations.

$$W_{rt}^{2} - V_t = W_{r,t-1}^{2} + W_{ht} - W_{ft} \tag{6.19}$$

$$W_{rt}^{1} = W_{r,t-1}^{1} - V_t \tag{6.20}$$

The variable V_t in the above expressions is nonnegative.

It may be noted that the constraint (6.12) does not contain the idle worker term. This condition together with low cost for regular time production compared to overtime cost of production ensures that the idle time and overtime will not exist simultaneously in the same period.

The constraints (6.13) $\sim$ (6.16) need special attention. As for the constraints (6.13) $\sim$ (6.15), the reader should keep in mind that the right-hand side quantities of these constraints do not represent the upper limits of the left-hand side variables in the sense they usually do, as in the case of the existing models. In the existing models, these limits are fixed. But in the proposed model, these are not fixed quantities. Instead, these are variables, and will receive special treatment as described in the following paragraphs.

The way the constraints (6.13) $\sim$ (6.16) will be handled is the major development of this research and needs to be discussed in detail. Equations (6.13) $\sim$ (6.15) will be explained first. When a linear programming problem is solved, at the optimal stage the dual variables corresponding to the inequality constraints represent the change of the objective function that can be obtained by a unit change of the corresponding resource. Notice that the variables W_{ht}, W_{ft}, and W_{it} can be controlled by the decision maker. By changing these variables it is possible to change the solution space. Speaking in terms of the decision-making process, it is possible to generate new alternatives. Since in multiobjective linear programming one objective function will be considered at a time, the values of the dual variables corresponding to these constraints can be conveniently used as sources of information for changing the values of $\overline{W}_{ht}$, $\overline{W}_{ft}$, and $\overline{W}_{it}$ in order to improve the objective function which is currently under the process of improvement.

Also, by changing the values of $\overline{W}_{ht}$, $\overline{W}_{ft}$, and $\overline{W}_{it}$ the decision maker can control the level of the work force in a particular period. He can keep track of how many periods the workers worked overtime, when the new workers are to be hired to eliminate overtime, and such other things related to work force level. Thus, the decision maker can employ the information about hiring, firing, overtime, and idle time during the past one or more periods in setting the values of $\overline{W}_{ht}$, $\overline{W}_{ft}$, and $\overline{W}_{it}$ for future planning.

So far as the constraint (6.16) is concerned, the reader may, in advance, be reminded of the fact that while one of the objective functions will be under the process of improvement, the other objective functions will act as constraints of the problem. So, at the optimal solution stage for a particular objective function, the dual variables corresponding to the other objective functions will provide the decision maker with the information about how much an objective function will have to be sacrificed in order to improve the one under the process of improvement.

From the above discussion it is clear that the decision maker may not be constrained to implement only one solution given by the model as is the case with the existing models. Rather, since he is able to get an insight to his system, the search can be extended until he is convinced that there cannot be a solution better than the one at hand. In other words, before the final decision is made, the decision maker exhausts all of the promising alternatives.

It may be observed that the values of $\overline{W}_{ht}$, $\overline{W}_{ft}$, and $\overline{W}_{it}$ can convert the model to a pure manpower pooling one or to a pure hiring and firing one.

Solution Technique

The Method of Satisfactory Goals (MSG) [6] will be used to solve the model. A modification of the method is required. This will be described after the MSG is presented. The different steps of the MSG are described below.

Step 1: Specify a set of maximum acceptable (feasible) initial goal levels, M_k, $k = 1, \ldots, K$, where K is the number of objective functions incorporated in the model.

Step 2: Identify the least satisfactory goal, say LS.

Step 3: Solve the following problem.

$$\text{Minimize} \quad f_{LS}(X)$$

$$\text{Subject to} \quad g_i(X) \le 0, \quad i=1, \ldots, m \tag{6.21}$$

$$f_k(X) \le M_k, \quad k=1, \ldots, K, \; k \ne LS \tag{6.22}$$

where,

$g_i(X)$ are the constraints of the problem,

$f_k(X)$ are the objective functions, and

X are n decision variables.

Step 4: Utilize the knowledge obtained from the dual variables related to the constraints in (6.21) and (6.22) in order to relax or tighten them to improve the value of f_{LS}.

Repeat Steps (2) - (4) until satisfactory values of all the goals (objective functions) are obtained.

The MSG has the drawback of specifying the initial (feasible) maximum acceptable levels of the objective functions (the right-hand side of the constraints (6.22)). But, so far as aggregate production planning problems are concerned, this is not a serious problem, because, as was seen earlier from Shearon's survey, managers emphasize the reduction of the cost over the entire planning horizon. Based on this information, the initial values of the right-hand side of the constraints in (6.16) will be computed as described below.

At first, the following problem will be solved.

$$\begin{array}{ll} \text{Minimize} & \text{Total production cost} \\ \text{Subject to} & (6.7) \sim (6.15) \end{array} \tag{6.23}$$

Then the values of the objective functions in (6.16) will be computed with the help of the results obtained from (6.23). These values will be used as the initial upper limits of the constraints (6.16). From this point on, the other steps of the MSG will be followed.

The modification described above offers the following benefits.

-- It eliminates the difficulty in choosing the initial values of the right-hand sides of (6.13) $\sim$ (6.16).

-- Since the starting point will be at the globally optimum solution with respect to the total cost criterion, the final solution is likely to be obtained with a smaller number of iterations.

-- Since the economically global solution is at hand, the deviations from this solution will act as inputs for subjective evaluation of the results obtained from Step 4 of the solution technique.

With the modification suggested for obtaining the initial feasible solution, the steps of MSG technique are shown below.

Step 1: Solve the problem stated in (6.23) and compute f_1, f_2, f_3, and f_4 given in (6.3) $\sim$ (6.6). Set these values as initial feasible maximum goal levels for f_1, f_2, f_3, and f_4 respectively.

Step 2: Identify the least satisfactory goal from f_1, f_2, f_3, and f_4. Call it f_{LS}. The subscript LS is the identification number for the least satisfactory goal (objective function).

Step 3: Solve the following problem.

$$\begin{aligned} &\text{Minimize } f_{LS} \\ &\text{Subject to} \quad (6.7) \sim (6.15) \\ &\text{and} \quad (6.16) \text{ with } k \neq LS \end{aligned} \tag{6.24}$$

Step 4: Utilize the knowledge obtained from the dual variables related to the constraints in (6.24) in order to relax or tighten them to improve the value of f_{LS}.

Repeat Steps (2) - (4) until satisfactory values of all the goals (objective functions) are obtained.

It may be noted that this method of solving MCAPP problems deserves some special attention. First, all nondominated extreme points do not have to be investigated. In fact, the method does not investigate any point for nondominance. Second, it explores only a subset of all feasible solutions. These two features will become clear from Figure 6.

In Figure 6, three objective functions f_1, f_2, and f_3 are considered. (Notice that all solutions are nondominated and all extreme points are nondominated extreme points.)

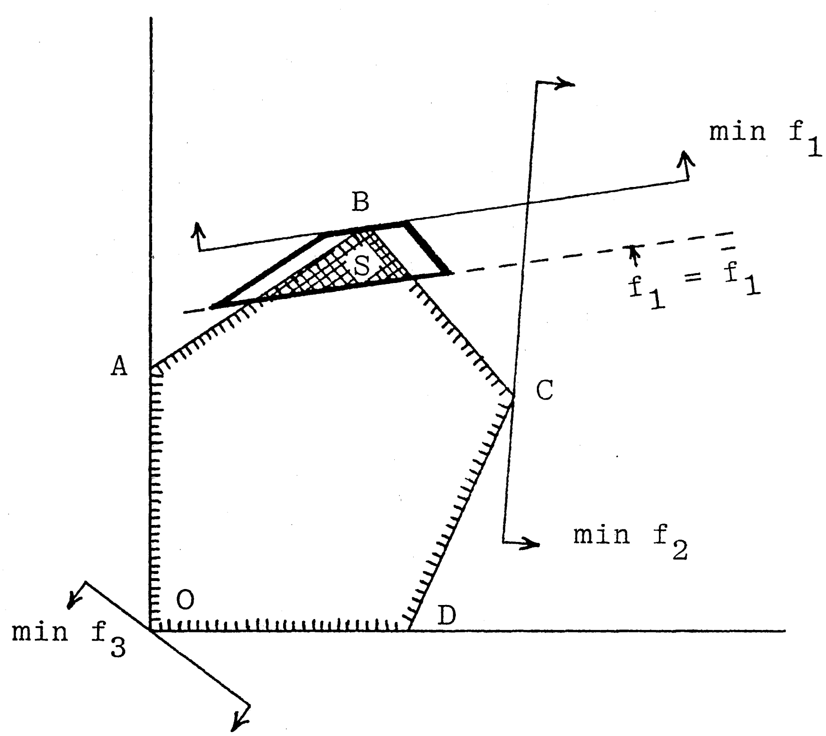


Figure 6. Solution Space to be Investigated

Assume that the minimum value of the overall function f_1 representing some criterion of major interest is at B. Then, obviously, the decision maker's objective will be not to deviate much from the point B. If the decision maker sets the maximum acceptable level of f_1 (the major criterion) to be equal to $\bar{f}_1$, the solutions that are worth exploring are only those represented by the shaded area S and some others surrounding S (the area within the thick lines). The area other than S within the thick lines represents the alternatives that have to be generated by adjusting the constraints represented by the lines AB and BC. In Figure 6, the area within the thick lines has been generated by arbitrarily relaxing AB and BC. In real situations, this has to be done according to the need of the decision maker. Thus, it is clear that with the help of this technique, a vast majority of the solution space can be safely excluded so far as MCAPP problems are concerned. Finally, the method is simple and easy to understand.

In summary, it can be said that the proposed model, together with the solution technique mentioned above, serves as a source of information for finding the most satisfactory solution.

CHAPTER VII

THE NATURE OF THE WORKING MODEL

Introduction

In this chapter, a method for reducing constraints and/ or variables will be described. The reader, by this time, might be aware of the fact that the model described in (6.17) contains a large number of constraints and variables. Consequently, a considerable amount of computational time is needed to solve this problem. Particularly in the case of MCAPP, the same problem has to be solved (with respect to different objective functions) several times before a satisfactory solution is found. As a result, the large computational time may render the model unattractive.

In this chapter, an attempt has been made to build the concept and the structure of a method that will help the model builder exclude safely some of the constraints and variables. Also, it will be made clear how this technique will allow the model builder to incorporate forcibly some of the features (options of production smoothing) or exclude some of the features at any particular period of interest. In other words, this technique will make the model dynamic. This approach has never been applied to aggregate production planning, and as such, it may not be clear to the reader until the end of this chapter. It may be noted that although the model will be solved using only two classes of workers, the concept behind the technique will be illustrated with more than two classes of workers.

Work Load Pattern

In this section, it will be demonstrated how the model builder may determine the approximate schedule of workers throughout the planning horizon. Initially, the productivity of workers in different classes at different periods is to be determined. This can be done by utilizing the knowledge of the learning rate applicable to the firm. It is assumed here that the workers will be employed throughout the planning horizon. With this information, it is possible to determine the number of products that will be produced by each class of workers at each period. Conversely, it is possible to determine how many classes of workers will be employed for regular time, overtime, etc. at different periods. Also, it is possible to determine how many products are required to be subcontracted, if after allocating regular time and overtime the demand cannot be met. However, since at certain periods some workers in a class or some classes of workers might be fired or layed off, in the presence of the effect of learning, this information is not perfect. It is worth mentioning that this allocation of work is done only with the work force that exists at the beginning of the planning horizon. For a typical demand and initial work force, the resulting worker schedule for a nine-period planning horizon will look like what is shown in Figure 7.

It should be noted that this figure does not represent a demand curve. Also, the curve is not proportional to the demand curve because the productivity is not the same for all the workers. From now on, the curve in Figure 7 will be referred to as 'work load pattern.'

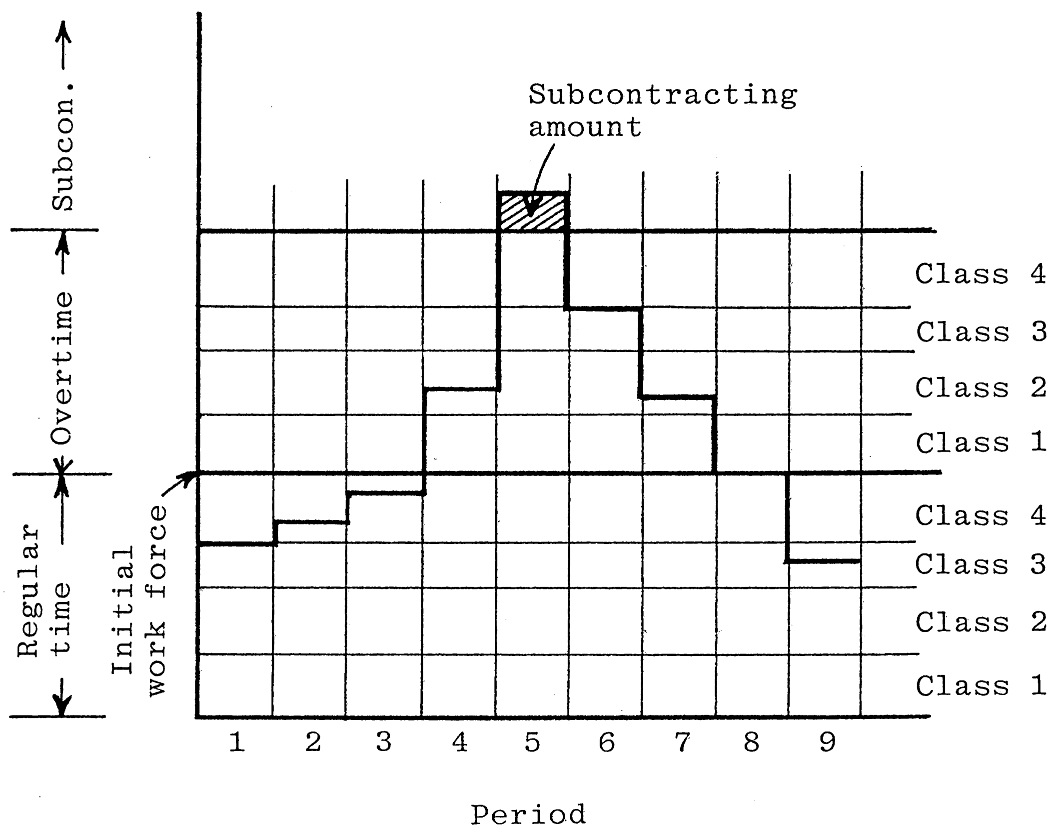


Figure 7. Work Load Pattern

Constraints and Variables Reduction Process

Certain Parts and Uncertain Parts

The author postulates that in general aggregate production planning problems can be thought of as consisting of two parts, namely, the certain part, and the uncertain part. The certain part is comprised of the options of production smoothing about which the decision maker is certain as to their presence or absence. On the other hand, the uncertain part is comprised of the options about which the decision maker is not certain.

For example, when demand increases beyond the capacity of the currently employed workers, the question of considering firing and idle time during the periods through which this increasing demand situation prevails does not arise at all. That is, the decision maker is certain about his strategy during this period in that he will never consider firing and idle time as options for production smoothing. On the other hand, the increasing demand situation mentioned above can be handled by hiring, overtime, or utilizing inventory built in the previous periods. However, the decision maker does not know which of these options or combination of options will give the most satisfactory result with respect to some criterion.

The three options mentioned above are as a whole considered as the uncertain part. In other words, the options of firing and idle time can be safely excluded from the model for those periods. In general, the constraints and/or variables related to the certain part do not need to be included in the model, while those for the uncertain part need to be included in the model. A method has been developed to determine the certain part and the uncertain part; this method aims at recognizing the work load pattern described earlier.

Efficient Utilization of Resources

While the method will seek to recognize the work load pattern, it will simultaneously apply another principle which the author calls "the principle of efficient utilization of resources," where the resources, mainly the regular work force, are the available options. The principle may be described as:

> Utilize the resources in each period in such a way that the effects of this utilization on the performance during this period and the later periods are most satisfactory, and not in a way such that once the period is over, it becomes apparent to the decision maker that there could have been some better combination of options to handle the situation of the last period.

Consider Figure 8 to get a clear picture of what has been said so far about the method. For simplicity, only one class of workers with constant productivity throughout the planning horizon is considered. Let the dotted line represent the production schedule corresponding to the work load pattern. Now, a hand-to-mouth strategy will dictate that the decision maker fire in period one, hire in period two, fire in period three, and so on. But, this strategy will involve a lot of money because of frequent hiring and firing.

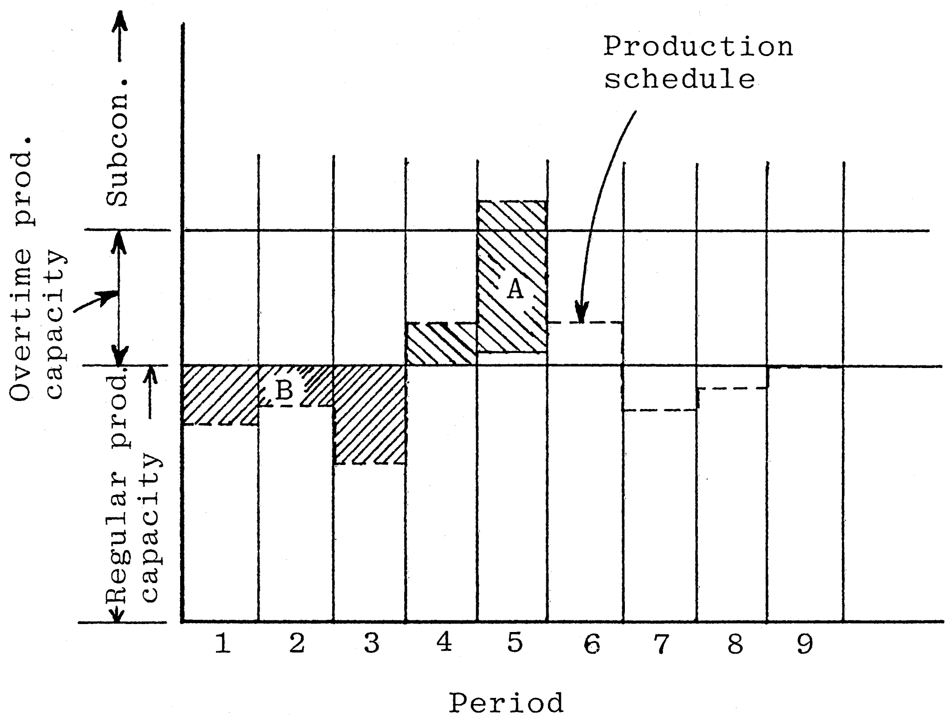


Figure 8. Efficient Utilization of Resources

If the principle of efficient utilization of resources is applied at periods one, two, and three, there is a possibility of making some products in these periods that are required in periods four, five, and six. This is assuming that the production cost in the earlier periods plus the inventory carrying cost to the later periods is less than the overtime production cost during the later periods. (Notice that the options of hiring at the later periods are ignored, because only the utilization of the existing work force is being considered.)

Assume that the production cost and inventory carrying cost permit products produced in the earlier periods to be carried to the later periods, and that the two shaded areas in Figure 8 are equal. In this situation, it is possible to maintain a constant work force up to the fourth period, and eliminate the options of firing in period one, hiring in period two, firing in period three, and overtime in period four. Note that the conventional aggregate production planning models implicitly do the same thing as is being done here, but the conventional models include all the options without checking whether or not the constraints are redundant. Because some of the options can be eliminated from consideration, the associated constraints as well as variables can also be eliminated.

The example shown here is extremely simple. In real situations where the demand patterns are irregular and several classes of workers with different productivity rates are involved, the computations for determining the possibility of making products in a period and carrying them to some future periods might become complex.

Formally, the principle of efficient utilization of resources will do the following:

If subcontracting (overtime) appears to exist at some future period(s), it will be investigated if this subcontracting (overtime) amount can be produced economically by regular time and/or overtime (regular time) at some previous periods and carried over to those future periods. When allocation of subcontracting is considered, preference is given to utilization of regular time first, then to overtime.

Let us take an example for the case of overtime, and assume that a worker gets \$1000 per month and his productivity at a certain period is 700 units per month during regular time. The variable cost is then 1000/700 = \$1.43 per unit. If the overtime cost is 1.5 times the regular time cost and overtime efficiency is 80% of the regular time production, the overtime production cost would be \$1.43 x 1.5/0.8 = \$ 2.68 per unit. If the inventory carrying cost per period per unit is \$ 0.2, the number of periods (NOP) through which a regular time product can be carried is given by the following:

$$1.43 + \text{NOP}\ (0.2) \leq 2.68$$

$$\text{or, NOP} \leq 6.25 \tag{7.1}$$

Since some fixed costs (hiring, firing), not known prior to solution, might exist, the actual value of NOP might be lower than 6.25. (Temporarily, the integer characteristic of NOP will be ignored.) Similar computations can be performed for the cases of balancing subcontracting cost and regular time cost, and subcontracting cost and overtime cost. Implicit in the assumption is that the subcontracting cost is higher than the overtime cost.

Variables Used in Recognizing the Work Load Pattern

A set of variables can be defined in such a way that the work load pattern can be expressed in terms of these variables. These variables will be used in applying the principle of efficient utilization of resources. They will also be used in the next section where the rules for excluding the constraints and/or variables will be described.

NGRBGN = Number of classes of the workers at the beginning of the planning horizon.

NRGMIN = Lowest skilled class that is fully scheduled for regular time throughout the planning horizon.

NRGMAX = Lowest skilled class to which the regular work schedule might extend.

t = Period under consideration.

NREG(t) = Lowest skilled worker class to which regular work schedule appears to extend after allocating regular work to higher skilled classes, assuming no hiring, no firing and no idle time in the previous periods.

REGAVL(t) = Amount of regular work of the class NREG(t) that is left unallocated when demand is less than the current worker capacity.

NRGAVL(t) = Number of regular workers (of the same productivity as that of NREG(t)) equivalent to the amount of work REGAVL(t).

NOVR(t) = Lowest skilled worker class number to which overtime schedule appears to extend after allocating overtime work to the highest skilled workers first, the next experienced class next, and so on, assuming no hiring, no firing, and no idle time in the previous periods.

OVRAVL(t) = Amount of overtime of the class NOVR(t) that has not been allocated because demand is such that the total overtime capacity does not have to be completely utilized.

NOVAVL(t) = Number of overtime workers (having the same productivity as that of workers in NOVR(t) during overtime) equivalent to the amount of work OVRAVL(t).

SUBCON(t) = Amount of subcontracting (in terms of units of product) that appears to exist in period t, after employing all regular time and overtime.

RGWCAP(n,t) = Regular work capacity (in terms of units of product) of the nth class of workers during period t.

OVRCAP(n,t) = Overtime capacity (in terms of units of product) of the nth class of workers during period t.

Figure 9 explains the meaning of the terms defined above.

In this figure:

NRGMIN = 2, NRGMAX = 4.

NREG(1) = 3, NREG(2) = 4, ..., NREG(9) = 3.

NRGAVL(2) = C; C may be equal to 0; for example,

NREG(8) = 4, but NRGAVL(8) = 0.

NOVR(1) = 0, ..., NOVR(4) = 2, ..., NOVR(9) = 0.

NOVAVL(5) = 0, NOVAVL(7) = D.

SUBCON(1) = 0; SUBCON(5) is equal to the shaded area.

The values of these variables for other periods are not given. They are readily available from the figure. The reader may be reminded that RGWCAP(n,t) ≠ RGWCAP(n,t'); also, RGWCAP(n,t) ≠ OVRCAP(n,t).

Observe that these variables are sufficient to represent the work load pattern. Once this is done, the task of recognizing the work load pattern is over. After doing this the principle of efficient utilization of resources will be used to get the modified work load pattern. Notice that the work load pattern is changed when some subcontracting or overtime in the later period(s) is reallocated to available regular time in earlier periods.

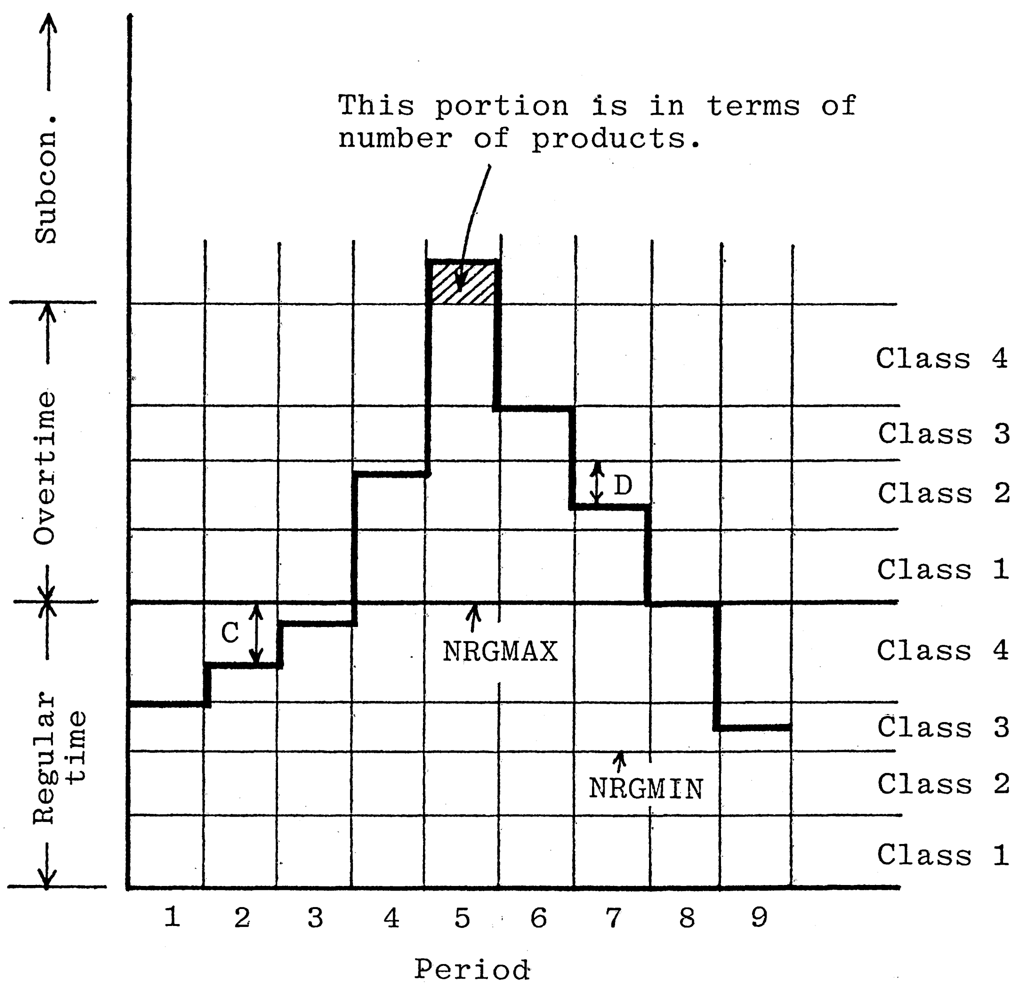


Figure 9. Quantitative Description of Variables

Rules for Elimination of Constraints and Variables

After determining the modified work load pattern, the rules for elimination of constraints and variables are applied. These rules are given below.

With respect to regular time, the following rule is applied.

> There will be no regular time constraints (see (6.3)) for workers of class numbers one to NRGMIN. Remove (adjust) the variables associated with this constraint from the other constraints of the model.

Notice that NRGMIN classes of workers will be working regular time throughout the planning horizon.

With respect to overtime the following rule is applied.

> There is no overtime in the period t', where t' is given by $t' = [t|NREG(t) \leq NRGMIN]$.

With respect to hiring the following rules are applied.

> There is no hiring in t', where t' satisfies any one of the following expressions.

$$\begin{aligned}
t' &= [t|SUBCON(t) < SUBCON(t-1)];\\
t' &= [t|NOVR(t) < NOVR(t-1)];\\
t' &= [t|NOVR(t) = NOVR(t-1),\\
&\qquad OVRAVL(t) \geq OVRAVL(t-1)]; \qquad (7.2)\\
t' &= [t|NREG(t) < NREG(t-1)];\\
t' &= [t|NREG(t) = NREG(t-1),\\
&\qquad REGAVL(t) \geq REGAVL(t-1)].
\end{aligned}$$

> The rules for firing are such that if hiring is present at any period, firing will not be present at that period except in the cases where it is not known what the correct option would be, and as such both options are specified.

> Regarding idle time, the rule is not to include this option in a period if subcontracting or overtime is present in that period.

The recognition of the work load pattern as well as the application of the rules described above is relatively simple in the cases where the subcontracting and/or overtime is apparent. But there are cases in the recognition phase where (with the present status of the technique) neither subcontracting nor overtime can be recognized in the work load pattern. An example of this case is described in Figure 10. Observe that the work load pattern indicates that there is no overtime in periods six through nine. But there is a possibility that overtime will exist during period nine. Similarly, idle time might be present during period seven.

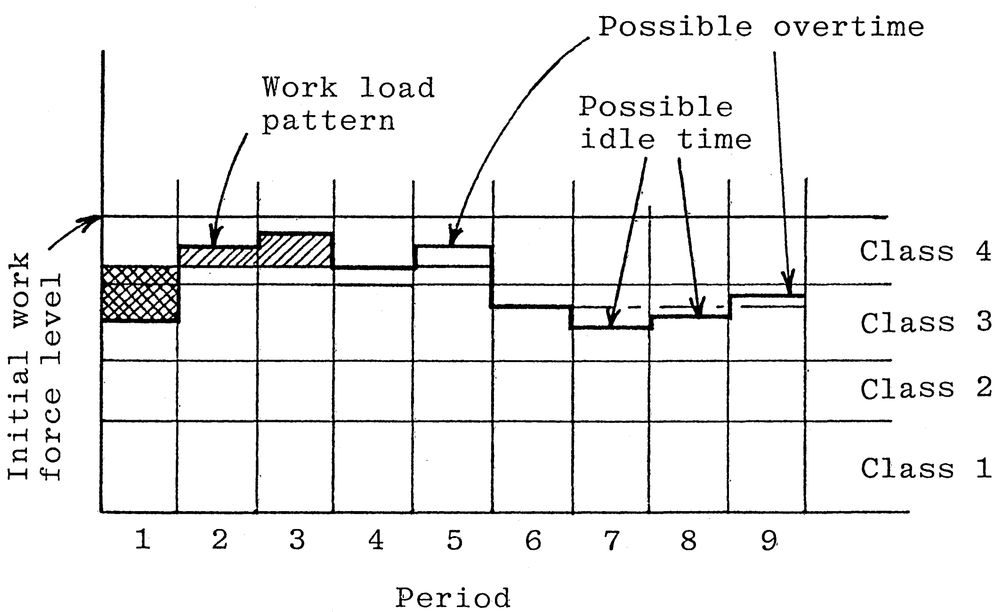


Figure 10. Hidden Possibility of Overtime and Idle Time

Since these situations are extremly difficult to detect by formal mathematical rules, the following rules are included for safety.

> Add an overtime constraint whenever there is a hiring, and an idle time constraint whenever there is a firing (determined by the rules used in this technique).

The reader might have realized by this time that a graphical display of the work load pattern is very helpful in identifying the possible options at different periods. It is possible to develop computer codes that will allow the decision maker to specify which options should be included. He may also specify which options are not to be included. For example, in the case of Figure 10, the decision maker might specify that there will be no firing and no hiring in periods seven and nine respectively. Instead, there will idle time and overtime in these periods respectively. Observe that the idle time constraint in period seven may be redundant if overtime production required in period nine can be made in the seventh period. This discussion suggests that the aggregate production planning can be done by the decision maker in a convenient and rational way with the aid of a computer graphics terminal.

In order to assist the reader in understanding how the technique works, the author feels that it is appropriate to describe the technique in detail. This is done in the next section.

The Technique

The technique developed for constructing the working model basically consists of two stages:

Stage 1: Identification of certain parts and uncertain parts.

Stage 2: Arrangement of the 'selected' constraints ((6.7) through (6.16)) and different objective functions in order to obtain a matrix form to be solved by linear programming. The matrix formed by the elements of the left hand-side of the constraints mentioned above will be called A.

<u>Stage 1</u>:

Stage 1 is comprised of the following steps.

Step 1: The work load pattern is identified in terms of NREG(t), REGAVL(t), NOVR(t), OVRAVL(t), and SUBCON(t).

Step 2: The rules (pages 76 through 78) are applied to specify various options of production smoothing for all periods.

Step 3: The principle of efficient utilization of resources is applied to get a modified work load pattern. This is done by performing the following operations.

(1) Allocate possible subcontracting in later periods to unallocated regular time and/or overtime in the earlier periods, preference being given to allocation of subcontracting to regular time.

(2) Allocate overtime of the later periods to unutilized regular time of the earlier periods. This is done when operation (1) is complete for all periods of the horizon.

(3) Repeat operation (1) once more.

The technique starts from the last period of the horizon.

For ease of understanding, operation (1) will be explained in detail. Based on the information obtained in Step 1, the period nearest the last period (including the last period itself) with subcontracting is identified. Note that SUBCON(t) > 0 indicates that there is a possibility of subcontracting in period t. Let this period be called TSUB. After finding a TSUB (if one exists), the technique looks for a period (nearest to TSUB) with unutilized regular time. Let this period (if it exists) be called TREG. Note that any one of the following conditions indicates the existence of TREG.

(i) NREG(t) < NGRBGN

(ii) NREG(t) = NGRBGN, REGAVL(t) > 0.0

An attempt is then made to allocate the subcontracting of TSUB to the unutilized regular time of TREG.

While performing operation (1), one of the following cases may occur:

Case 1: TREG exists, equation (7.1) is satisfied, but unutilized regular time in TREG is not sufficient to absorb all subcontracting in TSUB.

Case 2: TREG exists, equation (7.1) is satisfied, but the subcontracting amount of TSUB is less than unutilized regular time of TREG.

Case 3: TREG exists but (7.1) is not satisfied.

Case 4: TREG does not exist.

In the first case, all unutilized regular time is used and another TREG is sought. Again one of the four cases mentioned above may arise.

In the second case, all subcontracting in TSUB is allocated to unutilized regular time in TREG. Another TSUB is sought. (If the new TSUB is found to be less than TREG, a new TREG is sought.) If the new TSUB is found and operation (1) is attempted, again one of the four cases mentioned above may arise.

In the third and fourth cases, a period (less than but nearest to TSUB) with unutilized overtime is sought. Let this period be called TOVR (if exists). Operation (1) is attempted and again one of the four cases may occur, except that in this case unutilized overtime is considered instead of unutilized regular time.

Operation (2) is exactly the same as operation (1) except that an attempt is made to allocate overtime instead of the subcontracting amount. In operation (3), operation (1) is repeated, because it might happen that as a result of operation (2), one or more periods (previously found to have a tight overtime schedule) will have some unutilized overtime. Consequently, there is a possibility of allocating the remaining subcontracting (if any) to the unutilized overtime of those periods. If after operation (1) no subcontracting remains, operation (3) is not attempted. Step 3 may end with one or more of the following results:

(i) All subcontracting is allocated.

(ii) Some subcontracting may still be present.

(iii) All overtime is allocated.

(iv) Some overtime may still be present.

Records are maintained to denote each of these situations.

Step 4: Repeat Step 2 once more.

Before proceeding further, two special situations that might occur during the execution phase of Step 3 will be specifically mentioned. The first one is shown in Figure 11. Assume that subcontracting in periods eight and nine cannot be allocated to unutilized regular time (or overtime) of periods four through five. Also, assume that subcontracting of period three can be allocated to regular time of period one. Note that there is a discontinuity in the occurrence of TSUB. Appropriate codes have been developed to handle this situation.

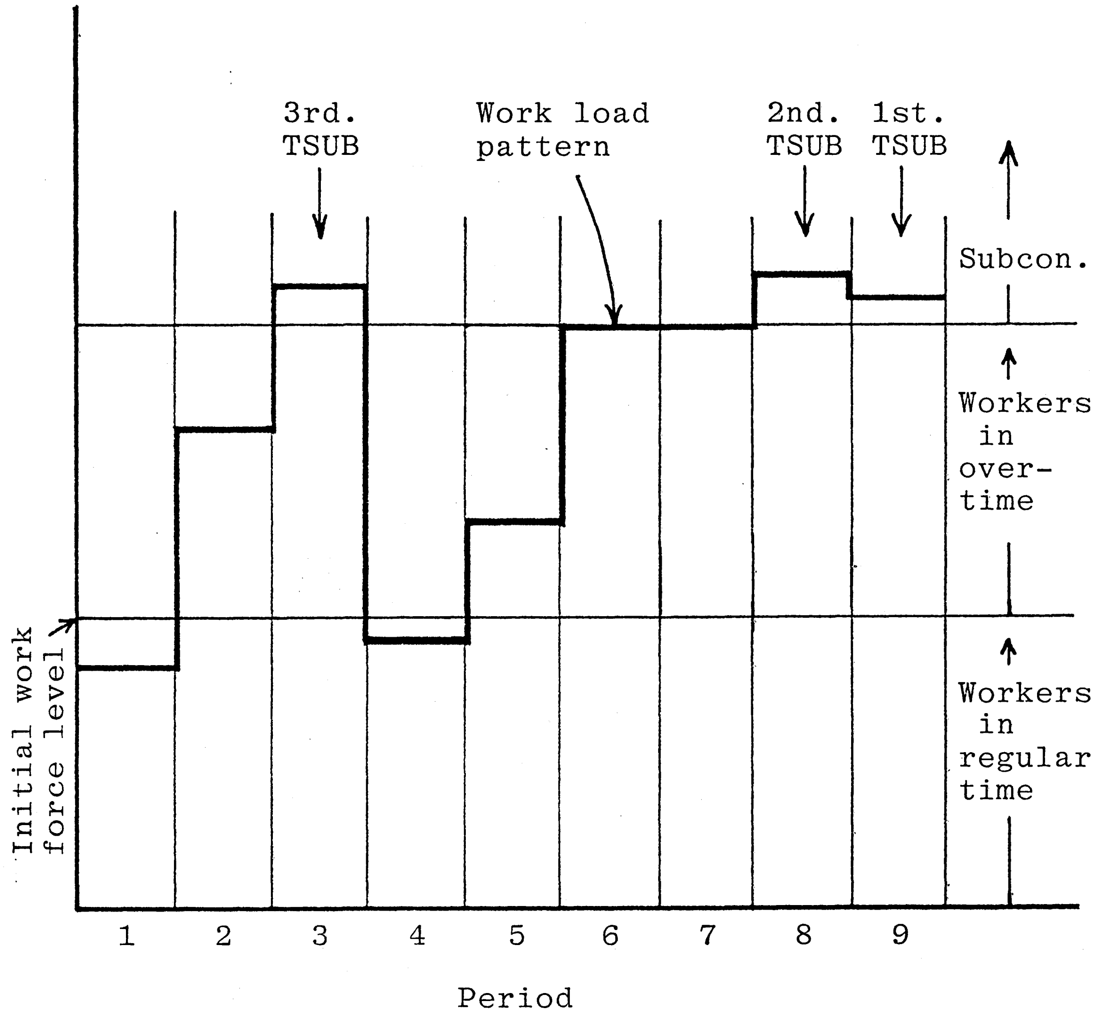


Figure 11. Discontinuity in the Occurrence of TSUB

Another special situation is depicted in Figure 12. Assume that the shaded areas above and below the line PQ are equal and that the allocation of subcontracting occurred after operation (3) in Step 3. In this case, the subcontracting amount of periods six and seven (starting with period six) is required to be shifted to fill the shaded area above the line PQ (on a last period first basis), because if inventory has to be carried, it should be done through the minimum number of periods. This shifting operation is done through a subroutine named LDADJS.

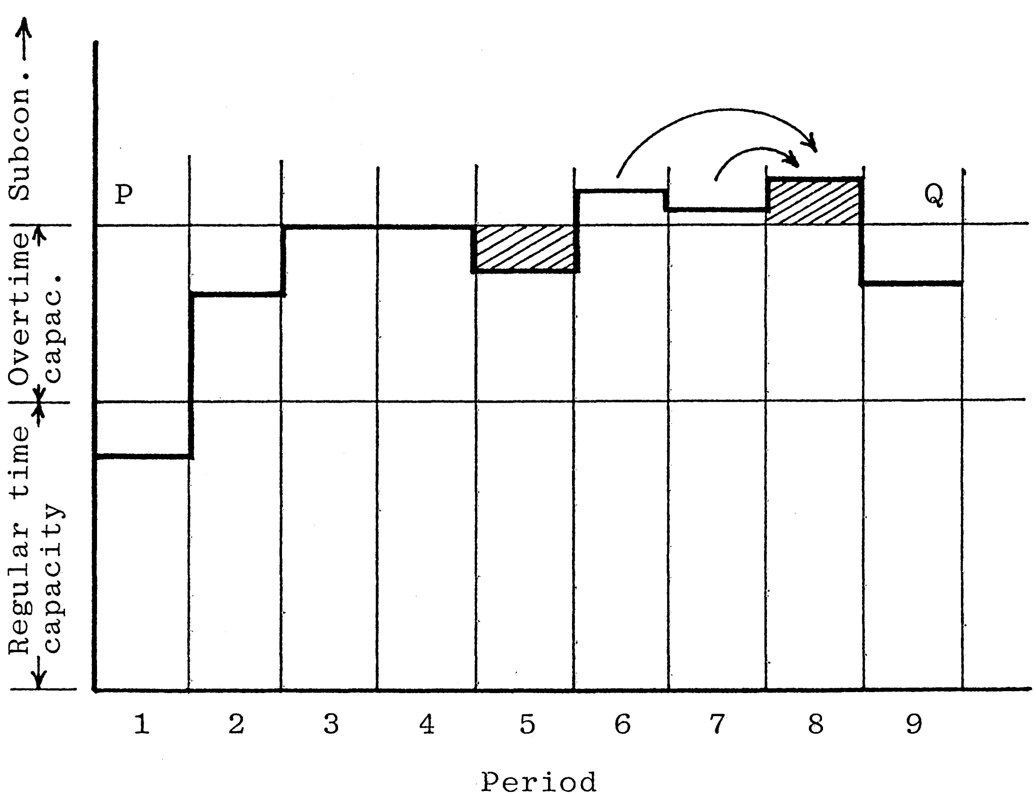


Figure 12. Adjustment of Subcontracting Amount After Step 3

These two situations are mentioned here to help the reader have a better understanding of the technique and the computer codes given in Appendix C.

The reason for repeating Step 2 in Step 4 is that there can be cases where hiring and firing are less expensive and the manager is not constrained to the amount of hiring or firing in any period. In that case, the model may consist of hiring and firing options (no overtime and no subcontracting). This feature cannot be incorporated in the model unless the rules (pages 76 through 78) are applied prior to obtaining a modified work load pattern. After the execution of Step 4, some of the options (identified in Step 2) might change. For example, if it is found that Step 2 indicates firing in a period whereas Step 4 indicates hiring in the same period, then both firing and hiring are included in that period.

In short, stage 1 ends with the identification of different options in all periods. During the execution of this stage, the information regarding these options is stored in different arrays. The arrays used for this purpose are:

NHFIR1(.) = stores codes for hiring or firing. If the code is for firing, it also stores which class might be fired.

HRFIR2(t) = stores maximum quantity likely to be hired or fired in period t. Initially this is roughly equal to the work force equivalent of the difference between demands in periods t and t-1.

IDLE(t) = stores the code for idle time. If the code indicates idle time, HRFIR2(t) is used to set the initial value for idle workers in period t.

NOVBAR(t) = stores the code for overtime. If overtime is likely, it also stores the class number through which overtime might extend

SBCNBR(t) = stores the code for subcontracting. If subcontracting is likely, it also stores the maximum amount of subcontracting likely.

Consider the case of hiring and firing for which two arrays, NHFIR1 and HRFIR2, have been used. The structures of these arrays are shown in Figure 13. For a particular period t, NHFIR1 reserves two cells, and HRFIR2 reserves one cell. The first cell of NHFIR1 stores the code for hiring and firing as described in Figure 13. The second cell stores the number of classes in which firing may exist. The second cell is checked only if the first cell contains a code for firing. The t-th cell of the array HRFIR2 stores the maximum amount that can be hired or fired (depending on the code of NHFIR1). As an example, with the present values of the two cells of NHFIR1 and one cell of HRFIR2 at the t-th period, the option of firing will be included, this firing will extend up to the class number two (the workers of the lowest skill), and the maximum amount of firing (the right-hand side of the constraint (6.14)) could be 7.5.

These codes and/or values stored in different arrays are then used to identify the constraints and/or variables to be included in (or excluded from) the model.

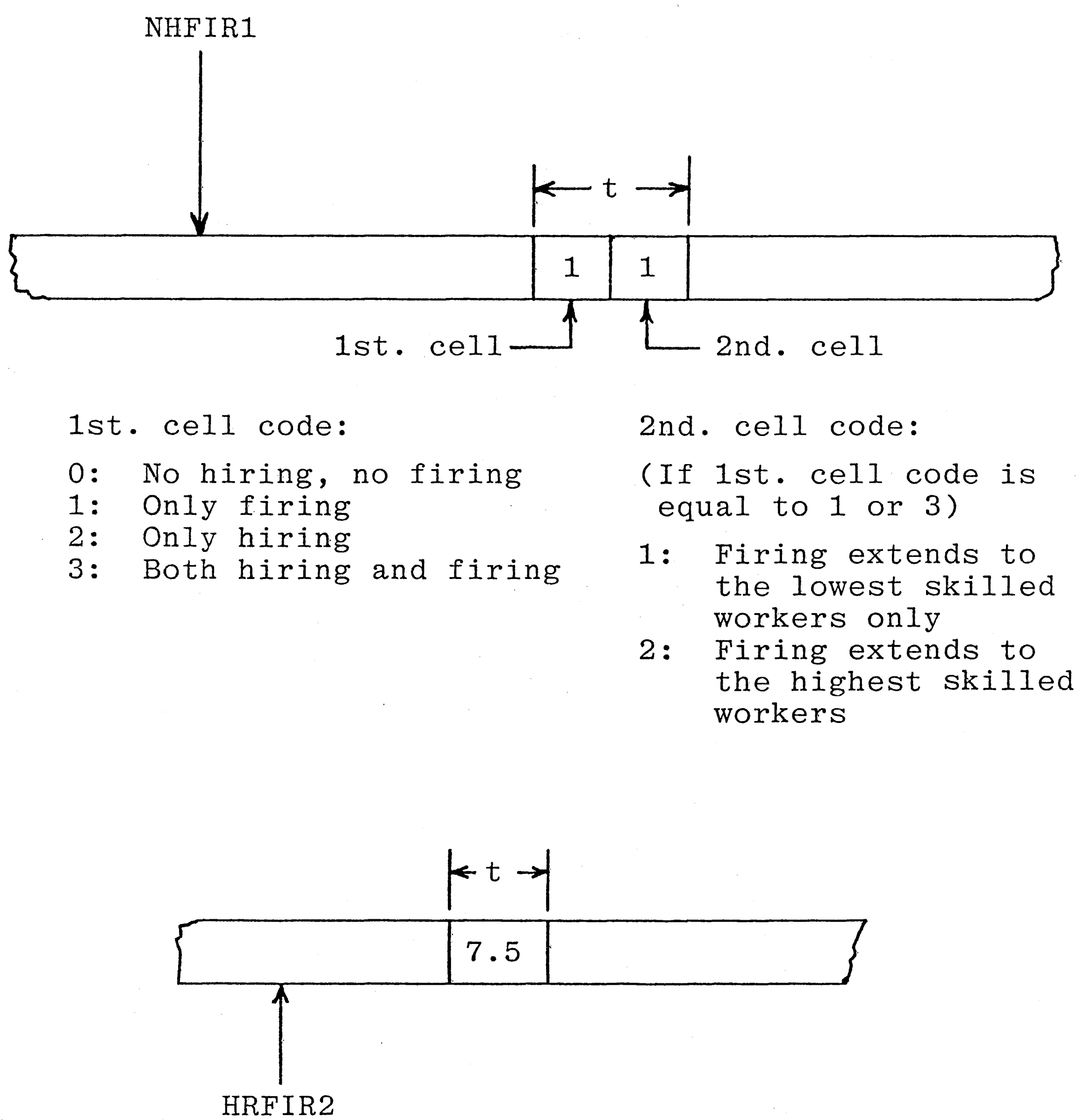


Figure 13. A Part of the Linking System to Identify the Options to be Included

Stage 2:

In this stage the matrix A and the objective functions are constructed with the help of the information obtained in Stage 1. A very efficient linking mechanism is required to keep track of each constraint and what it represents because in this case (the working model), the location of a particular constraint in the matrix A is not fixed as in the case of the fundamental model expressed by the equations (6.7) through (6.16). Some of the constraints may not be present. The same thing is true for the variables. In order to obtain a consistent model, the following things are done. First, the constraints and the variables (for a period) for the fundamental model are arranged as shown in Figure 14. Next, in order to keep record of which constraints and variables are retained in the working model, the following variables and arrays are defined.

ICNTHR = total number of hiring possible throughout the horizon.

ICNSHR(I) = the constraint number corresponding to the constraint for the I-th possible occurrence of hiring.

IVARHR(I) = the variable number corresponding to the variable for the I-th possible occurrence of hiring.

IPRDHR(I) = the period corresponding to the constraint ICNSHR(I).

IDOLHR(I) = the dual variable corresponding to the constraint ICNSHR(I).

Con- straints	Variables ⟶											
↓	P_{rt}	W^1_{rt}	W^2_{rt}	I_t	W_{ht}	W_{ft}	V_t	W^2_{it}	W^1_{it}	P^1_{ot}	P^2_{ot}	P_{st}
(6.8)												
(6.9)												
(6.10)												
(6.13)												
(6.14)												
(6.15)												
(6.12)												
(6.11)												

Constraint (6.7) occupies the first row of the system matrix, A.

Figure 14. Arrangement of the constraints and Variables for a Period

Similar variables and arrays are required for firing, overtime, etc. In some cases, however, all these variables are not required. For example, subcontracting needs only ICNTSB, IVARSB(I), and IPRDSB(I). While the matrix A is constructed (through subroutine STRCTR) containing the elements of a constraint, the information about this constraint, as well as the variables which constitute the constraint, is stored in the variables and arrays described above. Therefore, in the decision making-phase, whenever any information about a type of constraint is required, it can be easily retrieved.

It is obvious that the model obtained by applying the rules described earlier is dynamic in nature. It includes only those constraints which are appropriate for the situation involved. Also, the operations manager has complete freedom for inclusion of any options he thinks suitable for his firm. This can be done (while running the program interactively) by specifying the codes to be stored in NHFIR1(.), NOVBAR(t), etc., for the desired periods.
The performance of the resulting model with respect to the accuracy of the solution and reduction of the computational time will be discussed in the next chapter.

CHAPTER VIII

ANALYSIS OF RESULTS

Introduction

In this chapter the results obtained from the new model are analyzed to validate the new model and the new technique. Also numerical examples are furnished to demonstrate how better alternatives can be generated from the knowledge obtained from the interaction of an objective function with other(s), or from the interaction of an objective function with other variables. At the end of this chapter, possible extensions to this research are mentioned.

Before entering into the detailed analysis of the results, a few related items will be discussed. Note that subcontracting is one of the options of production smoothing. But in this model, no objective function regarding the minimization of subcontracting has been included. Consequently, in some cases, the amount of subcontracting might become very large to keep the costs due to other production smoothing options low. For this reason, whenever subcontracting appears to be a feasible option, an objective function regarding the minimization of the subcontracting cost has to be included in the model. This can be done with no difficulty. However, in the examples given in this chapter, the subcontracting quantity is constrained at zero by making the unit subcontracting cost very high. This is done for ease of comparison of the new model with two other models that do not include subcontracting.

Another point that needs to be mentioned concerns a modification of some of the objective functions. This is described in the following section.

A Modification

One major requirement of aggregate production planning is that hiring and firing of workers in any period cannot take place simultaneously. In all linear aggregate production planning models, this condition has been met by the following constraints:

$$W_t = W_{t-1} + W_{ht} - W_{ft} \tag{8.1}$$

In this case, the columns (of the matrix A defined earlier) associated with W_{ht} and W_{ft} are not linearly independent. Therefore, if any one of these two variables remains in the basis at the positive level, the other will remain as a nonbasic variable. But the situation is different in the case of the new model because of the following constraints:

$$\begin{aligned} W_{ht} &\le \overline{W}_{ht} \\ W_{ft} &\le \overline{W}_{ft} \end{aligned} \tag{8.2}$$

Note that in this case, the columns associated with W_{ht} and W_{ft} are linearly independent. So, both of these variables may remain in the basis at positive levels. This fact was observed while optimizing the functions of overtime cost, inventory cost, and idle time. From now on, these three functions will be called TYPE1 objective functions. It was also observed that this situation does not occur while optimizing the total cost function and the function associated with hiring cost, firing cost, and regular payroll. From now on, these two objective functions will be called TYPE2 objective functions.

One possible reason for hiring and firing being present in the same period might be that, since in the case of the TYPE1 objective functions there were no hiring and firing cost coefficients terms, the optimization program simply offered an optimal solution, as it does in all linear programming problems. In fact, the problem (of both hiring and firing being simultaneously present in the same period) disappeared when W_{ht} and W_{ft} were included in the TYPE1 objective functions. But since this two terms are not part of those objective functions, the cost coefficients should be negligibly small (almost equal to zero) so that the inclusion of these two terms does not affect the solution. The modified TYPE1 objective functions will be sometimes referred to as 'pseudo objective functions.'

The reader should not be disappointed with the model because of this modification, because it simply prevents both hiring and firing in the same period. The cost coefficients used in this case are \$.2 and \$.1 for hiring and firing respectively. This has virtually no effect on the final solution. For example, if one hires 100 workers, this extra cost will be only $20.0, which is negligible compared to any of the cost components. Besides this, the error terms can be seperated from the pseudo cost functions to get the actual cost values. Example two given later will demonstrate the effect of this term.

Validity of the New Model

In order to validate the new model, the results are first verified through hand computations to check if the results satisfy all specified relations. Specifically, such verification consists of determining (1) that inventory, production, and demand relationships agree; (2) that hiring and firing do not take place in the same period; (3) that overtime and idle time do not take place in the same period; (4) that the production schedule can be met with the capacity of the workers during the available regular time and overtime; (5) that there is no problem with the operating conditions when production is simulated over the horizon with the operating conditions offered by the model; and (6) that the cost figures given by the model agree with those obtained when production is simulated over the horizon with the operating conditions given by the model. An output showing the detail information obtained from a run is given in Appendix D.

Next, the results obtained from this model are compared with those obtained from two other models, namely, the Orrbeck model, and the Khoshnevis model. The reason for selecting the Orrbeck model as a basis for comparison is that the new model very closely resembles the Orrbeck model when only the total cost criterion is considered. The constraints as well as the objective function are linear in this model. The reason for choosing the Khoshnevis model, as another basis for comparison, is that the Khoshnevis model uses the cost structure of the Holt et al. model in which "original" cost components are linear. The comparison with the Khoshnevis model, then, provides a general idea as to how far the results of the new model (using the original linear cost structure) deviate from those obtained from the Khoshnevis model (with approximate nonlinear cost structure).

Comparison with the Orrbeck Model

Orrbeck considers two classes of workers. If a worker is hired at some period, he is considered to belong to the least experienced class during this period; in the next period he is transferred to the most experienced class. In the new model, however, this is not done. Rather, the classification is maintained throughout the planning horizon. The data used by Orrbeck is given below.

Planning horizon = 6 periods;

Initial inventory = 1000;

Initial work force: 200 (experienced);
50 (newly hired);

Productivity:

30 units/man/month (experienced);
25 units/man/month (newly hired);

Demand:

$D_1 = 11{,}000$	$D_4 = 12{,}300$
$D_2 = 11{,}500$	$D_5 = 8{,}400$
$D_3 = 9{,}000$	$D_6 = 9{,}200$

Cost coefficients:

Regular payroll: \$ 450/man/month (experienced);
\$ 400/man/month (newly hired);

Hiring: \$ 200/man;

Firing: \$ 100/man;

Inventory carrying cost: \$ 1.00/period/unit;

Overtime pay: 1.5 times the regular pay;

Maximum overtime duration: 0.5 times regular time.

The details of the results obtained in both the cases are given in Table I. Although the total cost in the case of the new model is lower than that in the case of the Orrbeck model, the comparison is not exact for the following reasons. As mentioned earlier, in the case of the Orrbeck model, the newly hired workers are transferred to the experienced class at the end of the period in which they were hired. So, the productivity of the newly hired workers becomes the same as that of the experienced class at the end of the period they are hired. In the case of the new model, this is not done.

The productivity rates (units/man/month) in two cases are given below.

TABLE I

COMPARISON OF RESULTS OF THE NEW MODEL AND THE ORRBECK MODEL

A: DECISIONS AND PROJECTIONS (NEW MODEL)

Period	Demand	Work force	Production	Hiring, firing	Inventory
1	11,000	358.3	10,750	108.3	750
2	11,500	358.3	10,750	0.0	0
3	9,000	355.0	10,650	3.3*	1,650
4	12,300	355.0	10,650	0.0	0
5	8,400	293.3	8,800	61.7*	400
6	9,200	293.3	8,800	0.0	0

B: COST ANALYSIS OF DECISIONS AND PROJECTIONS (NEW MODEL) IN DOLLARS

Peri.	Payroll	Hiring, Firing	Overtime	Inventory	Total
1	153,333.33	21,666.67	0.00	750.00	175,750.00
2	153,333.33	0.00	0.00	0.00	153,333.33
3	152,000.00	333.33*	0.00	1,650.00	153,983.33
4	152,000.00	0.00	0.00	0.00	152,000.00
5	127,333.33	6,166.67*	0.00	400.00	133,900.00
6	127,333.33	0.00	0.00	0.00	127,333.33
					896,299.99
					≃ 896,300.00

*Firing take place in these periods.

TABLE I (continued)

C: DECISIONS AND PROJECTIONS (ORRBECK MODEL)[$]					
Period	Demand	Work force	Production	Hiring, firing	Inventory
1	11,000	368.2[d]	10,454.4[d]	118.2[d]	454[d]
2	11,500	368.2[d]	11,045.5[d]	0.0	0
3	9,000	355.0	10,650.0	13.2[*d]	1,650
4	12,300	355.0	10,650.0	0.0	0
5	8,400	293.3	8,800.0	61.7[*]	400
6	9,200	293.3	8,800.0	0.0	0

[$]The total cost in this case is $943,080.00.

*Firing take place in these periods.

[d]In these periods, the decisions and projections are different from those obtained in the case of the new model.

Source: Orrbeck, M. G., Schuette, D. R., and Thompson, H. E., "The Effect of Worker Productivity on Production Smoothing," Management Science, Vol. 14, No. 6 (1968), pp. 332-342.

	Experienced	Newly hired
Orrbeck model	30	25
New model	30	30

Another point to note is that when the workers are transferred to the experienced class in the case of the Orrbeck model, they are paid more than the newly hired workers. These two factors, basically, are the causes of the difference between the costs obtained in the two cases.

In order for the comparison to be fair, the operating conditions in both cases have to be the same. The total cost obtained in the case of the new model is adjusted below for this purpose. Note that the two models suggest the same work force, production, and inventory levels from period three through period six. These three quantities differ only in the case of the first and the second periods. The extra cost that has to be incurred by the new model can be computed in the following way:

hiring 9.9 workers (beginning of the first period)	= \$ 200 x 9.9	= \$ 1980.00
firing 9.9 workers (end of the third period)	= \$ 100 x 9.9	= \$ 990.00
payroll (1st period)	= \$ (400 x 9.9 + 50 x 50)	= \$ 6460.00
payroll (2nd period)	= \$ (450 x 9.9+50 x 158.3)	= \$ 12370.00
payroll (3rd period)	= \$ 50 x 155	= \$ 7750.00
payroll (4th period)	= \$ 50 x 155	= \$ 7750.00
payroll (5th period)	= \$ 50 x 93.3	= \$ 4665.00
payroll (6th period)	= \$ 50 x 93.3	= \$ 4665.00
Subtotal		= \$ 46630.00
Less extra inventory cost	= \$ 750.00 - \$ 454.5	= \$ 295.50
Total		= \$ 46334.50

Therefore, if the conditions of the two models would be exactly same, the new model would operate at a 'maximum' cost equal to \$ 942634.50 (\$ 896300.00 + \$ 46334.50). This figure is \$ 445.50 (0.04%) lower compared to the Orrbeck model. The reason for using the word 'maximum' is that the additional cost has been found by making direct equivalence between the two models. Although this solution is optimal in the case of the Orrbeck model, this may not be optimal for the new model, because the adjusted solution has not been found through any optimization procedure. In other words, the difference could be more than 0.04%.

<u>Comparison With the Khoshnevis Model</u>

The raw data originally used by Holt et al. are:

Planning horizon: 10 periods

Initial inventory: 263

Initial work force: 81

Worker productivity: 5.67 units/man/month

Demand:

$D_1 = 430$	$D_6 = 375$
$D_2 = 447$	$D_7 = 292$
$D_3 = 440$	$D_8 = 458$
$D_4 = 316$	$D_9 = 400$
$D_5 = 397$	$D_{10} = 350$

Cost coefficients:

Regular payroll: \$ 340/man/month;

Hiring: \$ 180/man;

Firing: \$ 360/man;

Inventory carrying cost: \$ 20/unit/month;

Overtime cost: 1.5 times the regular pay.

Maximum overtime duration: 0.5 times the regular time.

The details of the results in the case of the new model are given in Table II. The results obtained in the case of the Khoshnevis model are given in Table III. The results show that the total cost in the case of the new model is 2.3 per cent lower than that obtained from the Khoshnevis model. Although this result shows that the performance of the new model is better than that of the Khoshnevis model, the author does not prefer to consider the Khoshnevis model (equivalently the Holt et al. model) as a basis for comparison because of the following reasons:

1. The cost function of the Holt et al. model is only approximate [15].
2. The Holt et al. model does not consider any bounds on the variables. Holt et al. explain that they did not need to place such bounds on the variables for the type of data they handled [15]. A logical conclusion is that the Holt et al. model might not offer a feasible schedule for some other data. It may be noted that in the case of the Holt et al. model, the relationship among production, inventory, and demand is the only requirement.

TABLE II

RESULTS OF THE NEW MODEL USING THE DATA OF THE KHOSHNEVIS MODEL

A: DECISIONS AND PROJECTIONS

Period	Demand	Work Force	Production	Hiring, Firing	Inventory
1	430	62.0	351.33	19.0*	184.3
2	447	62.0	351.33	0.0	88.7
3	440	62.0	351.33	0.0	0.0
4	316	62.0	351.33	0.0	35.3
5	397	65.0	368.33	3.0	6.7
6	375	65.0	368.33	0.0	0.0
7	292	65.0	368.33	0.0	76.3
8	458	65.0	381.63	0.0	0.0
9	400	65.0	400.00	0.0	0.0
10	350	65.0	350.00	0.0	0.0

B. COST ANALYSIS OF DECISIONS AND PROJECTIONS ($)

Peri.	Payroll	Hiring, Firing	Overtime	Inventory	Total
1	21,067.6	6,853.1*	0.0	3,686.7	31,607.4
2	21,067.6	0.0	0.0	1,773.3	22,840.9
3	21,067.6	0.0	0.0	0.0	21,067.6
4	21,067.6	0.0	0.0	706.7	21,774.3
5	22,087.0	539.7	0.0	133.3	22,760.0
6	22,087.0	0.0	0.0	0.0	22,087.0
7	22,087.0	0.0	0.0	1,526.7	23,613.7
8	22,087.0	0.0	1,199.3	0.0	23,286.3
9	22,087.0	0.0	2,848.3	0.0	24,935.3
10	22,087.0	0.0	0.0	0.0	22,087.0

*Firing takes place in this period.

TABLE III

RESULTS OF THE KHOSHNEVIS MODEL

A: DECISIONS AND PROJECTIONS

Month	Demand	Work force	Production	Inventory	Avg. Product.
1	430	77.7	470.5	303.5	5.67
2	447	74.3	444.1	300.6	5.67
3	440	70.9	417.1	277.7	5.67
4	316	67.7	381.7	343.4	5.67
5	397	65.1	376.2	322.5	5.67
6	375	62.7	363.8	311.4	5.67
7	292	60.7	348.9	368.3	5.67
8	458	59.0	359.4	269.7	5.67
9	400	57.4	329.3	199.0	5.67
10	350	56.1	272.2	121.2	5.67

B: COST ANALYSIS OF DECISIONS AND PROJECTIONS ($)

Mo.	Payroll	Hiring, Firing	Overtime	Invento.	Total
1	26,406.07	715.19	2,447.78	22.45	29,591.40
2	25,247.18	747.03	1,978.09	31.04	28,003.34
3	24,105.72	724.73	1,476.17	147.88	26,454.50
4	23,029.24	644.56	511.47	45.05	24,230.33
5	22,122.18	457.64	986.83	0.53	23,567.18
6	21,327.44	351.32	1,015.87	6.13	22,700.76
7	20,639.67	263.11	810.97	192.46	21,906.21
8	20,068.63	181.38	1,936.16	208.91	22,395.08
9	19,509.80	173.70	740.77	1,207.64	21,631.91
10	19,080.42	102.55	-1,408.60	3,259.04	21,033.41
					241,514.22

Source: Khoshnevis, B., "Aggregate Production Planning Models Incorporating Dynamic Productivity," a dissertation submitted to the Faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy, December, 1979.

3. The Holt et al. model gives negative overtime cost which is beyond common sense.

However, on the basis that the results (in the case of the new model) are valid, and very close to the Orrbeck model, it may be inferred that the new model can be considered a valid model. Since the model has been proved to be valid, from now on, the discussions will be based on the results obtained from the new model.

Validity of the Concepts Developed

It has been mentioned earlier that the generation of new alternatives and the gathering of information about them are fundamental to the decision-making process. It was further claimed that it is possible to get information about an alternative before it is generated. It has also been mentioned that it is possible to use the information obtained from the interaction of the decision variables with an objective function, and from the interaction of one of the objective functions with the other objective function(s). Two numerical examples are furnished below to demonstrate these points.

The first example (Table IV) shows how a better alternative can be found from the information generated from the interaction of one of the objective functions with the others. Eleven trials have been made. At the end of each trial the dual variables (shadow price) corresponding to different objective functions were observed. The right-hand

TABLE IV

GENERATION OF NEW ALTERNATIVES FROM THE KNOWLEDGE OF SHADOW PRICE (INTERACTION AMONG OBJECTIVE FUNCTIONS)

TRIAL NUMBER	OBJECTIVE FUNCTION UNDER IMPROVEMENT	OBJECTIVE RELAXED (AMOUNT RELAXED)	OBJECTIVE FUNCTION VALUES (CORRESPONDING SHADOW PRICES)					EXPECTED IMPROVEMENT	ACTUAL IMPROVEMENT	COMMENTS
			HIRING + FIRING + REG. PAYROLL	OVERTIME COST	INVENTORY COST	IDLE WORKER	TOTAL PROD. COST			
1	TOTAL COST	-	893,500.00	0.00	2,800.00	0.00	896,300.00	-	-	Initial solution
2	HIRING + FIRING + REGULAR PAYROLL (HFR)	-	893,500.00 (0.00)	0.00 (1.17)	2,800.00 (10.00)	0.00 (50.00)	896,300.00 (0.00)	-	-	Starting point with HFR
3		OVERTIME (3.00)	893,500.00 (0.00)	0.00 (0.00)	2,800.00 (0.00)	0.00 (0.00)	896,300.00 (15.71)	3.51	0.00	Total cost function is binding
4		TOTAL COST (3.51)	893,497.25 (0.00)	3.00 (0.92)	2,800.00 (0.00)	0.00 (0.00)	896,300.25 (0.00)	55.14	2.76	Action of 3rd trial is effective
5		OVERTIME (17.00)	893,481.67 (0.00)	20.00 (0.92)	2,800.00 (0.00)	0.00 (0.00)	896,301.67 (0.00)	15.64	15.64	
6		OVERTIME (20.00)	893,463.33 (0.00)	40.00 (0.92)	2,800.00 (0.00)	0.00 (0.00)	896,303.33 (0.00)	18.40	18.40	
7		OVERTIME (40.00)	893,444.84 (0.00)	60.17 (0.00)	2,798.50 (0.00)	0.00 (0.00)	896,303.51 (15.71)	36.80	18.56	Total cost is binding; Only 20.17 effective
8		TOTAL COST (6.49)	893,426.67 (0.00)	80.00 (0.92)	2,800.00 (0.00)	0.00 (0.00)	896,306.67 (0.00)	101.96	18.23	Remaining overtime (19.83) is effective
9	INVENTORY COST	-	893,426.67 (0.10)	80.00 (0.12)	2,798.00 (0.00)	0.00 (5.20)	896,304.67 (0.00)	-	-	Starting point with inventory cost
10		IDLE WORKER (1.00)	893,426.67 (0.15)	80.00 (0.16)	2,798.00 (0.00)	0.00 (0.00)	896,304.67 (0.00)	5.10	0.00	Idle worker constraint redundant
11		OVERTIME (1.00)	893,426.67 (0.15)	81.00 (0.16)	2,797.84 (0.00)	0.01 (0.00)	896,305.51 (0.00)	0.16	0.16	

Values in this column have been found by multiplying SHADOW PRICE AND AMOUNT of resource relaxed as shown by two-way arrows in two cases.

side of an objective function having a shadow price greater than zero was relaxed and the model was rerun. Table IV is self explanatory and describes the improvement process in detail. The function to be improved, the function to be relaxed, and the amount to be relaxed were chosen arbitrarily. The results show that the objective function under the process of improvement has been improved by the amount equal to the product of the shadow price and the amount relaxed except in some trials (three, four,and ten). These cases need explanation.

The reason for those trials not giving the expected improvement is that at the optimal point, several constraints were simultaneously binding. However, a few of them are likely to be redundant. When this occurs, the objective function may not improve, because the other constraints (in this case, constraints formed by one or more of the objective functions) may not allow the feasible space to be expanded. The reader may refer to Appendix A for proof.

The result of trial four needs further explanation. Note that in this case the improvement is only \$2.76 instead of \$55.14. This may be explained by saying that in this case, the constraint formed by the total cost function was redundant and as such, had no effect (refer to the results of the third trial). So, when the total cost constraint was relaxed, the improvement due to overtime cost relaxation, made in the third trial, came into effect which gave \$ 2.76 (= \$0.92 x 3). The reason for not getting \$3.51 is that the duals at one extreme point may not be the same as those at other extreme points. This point is also demonstrated in Appendix A. A point to remember while investigating the results of the third trial is that although a dual variable other than zero means that the corresponding constraint is binding, a binding constraint does not necessarily have to have a dual variable other than zero.

The second example (Table V) shows how the knowledge of the shadow price for a decision variable can help improve an objective function.

It is worth noting that in the case of the first example, an increment of total cost by $ 5.51 has reduced the hiring, firing and regular payroll cost by $ 73.33. This, of course, resulted in an increase of overtime cost by $'81.10. But a little overtime in this case, is preferable to increased hiring, firing, and regular payroll. The two examples given above validate the claims made earlier.

Validity of the New Technique

In this section the validity of the new technique is proven. The model is solved with the total cost as the objective function with two sets of data. For each data set, the new model was run twice; first, without utilizing the technique, next, utilizing the technique. The results obtained in these cases are displayed in Table VI. The table shows that in the case of data of the Orrbeck model, the two results are exactly equal. But in the case of data

TABLE V

GENERATION OF NEW ALTERNATIVES FROM THE KNOWLEDGE OF SHADOW PRICE (INTERACTION OF A DECISION VARIABLE WITH AN OBJECTIVE FUNCTION)+

TRIAL NUMBER	OBJCT. FUNC. UNDER IMPROV.	OPTION (PERIOD) (SHADOW) (AMOUNT)	OBJECTIVE FUNCTION VALUES					EXPECTED IMPRVMNT.	ACTUAL IMPRV. (PSEUDO)	COMMENTS
			HIR. + FIR. + RG. PAY	OVERTIME COST (PSEUDO)	INVENTORY COST	IDLE WORKER	TOTAL COST			
1	OVERTIME COST	HIRING (1) (257.31) (-)	893,426.67	3,491.43 (3,521.85)	2,574.57	0.0	899,492.67	-	-	Initial solution
2	OVERTIME COST	HIRING (1) (257.31)* (1.00)*	893,426.67	3,234.28 (3,264.53)	2,574.57	0.0	899,235.52	257.31	257.15 (257.32)	Difference between actual and pseudo improvement is only 0.17

Value in this column has been found by multiplying SHADOW PRICE and AMOUNT RELAXED as shown by marking the two quantities with *.

+ Data used in Example one and Example two are different.

TABLE VI

COMPARISON OF RESULTS WITH AND WITHOUT THE PROPOSED TECHNIQUE

Data used	Items compared	Fundamental model	Working model	Reduction or Difference
Orrbeck (6 period)	Total cost($)	896,300.00	896,300.00	0.00%
	Number of constraints	66	48	18
	Number of Variables	54	36	18
	CPU time (sec.)	61.61	17.25	72.00%
Khoshnevis (10 period)	Total cost($)	236,059.66	236,790.68	+ 0.31%
	Number of constraints	106	76	30
	Number of Variables	90	60	30
	CPU time (sec.)	241.32	60.50	74.93%

of the Holt et al. model there is a very small difference (0.31%) between the two cost figures. The author observes that the values of the decision variables obtained with and without employing the technique are exactly same in the case of data of the Orrbeck model; whereas, in the case of data of the Holt et al. model, most of the decision variables assumed the same values. This indicates that the exactness of solutions found with and without the technique is not due to the existence of alternate optimum solutions.

It may be noted that when the new technique is used to formulate the model, the number of the constraints, the number of the variables, and the CPU time are reduced significantly.

The author observes that when a total cost function is considered, then irrespective of the type of data, the results obtained with and without the technique are very close. But when other objective functions were considered, the differences are found to be greater. For this reason, the technique was not utilized while solving the problems given in Examples one and two. However, the significant reduction in the number of constraints, variables, and CPU time indicates that the technique has tremendous potential in the area of aggregate production planning. Observe that the total reduction of the CPU time can be as high as 75.00%. The technique can be particularly helpful in the area of MCAPP where the same problem has to be solved several times.

The results bear the testimony that the way the technique has been developed, is based on sound logic. However, as there are cases where the results with and without the technique are not 100% equal, the rules used in eliminating the constraints and variables need further development.

Generalization of the Concepts

In this section it will be shown that the decision-making concepts developed in this research are not limited to MCAPP problems only. Rather, these concepts, that is generation of new alternatives and gathering information about them, are keys to any decision-making problem. As such, the concepts developed can be generalized and applied to a general class of problems as explained below.

Consider the following decision-making problem:

$$\begin{aligned} &\text{Minimize} \quad [f_1(X),\ f_2(X),\ \ldots\ ,\ f_K(X)] \\ &\text{Subject to} \quad AX \leq b \end{aligned} \tag{8.3}$$

where $X = [x_1, x_2, \ldots , x_n]$ are n decision variables, $AX \leq b$ are technological constraints associated with the problem, and $f_k(x)$ are K criteria with respect to which the decision has to be made.

One way to solve this type of problem is to apply either goal programming or multiobjective linear programming. Obviously, the decision maker would expect the same difficulties as were experienced in the case of the MCAPP problem. However, it may be noted that the problem stated in (8.3) may be augmented by including the constraints of the form:

$$Y \leq \overline{Y} \tag{8.4}$$

where $Y \in X$, and $\overline{Y}$ are the upper bounds for Y. The reason for considering only a subset of X in (8.4) is twofold. First, the problem will be prevented from becoming too large. Second, the decision maker might not be interested in the change of all variables. Note that with this augmentation, the decision maker can apply the concepts developed earlier. In fact, MCAPP problems are only a subset of the general problem stated in (8.3) and (8.4).

Future Research Directions

It has been mentioned earlier that this research is the first step towards solving a more general MCAPP problem involving a nonlinear cost structure. With this point in mind, the author recommends the following for further research:

1. Incorporating a nonlinear cost structure. Notice that in expression (7.1) the inventory cost structure is considered to be linear. It might be possible to include nonlinear cost functions for different cost components. This inclusion, however, need not change the cost structure of the newly developed model to a nonlinear one if the following recommendations are utilized. The cost coefficients

of the linear model need to be chosen in such a way that at the optimal solution stage these cost coefficients represent those obtained from the nonlinear cost structure (see Figure 15).
In Figure 15, C is the cost for a certain option at the beginning of the solution. Through each iteration of the solution process, the path of successive cost values proceeds toward C' as the cost coefficients of the linear cost structure are changed. Finding a technique to get this convergence is extremely difficult. But the discussions in the next paragraph will show that if this can be done for the first rolling horizon (this may require several iterations), convergence can be maintained within a reasonable accuracy for the subsequent rolling horizons without much difficulty.

2. Using the results obtained in the previous rolling horizon as inputs to the problem of the immediately next rolling horizon. All the aggregate production planning models, developed so far, solve the problem for a rolling horizon, and when the new rolling horizon comes, the problem is solved again ignoring the results obtained in the immediate past rolling horizon. But it appears that the solutions obtained from the immediate past rolling horizon can be of great help. Notice that the demand

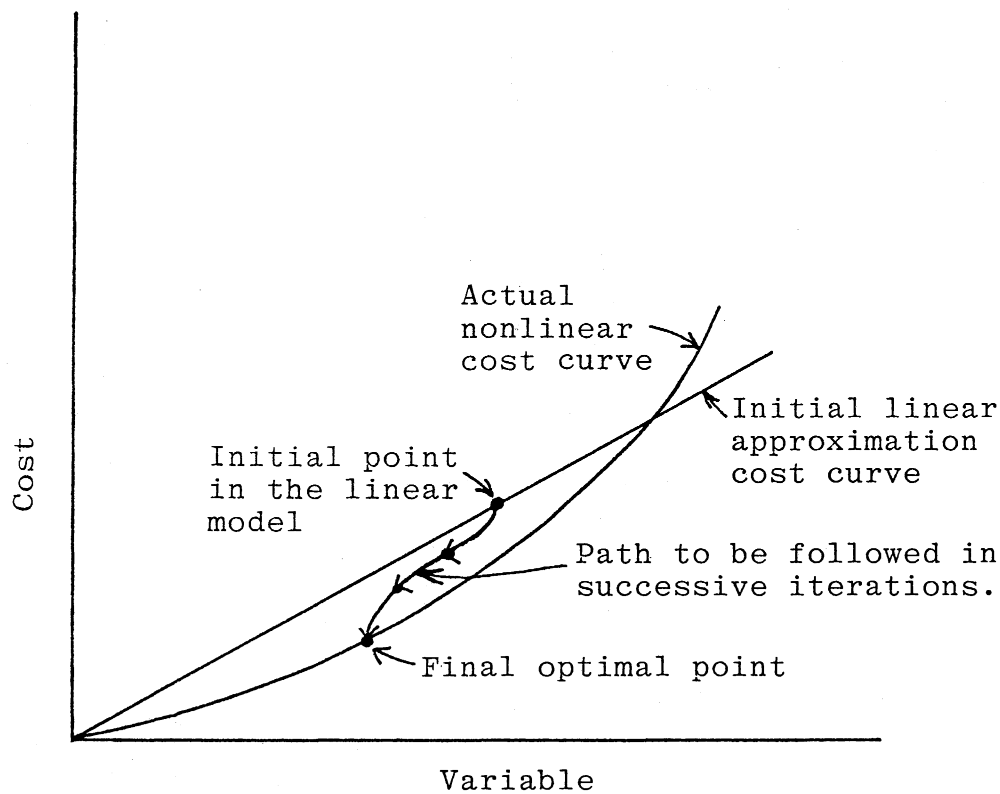


Figure 15. Convergence From Linear Cost Structure to Nonlinear Cost Structure

forecasted at a certain period, in general, will not be significantly different from the forecast made one period later. As such, the solution obtained for a particular period in a rolling horizon is not expected to be significantly different from the corresponding period of the next rolling horizon. It is, therefore, possible to use the information about the operating conditions (the right-hand side of the constraints (6.13)$\sim$(6.16) and the cost coefficients mentioned in part 1 above) of the different periods for a rolling horizon as the inputs for the corresponding periods of the next immediate rolling horizon. This will help reduce the computational time for the subsequent rolling horizons.

3. Including computer graphics capabilities. Today, with the help of computer graphics, it is possible to display different options of production smoothing in different colors for different periods. A graphical display, obviously, will act as a visual aid to the decision maker.

BIBLIOGRAPHY

1. Bitran, G. R., Hass, E. A., and Hax, A. C., "Hierarchical Production Planning: A Two-Stage System," Operations Research, Vol. 30, No. 2(1982), pp. 232-251.

2. Bowman, E. H., "Production Scheduling by the Transportation Method of Linear Programming," Operations Research, Vol. 4, No. 1 (1956), pp. 100-103.

3. Bowman, E. H., "Consistency and Optimality in Managerial Decision Making," Management Science, Vol. 9, No. 2 (1963), pp.310-321.

4. Buffa, E. S., and Miller, J. G., Production-Inventory Systems Planning and Control, Richard D. Irwin, Inc., Homewood, Illinois, 1979.

5. Buffa, E. S., and Toubert, W. H., Production-Inventory Systems Planning and Control, Richard D. Irwin, Inc., Homewood, Illinois, 1972.

6. Ching-Lai, H., and Masud, A. S. M., Multiple Objective Decision Making -- Methods and Applications -- A State-of-the Art Survey, Springer-Verlag, New York, 1979.

7. Conrath, D. W., and Hamilton, W. F., "The Economics of Manpower Pooling," Management Science, Vol. 18, No. 2 (1971), pp. 19-29.

8. Dempster, M. A. H., Fisher, M. L., Jansen, L., Lageweg, B. J., Lenstra, J. K., and Kan, A. H. G. R., "Analytical Evaluation of Hierarchical Planning Systems," Operations Research, Vol. 29, No. 4 (1981), pp. 707-716.

9. Ebert, R. J., "Aggregate Planning with Learning Curve Productivity," Management Science, Vol. 23, No. 2 (1976).

10. Eilon, S., "Five Approaches to Aggregate Production Planning," AIIE Transaction, Vol. 7, No. 2 (1975), pp. 118-131.

11. Goodman, D. A., "A Goal Programming Approach to Aggregate Planning of Production and Workforce," Management Science, Vol. 20, No. 12 (1974), pp. 1569-1575.

12. Hanssman, F., and Hess, S. W., "A Linear Programming Approach to Production and Employment Scheduling," Management Technology, No. 1 (1960), pp. 46-51.

13. Hindelang, T. J., and Hill, J. L., "A New Model for Aggregate Output Planning," OMEGA, The Int. J. of Management Science, Vol. 6, No. 3 (1978), pp. 267-272.

14. Hirshmann, W. V., "Profit From the Learning Curve," Harvard Business Review, Vol. 42 (1964), pp. 125-139.

15. Holt, C. C., Modigliani, F., Muth, J. F., and Simon, H. A., Planning Production Inventories and Work Force, Prentice-Hall, Englewood, Cliffs, N. J., 1960.

16. Jones, C. H., "Parametric Production Planning," Management Science, Vol. 13, No. 11 (1967), pp. 843-866.

17. Khoshnevis, B., "Aggregate Production Planning Models Incorporating Dynamic Productivity," a dissertation submitted to the Faculty of the Graduate College of the Oklahoma State University in Partial fulfillment of the requirements for the degree of Doctor of Philosophy, December, 1979.

18. Lawrence, K. D., and Burbridge, J. J., "A Multiple Goal Linear Programming Model for Coordinated Production and Logistics Planning," Int. J. Prod. Res., Vol. 14, No. 2 (1976), pp. 215-222.

19. Lee, W. B., and Khumawala, B. M., "Simulation Testing of Aggregate Production Planning Models in an Implementation Methodology," Management Science, Vol. 20, No. 6 (1974), pp. 903-911.

20. McGarrah, R. E., "Production Programming," The Journal of Industrial Engineering, Vol. 7, No. 6 (1956), pp. 263-271.

21. Mellichamp, J. M., and Love, R. M., "Production Switching Heuristics for the Aggregate Planning Problem," Management Science, Vol. 24, No. 12 (1978), pp. 1242-1251.

22. Orrbeck, M. G., Schuette, D. R., and Thompson, H. E., "The Effect of Worker Productivity on Production

Smoothing," Management Science, Vol. 14, No. 6 (1968), pp. 332-342.

23. Schwarz, L. B., and Johnson, R. E., An Appraisal of the Emperical Performance of the Linear Decision Rule for Aggregate Planning," Management Science, Vol. 24, No. 8 (1978), pp. 844-849.

24. Shearon, W. T., "A Study of the Aggregate Production Planning Problem, " a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Business Administration, The Colgate Darden Graduate School of Business Administration, University of Virginia, August, 1974.

25. Taubert, W. H., "A Search Decision Rule for the Aggregate Scheduling Problem," Management Science, Vol. 14, No. 6 (1968), pp. 343-358.

26. Zeleny, M., Multiple Criteria Decision Making, McGraw-Hill, Inc., 1982.

APPENDICES

APPENDIX A

EFFECT OF REDUNDANCY IN LINEAR PROGRAMMING

It is a well known fact that at the optimal stage of linear programming, the dual variable corresponding to a nonnegativity constraint is a measure of change of the objective function. Specifically, the objective function will be improved by an amount equal to the value of the dual variable, if the amount of the resource corresponding to the nonnegativity constraint is increased by one unit. Although what has been said above is mathematically true, it may not be possible to get any improvement even if the resource is increased. This can be easily seen from the figure on the next page. In this figure, the optimum solution is at point O. Since all the constraints are binding, the duals corresponding to some of these constraints are expected to have values other than zero. This fact will be illustrated. Assume that the dual variables corresponding to the constraints one and three are not zero. Then an increment of the right-hand side of either of these two constraints should offer an improvement. But a careful observation will reveal that even when any one of these constraints is relaxed, the desired improvement cannot be achieved, because the other two constraints do not permit the solution space to be expanded.

The fact that the objective function value will remain stationary in the new configuration (with the change in the right-hand side value of any one of the constraints) is not the only point. Since the right-hand side vector of the original linear programming problem will change, the dual

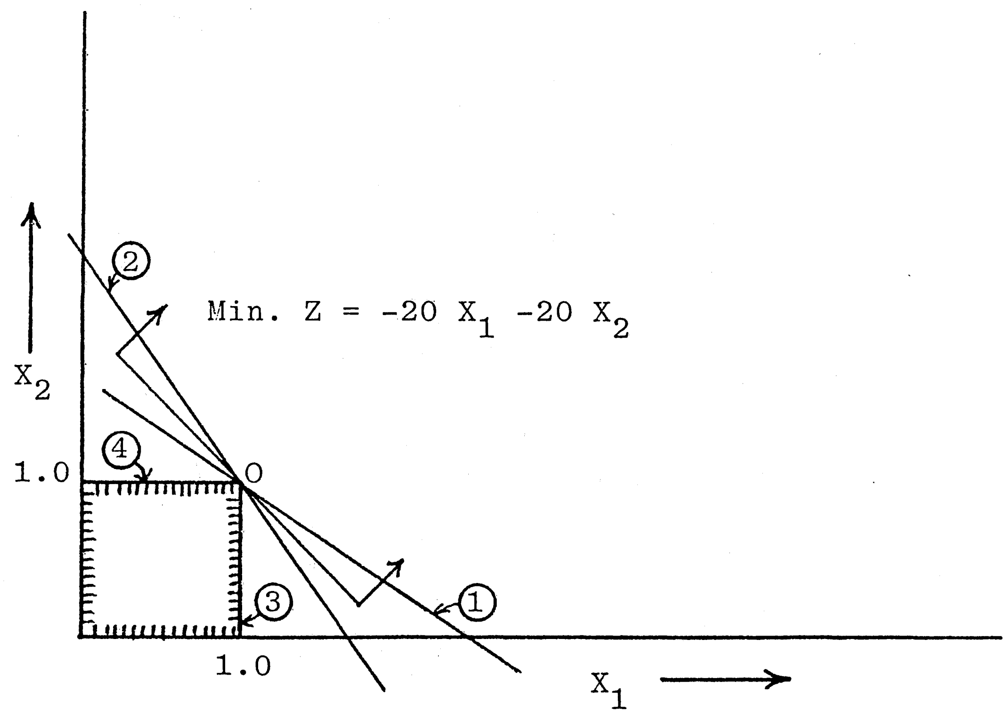


Figure 16. Effect of Redundancy on Improvement of Function Based on Shadow Price

solution is also expected to change to satisfy the complementary slackness conditions.

These two things have been demonstrated below with the help of a very simple example.

$$
\begin{aligned}
\text{Minimize} \quad & z = -\,20\,X_1 - 20\,X_2 \\
\text{Subject to} \quad & 2\,X_1 + 3\,X_2 \le b_1\ (5) \\
& 3\,X_1 + 2\,X_2 \le b_2\ (5) \\
& X_1 \le b_3\ (1) \\
& X_2 \le b_4\ (1) \\
& X_1\,,\ X_2 \ge 0
\end{aligned}
$$

The values in the parentheses on the right-hand sides of the constraints indicate the initial values. The b values were changed step by step. The results of these changes are summarized in Table VII. Table VII verifies what has been said above.

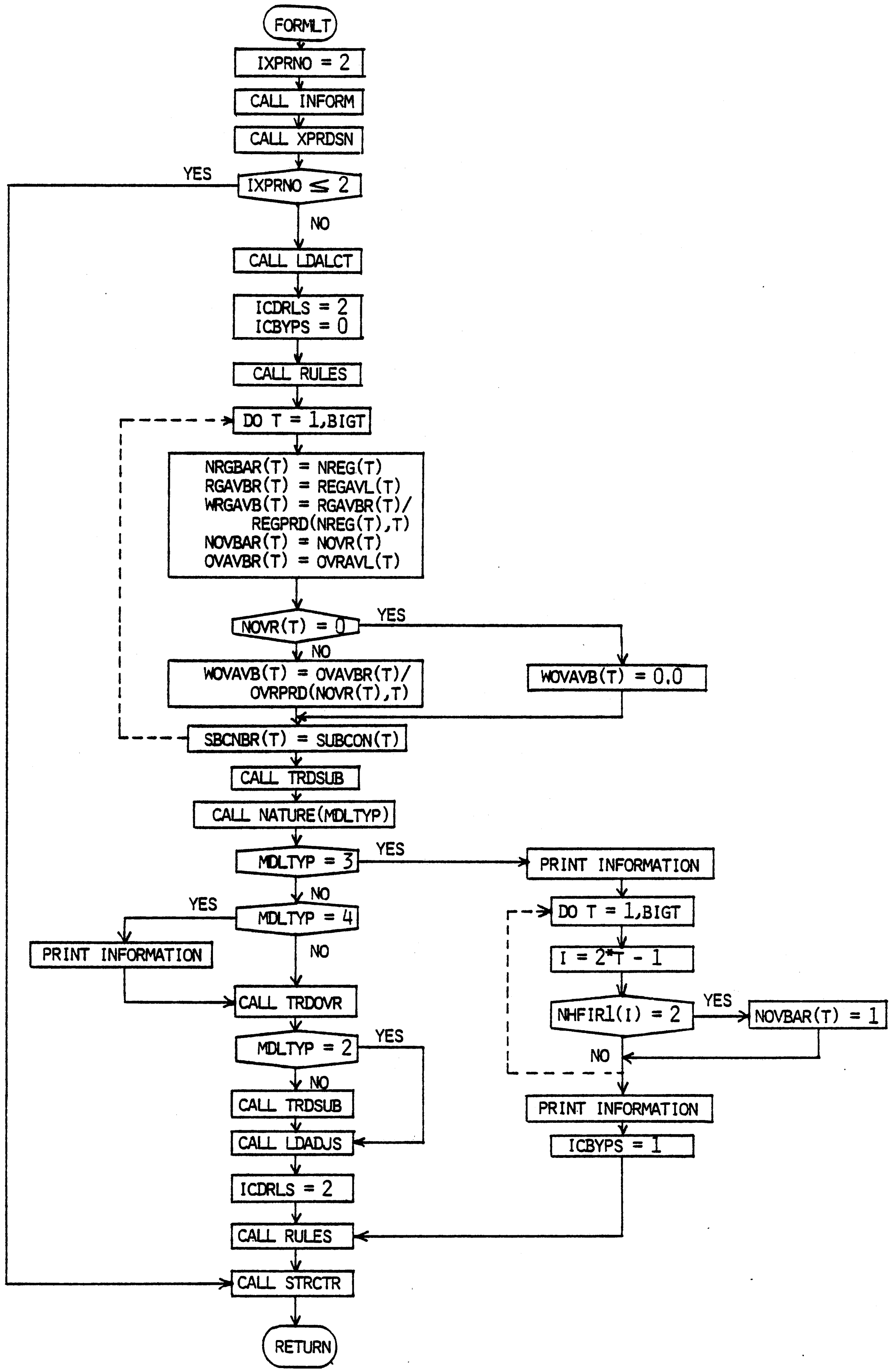
TABLE VII

EFFECT OF REDUNDANCY

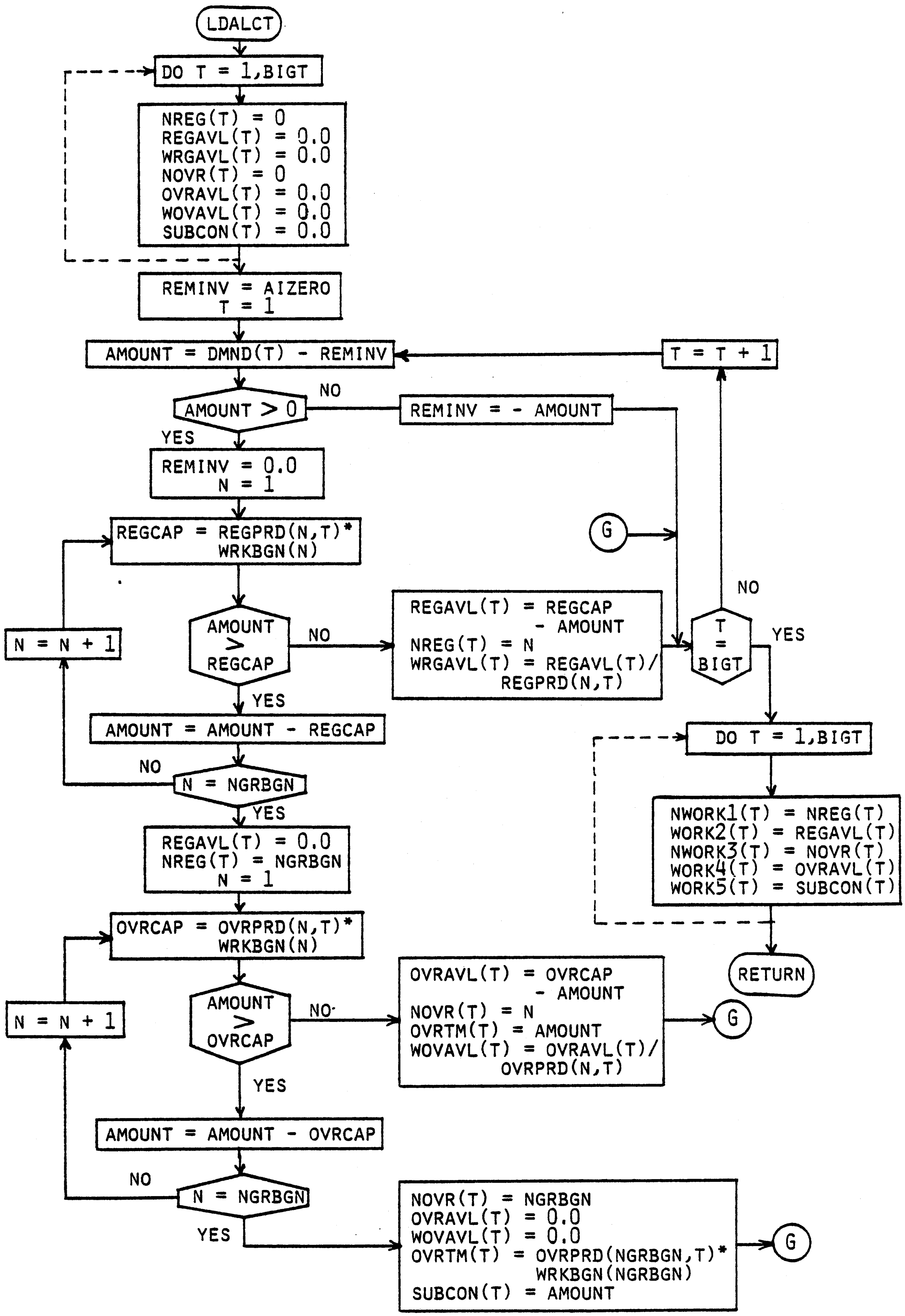
Trial No.	Relaxed		Shadow price for constraint number				Func. value	Change		Remarks
	Constr. No.	Amount	1	2	3	4		Expct.	Actual	
1	-	-	-6.67	0.00	-6.67	0.00	-40.00	-	-	Initial Soln.
2	1	3.0	0.00	-6.67	0.00	-6.67	-40.00	-20.00	0.00	Relaxed constraint is redundant
3	2	3.0	0.00	0.00	-20.00	-20.00	-40.00	-20.00	0.00	Relaxed constraint is redundant
4	3	0.1	0.00	0.00	-20.00	-20.00	-42.00	-2.00	-2.00	
5	3	0.1	0.00	0.00	-20.00	-20.00	-44.00	-2.00	-2.00	
6	3	0.2	0.00	0.00	-20.00	-20.00	-48.00	-4.00	-4.00	
7	4	0.4	0.00	0.00	-20.00	-20.00	-56.00	-8.00	-8.00	
8	3 4	0.2 0.2	-4.00	-4.00	0.00	0.00	-64.00	-8.00	-8.00	
9	1	2.0	0.00	-6.67	0.00	-6.67	-64.00	-8.00	0.00	Relaxed constraint is redundant

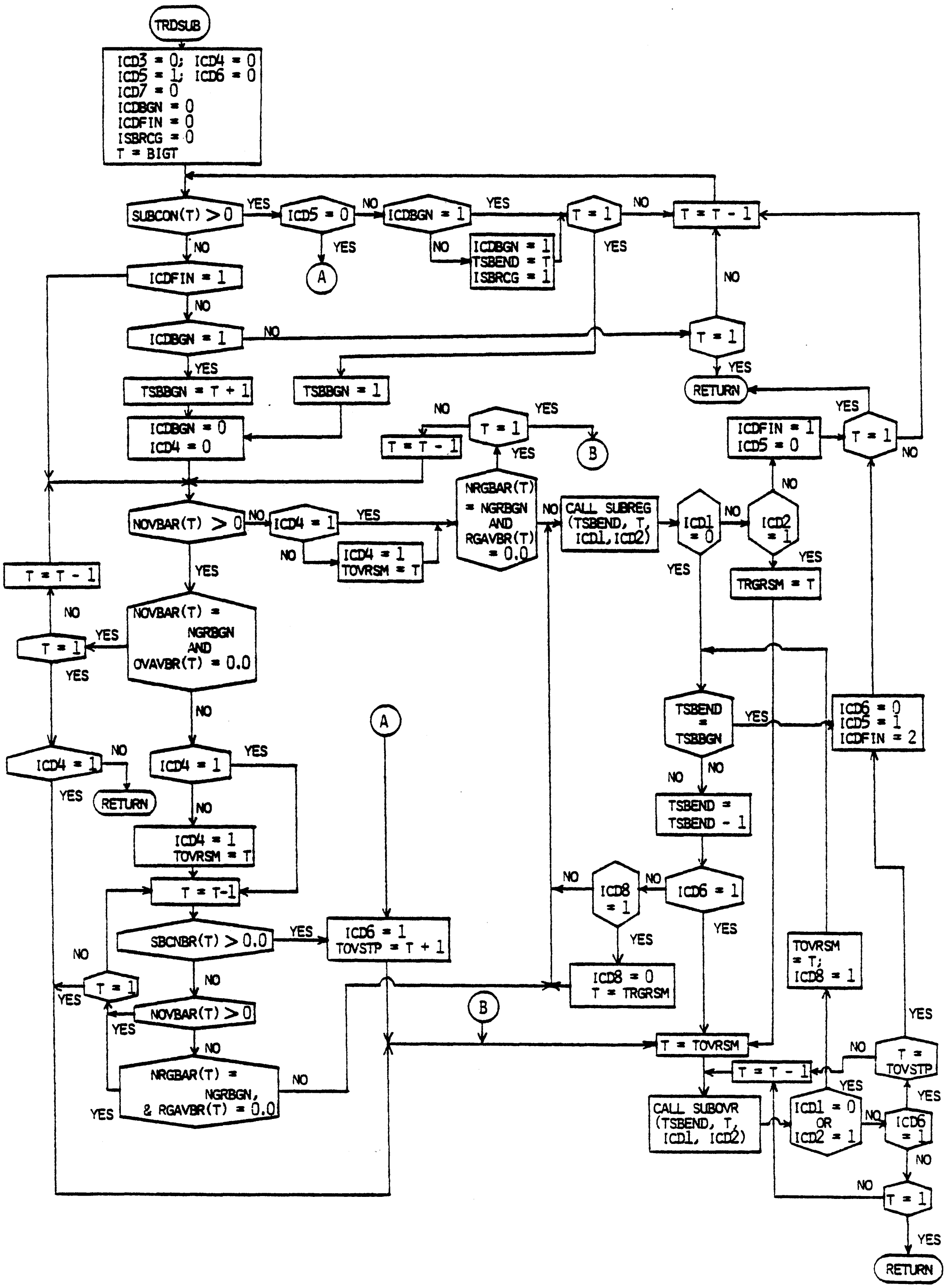
APPENDIX B

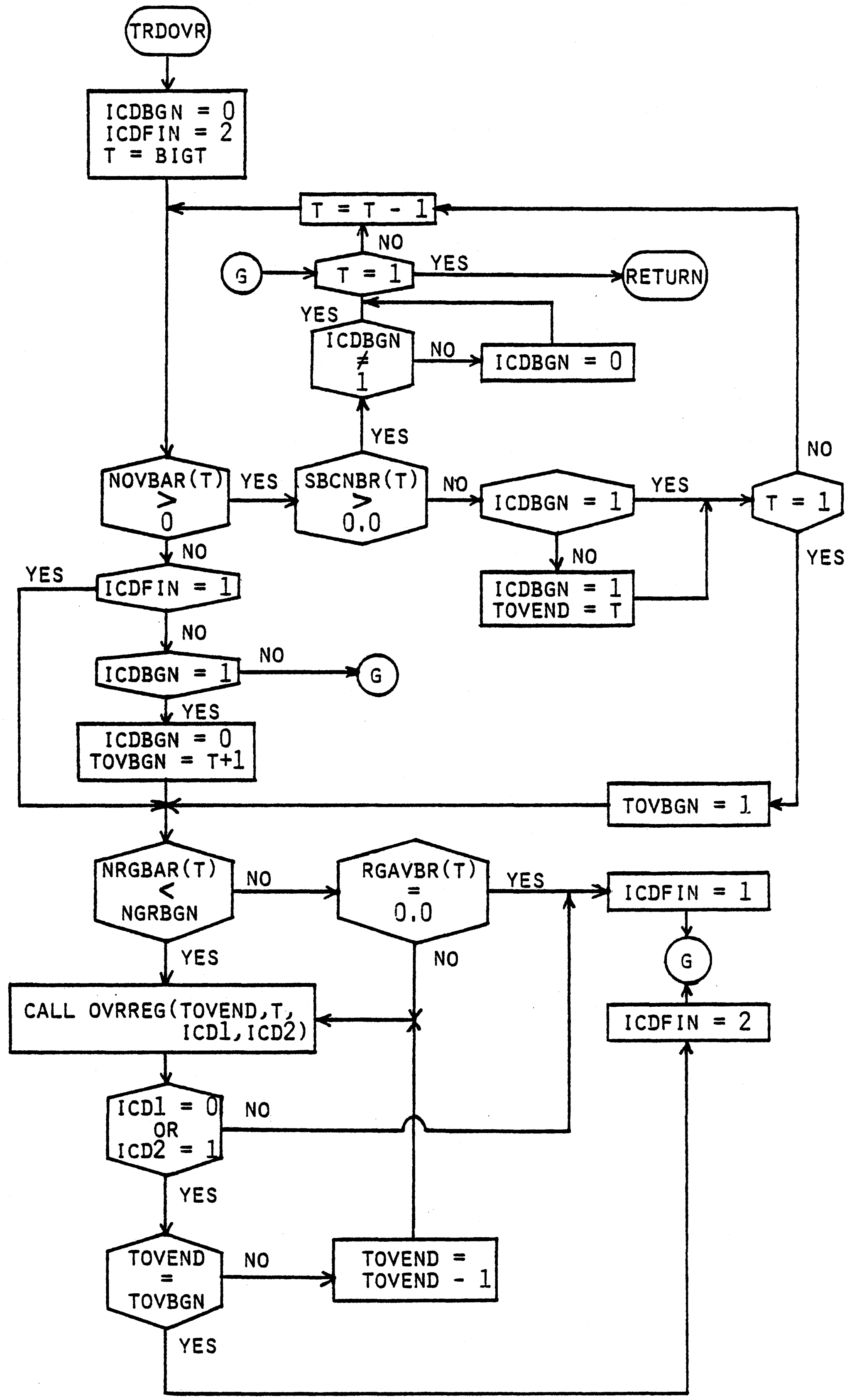
FLOW CHARTS FOR COMPLICATED SUBROUTINES

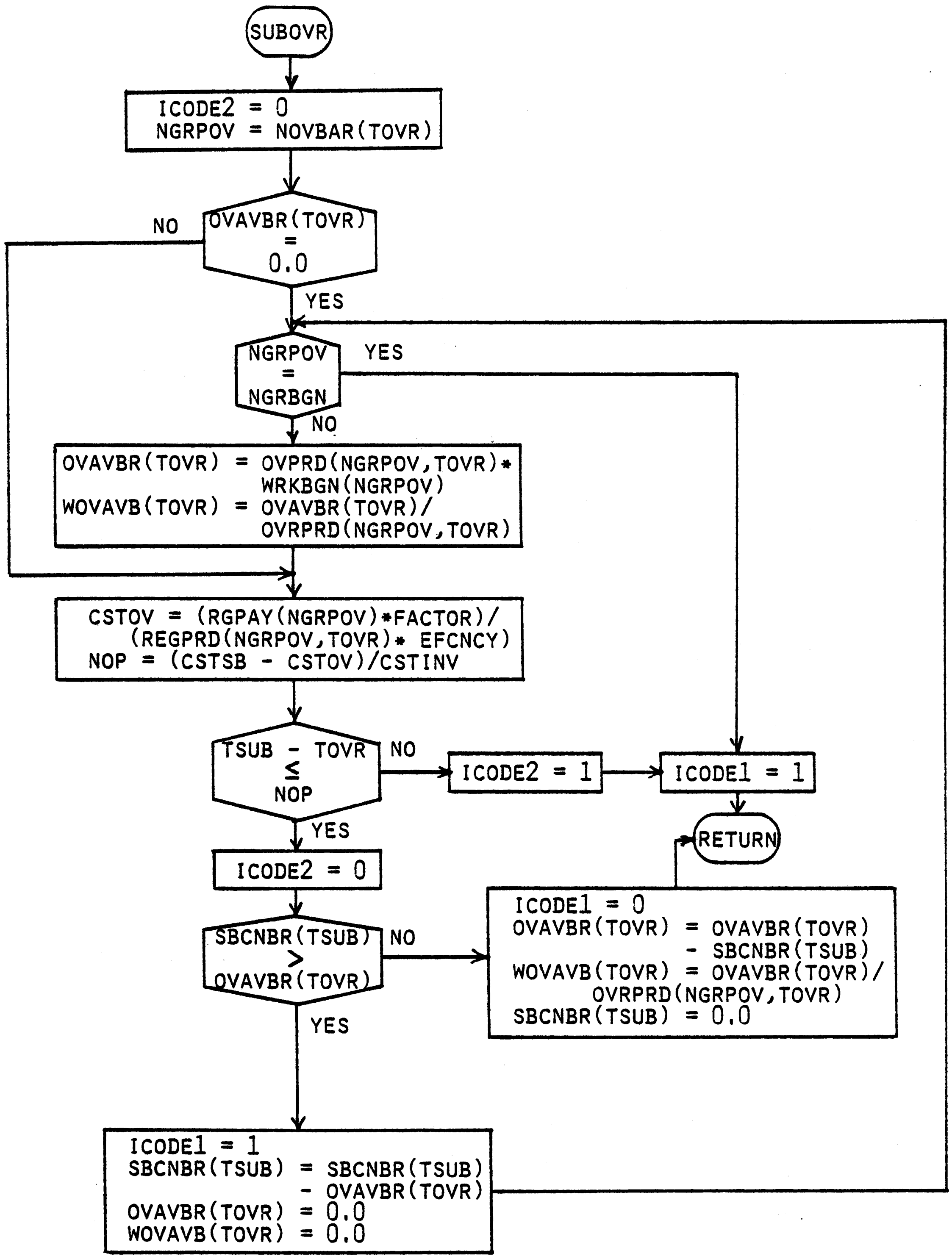


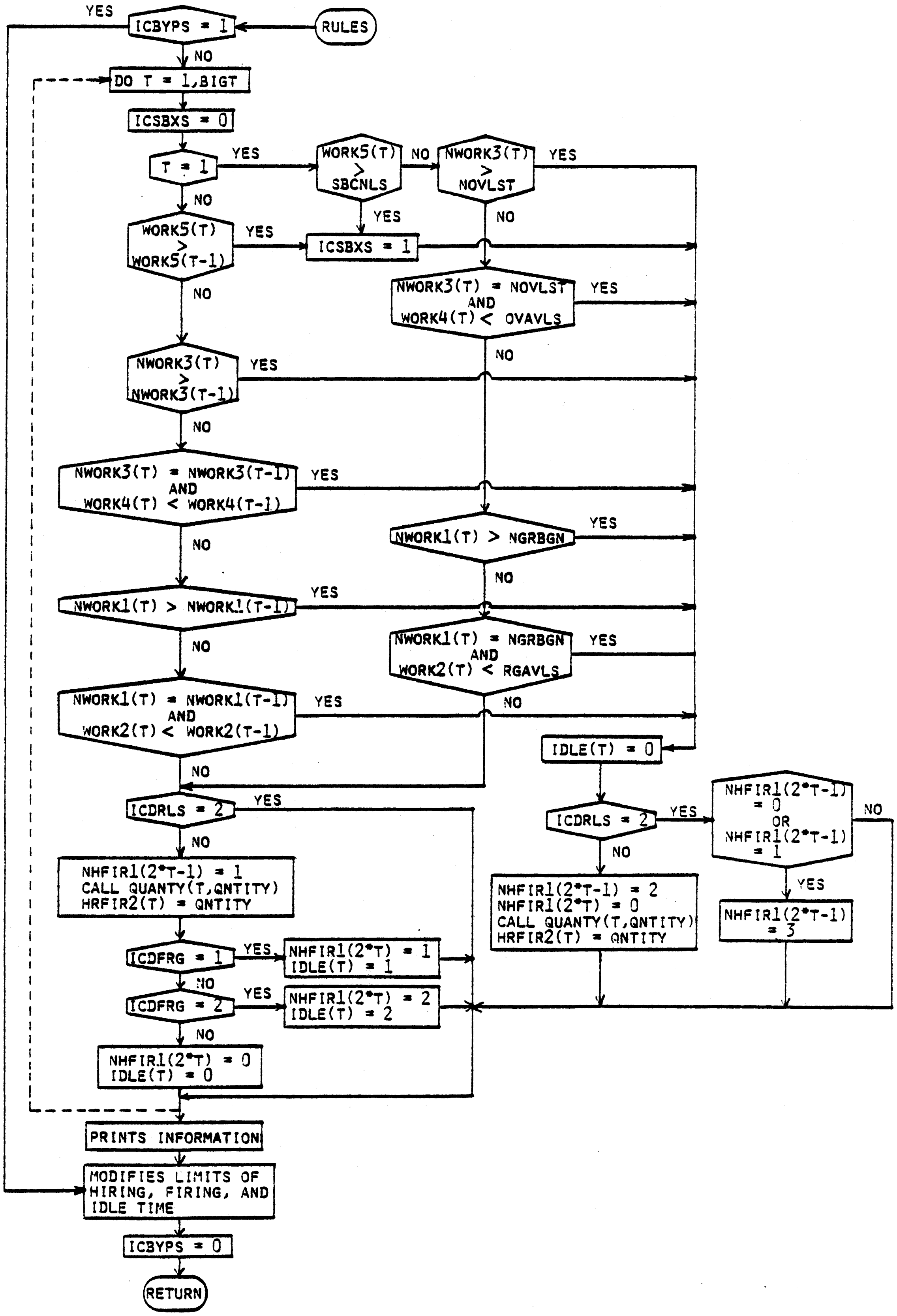
FORMLT
IXPRNO = 2
CALL INFORM
CALL XPRDSN
YES
IXPRNO ≤ 2
NO
CALL LDALCT
ICDRLS = 2
ICBYPS = 0
CALL RULES
DO T = 1,BIGT
NRGBAR(T) = NREG(T)
RGAVBR(T) = REGAVL(T)
WRGAVB(T) = RGAVBR(T)/
REGPRD(NREG(T),T)
NOVBAR(T) = NOVR(T)
OVAVBR(T) = OVRAVL(T)
NOVR(T) = 0
YES
NO
WOVAVB(T) = OVAVBR(T)/
OVRPRD(NOVR(T),T)
WOVAVB(T) = 0.0
SBCNBR(T) = SUBCON(T)
CALL TRDSUB
CALL NATURE(MDLTYP)
MDLTYP = 3
YES
PRINT INFORMATION
NO
YES
MDLTYP = 4
DO T = 1,BIGT
PRINT INFORMATION
NO
I = 2*T - 1
CALL TRDOVR
NHFIR1(I) = 2
YES
NOVBAR(T) = 1
MDLTYP = 2
YES
NO
NO
CALL TRDSUB
PRINT INFORMATION
CALL LDADJS
ICBYPS = 1
ICDRLS = 2
CALL RULES
CALL STRCTR
RETURN

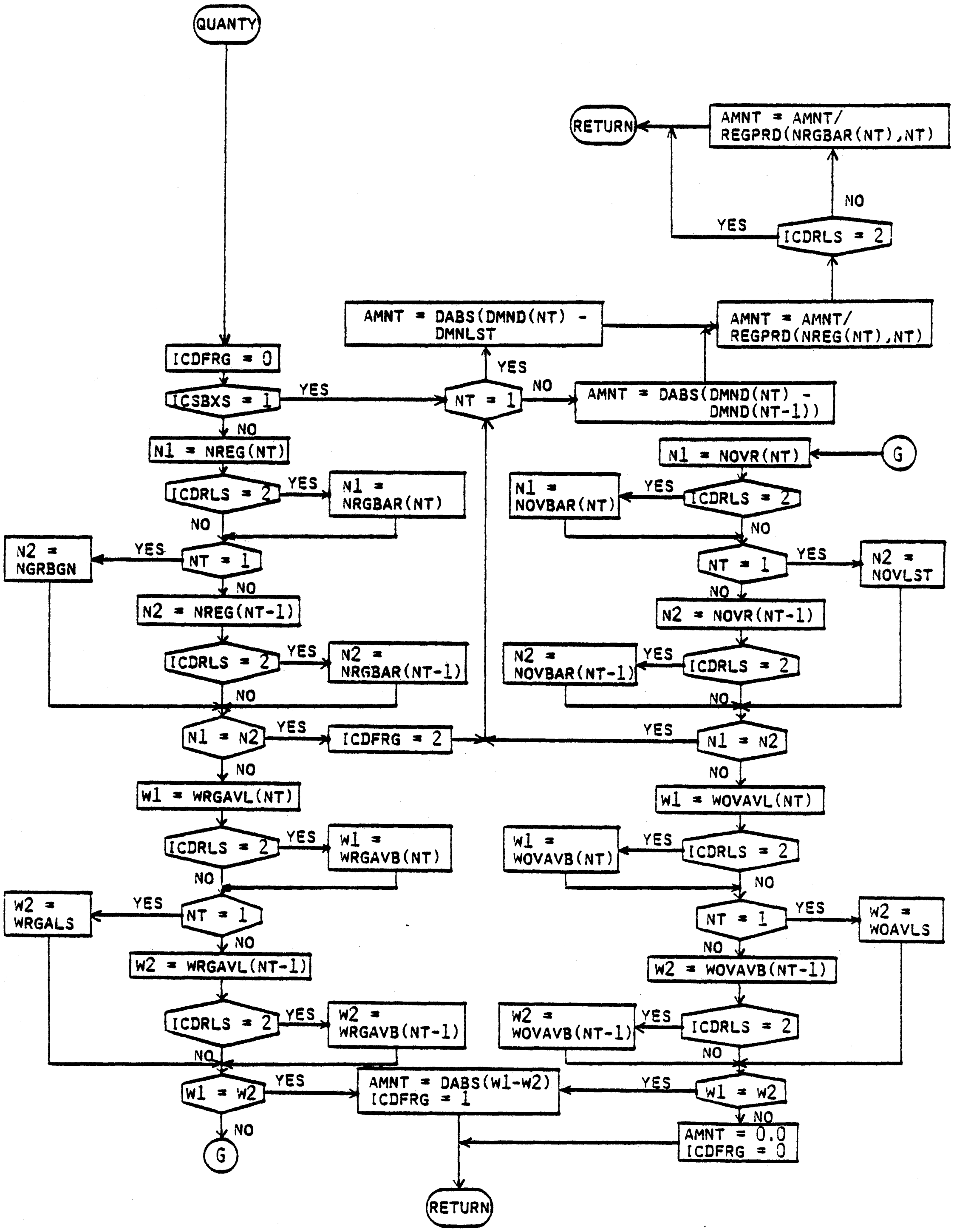

LDALCT
DO T = 1,BIGT
NREG(T) = 0
REGAVL(T) = 0.0
WRGAVL(T) = 0.0
NOVR(T) = 0
OVRAVL(T) = 0.0
WOVAVL(T) = 0.0
SUBCON(T) = 0.0
REMINV = AIZERO
T = 1
AMOUNT = DMND(T) - REMINV
T = T + 1
AMOUNT > 0
NO
REMINV = - AMOUNT
YES
REMINV = 0.0
N = 1
REGCAP = REGPRD(N,T)* WRKBGN(N)
G
NO
AMOUNT > REGCAP
NO
REGAVL(T) = REGCAP - AMOUNT
NREG(T) = N
WRGAVL(T) = REGAVL(T)/ REGPRD(N,T)
T = BIGT
YES
N = N + 1
YES
AMOUNT = AMOUNT - REGCAP
NO
N = NGRBGN
YES
DO T = 1,BIGT
NWORK1(T) = NREG(T)
WORK2(T) = REGAVL(T)
NWORK3(T) = NOVR(T)
WORK4(T) = OVRAVL(T)
WORK5(T) = SUBCON(T)
REGAVL(T) = 0.0
NREG(T) = NGRBGN
N = 1
OVRCAP = OVRPRD(N,T)* WRKBGN(N)
RETURN
AMOUNT > OVRCAP
NO
OVRAVL(T) = OVRCAP - AMOUNT
NOVR(T) = N
OVRTM(T) = AMOUNT
WOVAVL(T) = OVRAVL(T)/ OVRPRD(N,T)
G
N = N + 1
YES
AMOUNT = AMOUNT - OVRCAP
NO
N = NGRBGN
YES
NOVR(T) = NGRBGN
OVRAVL(T) = 0.0
WOVAVL(T) = 0.0
OVRTM(T) = OVRPRD(NGRBGN,T)* WRKBGN(NGRBGN)
SUBCON(T) = AMOUNT
G


TRDSUB
ICD3 = 0; ICD4 = 0
ICD5 = 1; ICD6 = 0
ICD7 = 0
ICDBGN = 0
ICDFIN = 0
ISBRCG = 0
T = BIGT
SUBCON(T) > 0
ICD5 = 0
ICDBGN = 1
T = 1
T = T - 1
ICDBGN = 1
TSBEND = T
ISBRCG = 1
ICDFIN = 1
ICDBGN = 1
TSBBGN = T + 1
TSBBGN = 1
ICDBGN = 0
ICD4 = 0
RETURN
ICDFIN = 1
ICD5 = 0
NOVBAR(T) > 0
ICD4 = 1
ICD4 = 1
TOVRSM = T
NRGBAR(T) = NGRBGN AND RGAVBR(T) = 0.0
CALL SUBREG (TSBEND, T, ICD1,ICD2)
ICD1 = 0
ICD2 = 1
TRGRSM = T
NOVBAR(T) = NGRBGN AND OVAVBR(T) = 0.0
TSBEND = TSBBGN
ICD6 = 0
ICD5 = 1
ICDFIN = 2
TSBEND = TSBEND - 1
ICD6 = 1
ICD8 = 1
ICD8 = 0
T = TRGRSM
T = T-1
SBCNBR(T) > 0.0
ICD6 = 1
TOVSTP = T + 1
NOVBAR(T) > 0
NRGBAR(T) = NGRBGN, & RGAVBR(T) = 0.0
TOVRSM = T;
ICD8 = 1
T = TOVRSM
T = TOVSTP
CALL SUBOVR (TSBEND, T, ICD1, ICD2)
ICD1 = 0 OR ICD2 = 1
ICD6 = 1
YES
NO
A
B


TRDOVR
ICDBGN = 0
ICDFIN = 2
T = BIGT
T = T - 1
NO
G
T = 1
YES
RETURN
YES
ICDBGN ≠ 1
NO
ICDBGN = 0
YES
NO
NOVBAR(T) > 0
YES
SBCNBR(T) > 0.0
NO
ICDBGN = 1
YES
T = 1
NO
YES
ICDFIN = 1
NO
ICDBGN = 1
TOVEND = T
NO
ICDBGN = 1
NO
G
YES
ICDBGN = 0
TOVBGN = T+1
TOVBGN = 1
NRGBAR(T) < NGRBGN
NO
RGAVBR(T) = 0.0
YES
ICDFIN = 1
YES
NO
G
CALL OVRREG(TOVEND,T, ICD1,ICD2)
ICDFIN = 2
ICD1 = 0 OR ICD2 = 1
NO
YES
TOVEND = TOVBGN
NO
TOVEND = TOVEND - 1
YES


SUBOVR
ICODE2 = 0
NGRPOV = NOVBAR(TOVR)
OVAVBR(TOVR) = 0.0
NO
YES
NGRPOV = NGRBGN
YES
NO
OVAVBR(TOVR) = OVPRD(NGRPOV,TOVR)* WRKBGN(NGRPOV)
WOVAVB(TOVR) = OVAVBR(TOVR)/ OVRPRD(NGRPOV,TOVR)
CSTOV = (RGPAY(NGRPOV)*FACTOR)/ (REGPRD(NGRPOV,TOVR)* EFCNCY)
NOP = (CSTSB - CSTOV)/CSTINV
TSUB - TOVR ≤ NOP
NO
ICODE2 = 1
ICODE1 = 1
YES
RETURN
ICODE2 = 0
SBCNBR(TSUB) > OVAVBR(TOVR)
NO
ICODE1 = 0
OVAVBR(TOVR) = OVAVBR(TOVR) - SBCNBR(TSUB)
WOVAVB(TOVR) = OVAVBR(TOVR)/ OVRPRD(NGRPOV,TOVR)
SBCNBR(TSUB) = 0.0
YES
ICODE1 = 1
SBCNBR(TSUB) = SBCNBR(TSUB) - OVAVBR(TOVR)
OVAVBR(TOVR) = 0.0
WOVAVB(TOVR) = 0.0


RULES
ICBYPS = 1
YES
NO
DO T = 1,BIGT
ICSBXS = 0
T = 1
WORK5(T) > SBCNLS
NWORK3(T) > NOVLST
WORK5(T) > WORK5(T-1)
ICSBXS = 1
NWORK3(T) = NOVLST AND WORK4(T) < OVAVLS
NWORK3(T) > NWORK3(T-1)
NWORK3(T) = NWORK3(T-1) AND WORK4(T) < WORK4(T-1)
NWORK1(T) > NGRBGN
NWORK1(T) > NWORK1(T-1)
NWORK1(T) = NGRBGN AND WORK2(T) < RGAVLS
NWORK1(T) = NWORK1(T-1) AND WORK2(T) < WORK2(T-1)
IDLE(T) = 0
ICDRLS = 2
NHFIR1(2*T-1) = 0 OR NHFIR1(2*T-1) = 1
NHFIR1(2*T-1) = 1
CALL QUANTY(T,QNTITY)
HRFIR2(T) = QNTITY
NHFIR1(2*T-1) = 2
NHFIR1(2*T) = 0
NHFIR1(2*T-1) = 3
ICDFRG = 1
NHFIR1(2*T) = 1
IDLE(T) = 1
ICDFRG = 2
NHFIR1(2*T) = 2
IDLE(T) = 2
NHFIR1(2*T) = 0
IDLE(T) = 0
PRINTS INFORMATION
MODIFIES LIMITS OF HIRING, FIRING, AND IDLE TIME
ICBYPS = 0
RETURN


QUANTY
RETURN
AMNT = AMNT/ REGPRD(NRGBAR(NT),NT)
NO
YES
ICDRLS = 2
AMNT = DABS(DMND(NT) - DMNLST
AMNT = AMNT/ REGPRD(NREG(NT),NT)
ICDFRG = 0
YES
ICSBXS = 1
NT = 1
NO
AMNT = DABS(DMND(NT) - DMND(NT-1))
N1 = NREG(NT)
N1 = NOVR(NT)
G
ICDRLS = 2
N1 = NRGBAR(NT)
N1 = NOVBAR(NT)
ICDRLS = 2
N2 = NGRBGN
NT = 1
NT = 1
N2 = NOVLST
N2 = NREG(NT-1)
N2 = NOVR(NT-1)
ICDRLS = 2
N2 = NRGBAR(NT-1)
N2 = NOVBAR(NT-1)
ICDRLS = 2
N1 = N2
ICDFRG = 2
N1 = N2
W1 = WRGAVL(NT)
W1 = WOVAVL(NT)
ICDRLS = 2
W1 = WRGAVB(NT)
W1 = WOVAVB(NT)
ICDRLS = 2
W2 = WRGALS
NT = 1
NT = 1
W2 = WOAVLS
W2 = WRGAVL(NT-1)
W2 = WOVAVB(NT-1)
ICDRLS = 2
W2 = WRGAVB(NT-1)
W2 = WOVAVB(NT-1)
ICDRLS = 2
W1 = W2
AMNT = DABS(W1-W2) ICDFRG = 1
W1 = W2
AMNT = 0.0 ICDFRG = 0
G
RETURN

APPENDIX C

FORTRAN PROGRAM LISTING

```
C     ----------------------------MAIN PROGRAM-------------------------E
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      COMMON/BL380/ICNSFR(12),ICNSHR(12),ICNSID(12),ICNSOB(12),
     1             ICNSOV(24),ICNSWF(12)
      COMMON/BL385/IDOLFR(12),IDOLHR(12),IDOLID(12),IDOOBJ(12),
     1             IDOLOV(24),IDOLWF(12)
      COMMON/BL390/ICNTFR,ICNTHR,ICNTIC,ICNTIV,ICNTOV,ICNTSB
C
C     THIS PROGRAM ASSISTS THE OPERATIONS MANAGER IN DECIDING
C     VARIOUS OPTIONS OF PRODUCTION SMOOTHING, NAMELY, REGULAR
C     WORK FORCE, HIRING AND FIRING AMOUNT, OVERTIME, IDLE TIME,
C     AND SUBCONTRACTING BASED ON MUTIPLE CRITERIA.
C     THE CRITERIA CAN BE ANYTHING PERTINENT TO THE FIRM AND
C     CAN BE CONVENIENTLY INCORPORATED IN THE MODEL THROUGH
C     THE SUBROUTINE STRCTR.  IN THIS CASE THE
C     FOLLOWING CRITERIA HAVE BEEN INCORPORATED:
C
C         1.  MINIMIZE REGULAR PAYROLL, HIRING, AND FIRING COST
C         2.  MINIMIZE INVENTORY COST
C         3.  MINIMIZE NUMBER OF IDLE WORKERS
C         4.  MINIMIZE OVERTIME COST
C
C  FORMULATES THE PROBLEM
      CALL FORMLT
C  SOLVES THE PROBLEM USING REVISED SIMPLEX METHOD
      CALL LNRPRG
C  COMPUTES VARIOUS COST COMPONENTS
      CALL CSTCMP
C  PRINTS RESULTS
      CALL RESULT
C  FINDS THE VALUES OF THE DUAL VARIABLES
      CALL DUALS
      STOP
      END
C
C     *****************************************************************
      SUBROUTINE FORMLT
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
C
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL380/ICNSFR(12),ICNSHR(12),ICNSID(12),ICNSOB(12),
     1             ICNSOV(24),ICNSWF(12)
      COMMON/BL385/IDOLFR(12),IDOLHR(12),IDOLID(12),IDOOBJ(12),
     1             IDOLOV(24),IDOLWF(12)
      COMMON/BL390/ICNTFR,ICNTHR,ICNTIC,ICNTIV,ICNTOV,ICNTSB
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL360/ICBYPS,ICDRLS,ICDFRG,ICSBXS,IXPRNO
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
```

```
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL460/NWORK1(12),NWORK3(12),WORK2(12),WORK4(12),WORK5(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                                WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                                WRGAVL(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
C
C     IXPRNO = MODEL NUMBER
C        1   =  CONSTANT PRODUCTIVITY (FUNDAMENTAL MODEL)
C        2   =  DYNAMIC  PRODUCTIVITY (FUNDAMENTAL MODEL)
C        3   =  CONSTANT PRODUCTIVITY (WORKING MODEL)
C        4   =  DYNAMIC  PRODUCTIVITY (WORKING MODEL)
C     ICDRLS =  CODE; FOR SPECIFYING OPTIONS BY USING THE RULES
C               BEFORE (1) AND AFTER (2) EMPLOYING THE PRINCIPLE OF
C               EFFICIENT UTILIZATION OF RESOURCES.
C     ICBYPS =  CODE USED TO CONTROL THE PROGRAM
C     FOR NREG, REGAVL, NOVR, OVRAVL, AND SUBCON SEE DISSERTATION
C     NRGBAR, RGAVBR,NOVBAR, OVAVBR, SBCNBR ARE RESPECTIVELY SAME
C     AS ABOVE.  DURING EXECUTION PHASE LATER VARIABLES ARE USED
C     KEEPING THE VALUES OF THE FORMER VARIABLES UNCHANGED.
C       BIGT =  PLANNING HORIZON
C     MDLTYP =  INDICATES NATURE OF MODEL STRUCTURE.  NATURE DEPENDS
C               ON WHETHER APPARENT OVERTIME OR SUBCONTRACTING CAN
C               BE TOTALLY ELIMINATED BY FIRST ATTEMPT TO EMPLOY
C               EFFICIENT UTILIZATION OF RESOURCES.  ITS VALUE IS
C               DETERMINED IN THE SUBROUTINE NATURE AND USED TO CONTROL
C               THE PROGRAM FLOW.
C
      IXPRNO = 2
      IF (IXPRNO.EQ.1) WRITE (6,610)
      IF (IXPRNO.EQ.2) WRITE (6,612)
      IF (IXPRNO.EQ.3) WRITE (6,614)
      IF (IXPRNO.EQ.4) WRITE (6,616)
      CALL INFORM
      CALL XPRDSN
      IF (IXPRNO.LE.2) GO TO 80
      CALL LDALCT
      ICDRLS = 1
      ICBYPS = 0
      CALL RULES
C
      DO 20 T = 1,BIGT
         NRGBAR(T) = NREG(T)
         RGAVBR(T) = REGAVL(T)
         WRGAVB(T) = RGAVBR(T)/REGPRD(NREG(T),T)
         NOVBAR(T) = NOVR(T)
         OVAVBR(T) = OVRAVL(T)
         IF (NOVR(T).EQ.0) GO TO 15
         WOVAVB(T) = OVAVBR(T)/OVRPRD(NOVR(T),T)
         GO TO 18
   15    WOVAVB(T) = 0.0
   18    SBCNBR(T) = SUBCON(T)
   20 CONTINUE
      CALL TRDSUB
      CALL NATURE(MDLTYP)
      IF (MDLTYP.EQ.3) THEN
         WRITE (6,630)
```

```
        DO 30 T = 1,BIGT
        I = 2*T-1
        IF (NHFIR1(I).EQ.2) NOVBAR(T) = 1
 30     CONTINUE
        WRITE (6,635)
        WRITE (6,640) (NOVBAR(T),T=1,BIGT)
        ICBYPS = 1
        GO TO 70
        ENDIF
      IF (MDLTYP.EQ.4) WRITE (6,650)
      CALL TRDOVR
      IF (MDLTYP.EQ.2) GO TO 60
      CALL TRDSUB
 60   CALL LDADJS
      ICDRLS = 2
 70   CALL RULES
 80   CALL STRCTR
 610  FORMAT (///,10X,'FUNDAMENTAL MODEL (CONSTANT PRODUCTIVITY) :',//)
 612  FORMAT (///,10X,'FUNDAMENTAL MODEL (DYNAMIC PRODUCTIVITY) :',//)
 614  FORMAT (///,10X,'WORKING MODEL (CONSTANT PRODUCTIVITY) :',//)
 616  FORMAT (///,10X,'WORKING MODEL (DYNAMIC PRODUCTIVITY) :',//)
 630  FORMAT (///,15X,'NO OVERTIME IF ALL WORKERS ARE KEPT.',/,15X,
     1'THIS MIGHT CAUSE IDLE TIME TO EXIST.',//)
 635  FORMAT (/,15X,'SINCE OVERTIME IS A FEASIBLE OPTION',/,
     1          15X,'THIS IS INCLUDED IN THE MODEL WHEN HIRING',/,
     1          15X,'MIGHT TAKE PLACE.  THE NEW OVERTIME SCHE-',/,
     1          15X,'DULE IS AS FOLLOWS.',//)
 640  FORMAT (/,30X,6I4,/)
 650  FORMAT (15X,'SUBCONTRACTING/OVERTIME IN LATER PERIODS',/,
     115X,'CANNOT BE ALLOCATED ECONOMICALLY TO EARLIER PERIODS',/,
     115X,'IF CURRENT WORKFORCE IS MAINTAINED',//)
      RETURN
      END
C
C     ****************************************************************
      SUBROUTINE INFORM
C     THIS SUBROUTINE INPUTS VARIOUS DATA
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL350/DMND(12)
      COMMON/BL355/FRRATE,HRRATE
      COMMON/BL440/OVAVLS,SBCNLS,UPLMWF,WOAVLS,WRGALS,
     1             RGAVLS,DMNLST,NOVLST
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
C
C     NGRBGN = NUMBER OF WORKER CLASSES IN THE BEGINNING OF HORIZON
C  WRKBGN(I) = NUMBER OF WORKER IN THE BEGINNING IN CLASS I
C              LS AT THE END OF A VARIABLE STANDS FOR THEIR  CORRES-
C              PONDING VALUES AT THE END OF THE LAST HORIZON
C      FRCTN = RATIO OF OVERTIME DURATION AND REGULAR TIME
C     EFCNCY = EFFICIENCY OF THE WORKER DURING OVERTIME
C     FACTOR = OVERTIME PAY (TIME)/REGULAR PAY (TIME)
C     AIZERO = INITIAL INVENTORY
C              RGPAY, HRRATE, FRRATE ARE REGULAR PAY, HIRING COST,
C              AND FIRING COST PER WORKER RESPECTIVELY
C    DMND(T) = DEMAND IN THE PERIOD T
```

```
C              CSTINV, CSTSUB ARE UNIT COST OF INVENTORY AND
C              SUBCONTRACTING RESPECTIVELY
C     UPLMWF = UPPER LIMIT OF THE REGULAR WORKFORCE
C   WIDLE(T) = IDLE WORKFORCE DURING THE PERIOD T
C
      BIGT = 10
      WRITE (6,600) BIGT
  600 FORMAT (///,10X,'NUMBER OF PERIODS IN THE HORIZON = ',I3,//)
      NGRBGN = 2
      WRITE (6,603) NGRBGN
  603 FORMAT (10X,'NO OF WORKER CLASSES IN THE BEGINNING = ',I3,//)
      WRKBGN(1) = 10.00
      WRKBGN(2) = 71.00
      DO 10 I = 1,NGRBGN
      WRITE (6,604) I,WRKBGN(I)
  604 FORMAT (10X,'WORKER IN THE CLASS ',I4,' = ',F7.2,/)
   10 CONTINUE
      FRCTN = 0.5
      WRITE (6,608) FRCTN
  608 FORMAT (//,10X,'OVERTIME DUR/RGLR. TIME DUR = ',F5.2,//)
      EFCNCY = 1.0
      WRITE (6,612) EFCNCY
  612 FORMAT (10X,'THE EFFICIENCY DURING OVERTIME = ',F5.2,//)
      FACTOR = 1.5
      WRITE (6,616) FACTOR
  616 FORMAT (10X,'OVERTIME PAY/REGULAR PAY = ',F7.2,//)
      SBCNLS = 0.0
      WRITE (6,620) SBCNLS
  620 FORMAT (10X,'SUBCONTRACTING IN THE LAST PERIOD = ',F7.2,//)
      NOVLST = 0
      WRITE (6,624) NOVLST
  624 FORMAT (10X,'NO. OF CLASSES WORKING OVERTIME LAST PERIOD = ',I4,
     1//)
      OVAVLS = 0.0
      WRITE (6,628) OVAVLS
  628 FORMAT (10X,'NO. OF WORKERS (IN LOWEST CLASS THROUGH WHICH',/,
     110X,'OVERTIME EXTENDED IN THE LAST PERIOD) NOT UTILIZED =',F7.2//)
      WOAVLS = 0.0
      WRGALS = 0.0
      RGAVLS = 0.0
      DMNLST = 400.00
      WRITE (6,642) DMNLST
  642 FORMAT (10X,'DEMAND IN THE LAST PERIOD = ',F8.2,/)
      AIZERO = 263.00
      WRITE (6,644) AIZERO
  644 FORMAT (10X,'INITIAL INVENTORY = ',F7.2,//)
      HRRATE = 180.00
      WRITE (6,648) HRRATE
  648 FORMAT (10X,'HIRING COST PER WORKER = ',F7.2,/)
      FRRATE = 360.00
      WRITE (6,652) FRRATE
  652 FORMAT (10X,'FIRING COST PER WORKER = ',F7.2,//)
      RGPAY(1) = 340.00
      RGPAY(2) = 340.00
      DO 20 I = 1,NGRBGN
      WRITE (6,656) I,RGPAY(I)
  656 FORMAT (10X,'REGULAR PAYROLL PER WORKER OF CLASS ',I3,' = ',
     1F8.2,/)
```

```
   20 CONTINUE
      UPLMWF = 600.00
      WRITE (6,660) UPLMWF
  660 FORMAT (/,10X,'UPPER LIMIT OF REGULAR WORKFORCE = ',F7.2,//)
      CSTINV = 20.00
      WRITE (6,664) CSTINV
  664 FORMAT (10X,'UNIT INVENTORY CARRYING COST = ',F7.2,/)
      CSTSB = 200.00
      WRITE (6,668) CSTSB
  668 FORMAT (10X,'UNIT SUBCONTRACTING COST = ',F7.2,///)
C
      DMND(1) = 430.00
      DMND(2) = 447.00
      DMND(3) = 440.00
      DMND(4) = 316.00
      DMND(5) = 397.00
      DMND(6) = 375.00
      DMND(7) = 292.00
      DMND(8) = 458.00
      DMND(9) = 400.00
      DMND(10)= 350.00
C
      DO 30 I = 1,BIGT
      WRITE (6,672) I,DMND(I)
  672 FORMAT (10X,'DEMAND IN PERIOD',I4,' = ',F10.2,/)
   30 CONTINUE
      ALBIDL = 20.00
      WRITE (6,676) ALBIDL
  676 FORMAT (10X,'NO OF ALLOWABLE IDLE WORKER AT ANY PERIOD = ',F7.2,
     1//)
      DO 50 T = 1,BIGT
   50 WIDLE(T) = ALBIDL
      RETURN
      END
C     ***************************************************************
      SUBROUTINE XPRDSN
C     THIS SUBROUTINE INPUTS PRODUCTIVITY OF WORKERS DEPENDING ON
C     THE EXPERIMENT NUMBER.
C     REGPRD(J,T) = REGULAR PRODUCTIVITY OF THE J-TH CLASS
C                   IN PERIOD T
C     OVRPRD(J,T) = OVERTIME PRODUCTIVITY OF THE J-TH CLASS
C                   IN PERIOD T
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL350/DMND(12)
      COMMON/BL360/ICBYPS,ICDRLS,ICDFRG,ICSBXS,IXPRNO
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                           WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                           WRGAVL(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
      COMMON/BL580/OVPRD(2),RGPRD(2)
```

```
      IF (IXPRNO.GT.2) GO TO 100
      WFCCNG = 500.00
      WRITE (6,600) WFCCNG
  600 FORMAT (10X,'MAXM. CHANGE IN WORKFORCE AT ANY PERIOD = ',F7.2,//)
      DO 10 T = 1,BIGT
         NREG(T)   = 2
         NRGBAR(T) = NREG(T)
         WRGAVL(T) = 0.0
         WRGAVB(T) = WRGAVL(T)
         NOVR(T)   = 2
         NOVBAR(T) = NOVR(T)
         WOVAVL(T) = 0.0
         WOVAVB(T) = WOVAVL(T)
         NHFIR1(2*T-1) = 3
         NHFIR1(2*T) = 2
         HRFIR2(T) = WFCCNG
         IDLE(T)   = 1
         SUBCON(T) = 0.0
         SBCNBR(T) = SUBCON(T)
   10 CONTINUE
C
  100 IF (IXPRNO.EQ.2.OR.IXPRNO.EQ.4) GO TO 200
      RGPRD(1) = 30.00
      RGPRD(2) = 30.00
      DO 15 I = 1,NGRBGN
      WRITE (6,604) I,RGPRD(I)
      OVPRD(I) = EFCNCY*RGPRD(I)
      WRITE (6,608) OVPRD(I)
  604 FORMAT (10X,'REGULAR TIME PRODUCTION RATE OF CLASS ',I3,' = ',
     1F8.2,/)
  608 FORMAT (10X,'ESTIMATED OVERTIME PRODUCTIVITY OF THIS CLASS = ',
     1F7.2,/)
   15 CONTINUE
      DO 20 T = 1,BIGT
      DO 20 I = 1,NGRBGN
      REGPRD(I,T) = RGPRD(I)
      OVRPRD(I,T) = OVPRD(I)
   20 CONTINUE
      RETURN
C THE FOLLOWING VALUES WERE USED FOR DIFFERENT PURPOSE
C
  200 REGPRD(1,1) = 9.490
      REGPRD(1,2) = 9.671
      REGPRD(1,3) = 8.986
      REGPRD(1,4) = 8.428
      REGPRD(1,5) = 8.103
      REGPRD(1,6) =10.114
      REGPRD(1,7) =10.914
      REGPRD(1,8) =11.475
      REGPRD(1,9) =11.907
      REGPRD(1,10)= 8.179
C
      DO 30 T = 1,BIGT
         OVRPRD(1,T) = EFCNCY*REGPRD(1,T)
   30 CONTINUE
      DO 40 T = 1,BIGT
         REGPRD(2,T) = REGPRD(1,T)
         OVRPRD(2,T) = OVRPRD(1,T)
```

```
   40 CONTINUE
      RETURN
      END
C     *****************************************************************
      SUBROUTINE LDALCT
C     THIS SUBROUTINE IDENTIFIES WORK LOAD PATTERN.
C     REMINV = REMAINING INITIAL INVENTORY IF TOTAL DEMAND IN FIRST
C              FEW PERIODS IS LESS THAN INITIAL INVENTORY.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL350/DMND(12)
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL460/NWORK1(12),NWORK3(12),WORK2(12),WORK4(12),WORK5(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                              WRGAVL(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
      DO 5 T = 1,BIGT
         NREG(T)   = 0
         REGAVL(T) = 0.0
         WRGAVL(T) = 0.0
         NOVR(T)   = 0
         OVRAVL(T) = 0.0
         WOVAVL(T) = 0.0
         SUBCON(T) = 0.0
    5 CONTINUE
      REMINV = AIZERO
      T = 1
   10 AMOUNT = DMND(T) - REMINV
      IF (AMOUNT.GT.0) GO TO 15
      REMINV = - AMOUNT
      GO TO 115
   15 REMINV = 0.0
C     ALLOCATES WORK TO REGULAR WORKFORCE
      N = 1
   20 REGCAP = REGPRD(N,T)*WRKBGN(N)
      IF (AMOUNT.GT.REGCAP) GO TO 70
      REGAVL(T) = REGCAP - AMOUNT
      NREG(T) = N
      WRGAVL(T) = REGAVL(T)/REGPRD(N,T)
      GO TO 115
   70 AMOUNT = AMOUNT-REGCAP
      IF (N.EQ.NGRBGN) GO TO 80
      N = N+1
      GO TO 20
   80 REGAVL(T) = 0.0
      NREG(T) = NGRBGN
C     ALLOCATES WORK TO OVERTIME
      N = 1
   90 OVRCAP = OVRPRD(N,T)*WRKBGN(N)
      IF (AMOUNT.GT.OVRCAP) GO TO 100
      OVRAVL(T) = OVRCAP-AMOUNT
      NOVR(T) = N
      OVRTM(T) = AMOUNT
      WOVAVL(T) = OVRAVL(T)/OVRPRD(N,T)
      GO TO 115
```

```
  100 AMOUNT = AMOUNT-OVRCAP
      IF (N.EQ.NGRBGN) GO TO 110
      N = N+1
      GO TO 90
C     IF DEMAND CANNOT BE FULFILLED BY REGULAR TIME
C     AND OVERTIME SUBCONTRACTING MIGHT EXIST
  110 NOVR(T) = NGRBGN
      OVRAVL(T) = 0.0
      WOVAVL(T) = 0.0
      OVRTM(T) = OVRPRD(NGRBGN,T) * WRKBGN(NGRBGN)
      SUBCON(T) = AMOUNT
  115 IF (T.EQ.BIGT) GO TO 120
      T = T+1
      GO TO 10
C
  120 DO 130 T = 1,BIGT
         NWORK1(T) = NREG(T)
         WORK2(T)  = REGAVL(T)
         NWORK3(T) = NOVR(T)
         WORK4(T)  = OVRAVL(T)
         WORK5(T)  = SUBCON(T)
  130 CONTINUE
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE TRDSUB
C     THIS SUBROUTINE PERFORMS OPERATION (1) OF STEP-3 IN
C     STAGE 1 OF CONSTRUCTION PROCESS OF THE MATRIX A
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T,TSBBGN,TSBEND,TOVRSM,TOVSTP,TRGRSM
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                             WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                             WRGAVL(12)
C
      ICD3 = 0
      ICD4 = 0
      ICD5 = 1
      ICD6 = 0
      ICD7 = 0
      ICDBGN = 0
      ICDFIN = 2
      ISBRCG = 0
      T = BIGT
C
   10 IF (SUBCON(T).GT.0.0) GO TO 40
      IF (ICDFIN.EQ.1) GO TO 100
      IF ( ICDBGN.EQ.1) GO TO 70
      IF (T.NE.1) GO TO 60
      RETURN
   40 IF (ICD5.EQ.0) GO TO 210
      IF (ICDBGN.EQ.1) GO TO 50
      ICDBGN = 1
      TSBEND = T
```

```
      ISBRCG = 1
   50 IF (T.EQ.1) GO TO 80
   60 T = T-1
      GO TO 10
   70 TSBBGN = T+1
      GO TO 90
   80 TSBBGN = 1
   90 ICDBGN = 0
      ICD4 = 0
C
  100 IF (NOVBAR(T).GT.0) GO TO 130
      IF (ICD4.EQ.1) GO TO 110
      ICD4 = 1
      TOVRSM = T
  110 IF (NRGBAR(T).EQ.NGRBGN.AND.RGAVBR(T).EQ.0.0) GO TO 120
      GO TO 1000
  120 IF (T.EQ.1) GO TO 250
      T = T-1
      GO TO 100
  130 IF (NOVBAR(T).EQ.NGRBGN.AND.OVAVBR(T).EQ.0.0) GO TO 140
      GO TO 150
  140 IF (T.EQ.1) GO TO 180
      T = T-1
      GO TO 100
  150 IF (ICD4.EQ.1) GO TO 170
      ICD4 = 1
      TOVRSM = T
      GO TO 170
  160 IF (T.EQ.1) GO TO 250
  170 T = T-1
      IF (SBCNBR(T).GT.0.0) GO TO 210
      IF (NOVBAR(T).GT.0) GO TO 160
      IF (NRGBAR(T).EQ.NGRBGN.AND.RGAVBR(T).EQ.0.0) GO TO 160
      GO TO 1000
  180 IF (ICD4.EQ.1) GO TO 250
      GO TO 3000
  210 ICD6 = 1
      TOVSTP = T+1
      GO TO 250
C
 1000 CALL SUBREG(TSBEND,T,ICD1,ICD2)
      IF (ICD1.EQ.0) GO TO 220
      IF (ICD2.EQ.1) GO TO 240
      GO TO 400
  220 IF (TSBEND.EQ.TSBBGN) GO TO 300
      TSBEND = TSBEND-1
      IF (ICD6.EQ.1) GO TO 250
      IF (ICD8.EQ.1) GO TO 230
      GO TO 1000
  230 ICD8 = 0
      T = TRGRSM
      GO TO 1000
  240 TRGRSM = T
  250 T = TOVRSM
 2000 CALL SUBOVR(TSBEND,T,ICD1,ICD2)
      IF (ICD1.EQ.0.OR.ICD2.EQ.1) GO TO 260
      GO TO 270
  260 TOVRSM = T
```

```
      ICD8 = 1
      GO TO 220
  270 IF (ICD6.EQ.1) GO TO 290
      IF (T.EQ.1) GO TO 3000
  280 T = T-1
      GO TO 2000
  290 IF (T.EQ.TOVSTP) GO TO 300
      GO TO 280
C
  300 ICD6 = 0
      ICD5 = 1
      ICDFIN = 2
      GO TO 410
  400 ICD5 = 0
      ICDFIN = 1
  410 IF (T.EQ.1) GO TO 3000
      GO TO 60
 3000 RETURN
      END
C
C     ******************************************************************
      SUBROUTINE NATURE(ICODE)
C     THIS SUBROUTINE IS A PART OF EFFICIENT UTILIZATION OF RESOURCES.
C     AFTER OPERATION (1) IT IS CALLED AND ICODE IS ASSIGNED A VALUE.
C     THIS VALUE DETERMINES IF OPERATIONS (2) AND (3) ARE TO BE
C     PERFORMED OR NOT.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
      IF (ISBRCG.NE.0) GO TO 20
      T = 1
   10 IF (NOVBAR(T).GT.0) GO TO 25
      IF (T.EQ.BIGT) GO TO 60
      T = T+1
      GO TO 10
   20 ICODE = 1
      GO TO 30
   25 ICODE = 2
C
   30 T = 1
   40 IF (NRGBAR(T).LT.NGRBGN) RETURN
      IF (RGAVBR(T).EQ.0.0) GO TO 50
      RETURN
   50 IF (T.EQ.BIGT) GO TO 70
      T = T+1
      GO TO 40
   60 ICODE = 3
      RETURN
   70 ICODE = 4
      RETURN
      END
C
C     ******************************************************************
      SUBROUTINE TRDOVR
```

```
C     THIS SUBROUTINE IS PART OF OPERATION (2) OF STEP-2 OF STAGE 1
C     OF CONSTRUCTING THE WORKING MODEL.  IT FINDS TOVR AND TREG.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T,TOVBGN,TOVEND
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
C
      ICDBGN = 0
      ICDFIN = 2
      T = BIGT
   10 IF (NOVBAR(T).GT.0) GO TO 40
      IF (ICDFIN.EQ.1) GO TO 110
      IF (ICDBGN.EQ.1) GO TO 90
      GO TO 70
   40 IF (SBCNBR(T).GT.0.0) GO TO 60
      IF (ICDBGN.EQ.1) GO TO 50
      ICDBGN = 1
      TOVEND = T
   50 IF (T.EQ.1) GO TO 100
      GO TO 80
   60 IF (ICDBGN.NE.1) GO TO 70
      ICDBGN = 0
   70 IF (T.EQ.1) GO TO 160
   80 T = T-1
      GO TO 10
   90 ICDBGN = 0
      TOVBGN = T +1
      GO TO 110
  100 TOVBGN = 1
C
  110 IF (NRGBAR(T).LT.NGRBGN) GO TO 120
      IF (RGAVBR(T).EQ.0.0) GO TO 140
  120 CALL OVRREG(TOVEND,T,ICD1,ICD2)
      IF (ICD1.EQ.0.OR.ICD2.EQ.1) GO TO 130
      GO TO 140
  130 IF (TOVEND.EQ.TOVBGN) GO TO 150
      TOVEND = TOVEND-1
      GO TO 120
  140 ICDFIN = 1
      GO TO 70
  150 ICDFIN = 2
      GO TO 70
  160 RETURN
      END
C
C     ******************************************************************
      SUBROUTINE SUBOVR(TSUB,TOVR,ICODE1,ICODE2)
C     THIS SUBROUTINE ATTEMPTS TO ALLOCATE SUBCONTRACTING AMOUNT IN
C     TSUB TO UNUTILIZED OVERTIME IN TOVR.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,TOVR,TSUB
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
```

```
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
      ICODE2 = 0
      NGRPOV = NOVBAR(TOVR)
      IF (OVAVBR(TOVR).EQ.0.0) GO TO 10
      GO TO 20
   10 IF (NGRPOV.EQ.NGRBGN) GO TO 30
      NGRPOV = NGRPOV+1
      OVAVBR(TOVR) = OVRPRD(NGRPOV,TOVR)*WRKBGN(NGRPOV)
      WOVAVB(TOVR) = OVAVBR(TOVR)/OVRPRD(NGRPOV,TOVR)
   20 CSTOV = (RGPAY(NGRPOV)*FACTOR)/(REGPRD(NGRPOV,TOVR)*EFCNCY)
      NOP = (CSTSB-CSTOV)/CSTINV
      IF ((TSUB-TOVR).LE.NOP) GO TO 40
      ICODE2 = 1
   30 ICODE1 = 1
      GO TO 60
   40 ICODE2 = 0
      IF (SBCNBR(TSUB).GT.OVAVBR(TOVR)) GO TO 50
      ICODE1 = 0
      OVAVBR(TOVR) = OVAVBR(TOVR)-SBCNBR(TSUB)
      WOVAVB(TOVR) = OVAVBR(TOVR)/OVRPRD(NGRPOV,TOVR)
      SBCNBR(TSUB) = 0.0
      GO TO 60
   50 ICODE1 = 1
      SBCNBR(TSUB) = SBCNBR(TSUB)-OVAVBR(TOVR)
      OVAVBR(TOVR) = 0.0
      WOVAVB(TOVR) = 0.0
      GO TO 10
   60 RETURN
      END
C
C     ****************************************************************8
      SUBROUTINE SUBREG(TSUB,TREG,ICODE1,ICODE2)
C     THIS SUBROUTINE ATTEMPTS TO ALLOCATE SUBCONTRACTING AMOUNT IN
C     TSUB TO REGULAR TIME IN TREG.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,TREG,TSUB
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
C
      ICODE2 = 0
      NGRPRG = NRGBAR(TREG)
      IF (RGAVBR(TREG).EQ.0.0) GO TO 10
      GO TO 20
   10 IF (NGRPRG.EQ.NGRBGN) GO TO 30
      NGRPRG = NGRPRG+1
      RGAVBR(TREG) = REGPRD(NGRPRG,TREG)*WRKBGN(NGRPRG)
      WRGAVB(TREG) = RGAVBR(TREG)/REGPRD(NGRPRG,TREG)
   20 CSTRG = RGPAY(NGRPRG)/REGPRD(NGRPRG,TREG)
      NOP = (CSTSB-CSTRG)/CSTINV
```

```
      IF ((TSUB-TREG).LE.NOP) GO TO 40
      ICODE2 = 1
   30 ICODE1 = 1
      GO TO 60
   40 ICODE2 = 0
      IF (SBCNBR(TSUB).GT.RGAVBR(TREG)) GO TO 50
      ICODE1 = 0
      RGAVBR(TREG) = RGAVBR(TREG)-SBCNBR(TSUB)
      WRGAVB(TREG) = RGAVBR(TREG)/REGPRD(NGRPRG,TREG)
      GO TO 60
   50 ICODE1 = 1
      SBCNBR(TSUB) = SBCNBR(TSUB)-RGAVBR(TREG)
      RGAVBR(TREG) = 0.0
      WRGAVB(TREG) = 0.0
      GO TO 10
   60 RETURN
      END
C
C     ****************************************************************
      SUBROUTINE OVRREG(TOVR,TREG,ICODE1,ICODE2)
C     THIS SUBROUTINE ATTEMPTS TO ALLOCATE OVERTIME IN TOVR TO
C     REGULAR TIME IN TREG.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,TOVR,TREG
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
C
      ICODE2 = 0
      NGRPRG = NRGBAR(TREG)
      NGRPOV = NOVBAR(TOVR)
      IF (OVAVBR(TOVR).EQ.0.0) GO TO 10
      GO TO 40
   10 IF (NGRPOV.EQ.NGRBGN) GO TO 20
      GO TO 30
   20 ICODE1 = 0
      GO TO 100
   30 NGRPOV = NGRPOV-1
      OVAVBR(TOVR) = OVRPRD(NGRPOV,TOVR)*WRKBGN(NGRPOV)
      WOVAVB(TOVR) = OVAVBR(TOVR)/OVRPRD(NGRPOV,TOVR)
   40 CSTOV = (RGPAY(NGRPOV)*FACTOR)/(REGPRD(NGRPOV,TOVR)*EFCNCY)
      IF (RGAVBR(TREG).EQ.0.0) GO TO 50
      GO TO 60
   50 IF (NGRPRG.EQ.NGRBGN) GO TO 70
      NGRPRG = NGRPRG+1
      RGAVBR(TREG) = REGPRD(NGRPRG,TREG)*WRKBGN(NGRPRG)
      WRGAVB(TREG) =RGAVBR(TREG)/REGPRD(NGRPRG,TREG)
   60 CSTRG = RGPAY(NGRPRG)/REGPRD(NGRPRG,TREG)
      NOP = (CSTOV-CSTRG)/CSTINV
      IF ((TOVR-TREG).LE.NOP) GO TO 80
      ICODE2 = 1
   70 ICODE1 = 1
      GO TO 100
```

```
   80 ICODE2 = 0
      IF (OVAVBR(TOVR).GT.RGAVBR(TREG)) GO TO 90
      RGAVBR(TREG) = RGAVBR(TREG)-OVAVBR(TOVR)
      WRGAVB(TREG) = RGAVBR(TREG)/REGPRD(NGRPRG,TREG)
      OVAVBR(TOVR) = 0.0
      WOVAVB(TOVR) = 0.0
      GO TO 10
   90 OVAVBR(TOVR) = OVAVBR(TOVR)-RGAVBR(TREG)
      WOVAVB(TOVR) = OVAVBR(TOVR)/OVRPRD(NGRPOV,TOVR)
      RGAVBR(TREG) = 0.0
      WRGAVB(TREG) = 0.0
      GO TO 50
  100 RETURN
      END
C
C     *******************************************************************
      SUBROUTINE LDADJS
C     THIS SUBROUTINE PERFORMS THE OPERATION DESCRIBED IN PAGE 84
C     OF DISSERTATION.
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T,TSBBGN,TSBEND
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL460/NWORK1(12),NWORK3(12),WORK2(12),WORK4(12),WORK5(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                               WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                               WRGAVL(12)
      DO 20 T = 1,BIGT
      IF (NRGBAR(T).LT.NGRBGN) GO TO 10
      IF (NRGBAR(T).EQ.NGRBGN.AND.RGAVBR(T).EQ.0.0) GO TO 15
      IDLE(T) = 1
      GO TO 20
   10 IDLE(T) = 2
      GO TO 20
   15 IDLE(T) = 0
   20 CONTINUE
      IF (ISBRCG.EQ.0) GO TO 180
      T = BIGT
   30 ICDBGN = 0
      IF (SUBCON(T).GT.0.0) GO TO 40
      GO TO 70
   40 IF (ICDBGN.EQ.1) GO TO 50
      ICDBGN = 1
      TSBEND = T
   50 IF (T.EQ.1) GO TO 100
   60 T = T-1
      GO TO 30
   70 IF (ICDBGN.EQ.1) GO TO 90
   80 IF (T.EQ.1) GO TO 180
      GO TO 60
   90 TSBBGN = T+1
      GO TO 110
  100 TSBBGN = 1
  110 ICDBGN = 0
  120 IF (SBCNBR(TSBEND).LT.SUBCON(TSBEND)) GO TO 150
  130 IF (TSBEND.EQ.TSBBGN) GO TO 80
```

```
  140 TSBEND = TSBEND-1
      GO TO 120
  150 SHIFT = SUBCON(TSBEND)-SBCNBR(TSBEND)
      SBCNBR(TSBEND) = SUBCON(TSBEND)
      IF (SBCNBR(TSBBGN).GT.SHIFT) GO TO 160
      GO TO 170
  160 SBCNBR(TSBBGN) = SBCNBR(TSBBGN)-SHIFT
      SHIFT = 0.0
      GO TO 130
  170 SHIFT = SHIFT-SBCNBR(TSBBGN)
      SBCNBR(TSBBGN) = 0.0
      TSBBGN = TSBBGN+1
      IF (SBCNBR(TSBBGN).LT.SHIFT) GO TO 170
      SBCNBR(TSBBGN) = SBCNBR(TSBBGN)-SHIFT
      SHIFT = 0.0
      GO TO 140
C
  180 DO 190 T = 1,BIGT
         NWORK1(T) = NRGBAR(T)
         WORK2(T)  = RGAVBR(T)
         NWORK3(T) = NOVBAR(T)
         WORK4(T)  = OVAVBR(T)
         WORK5(T)  = SBCNBR(T)
  190 CONTINUE
      RETURN
      END
C
C     ****************************************************************
      SUBROUTINE RULES
C     THIS SUBROUTINE PERFORMS STEP-2 AND STEP-4 OF STAGE 1 OF
C     CONSTRUCTION OF THE WORKING MODEL.
C     IT ALSO SETS INITIAL VALUES FOR RIGHT-HAND SIDE VALUES
C     FOR HIRING, FIRING, AND IDLE TIME CONSTRAINTS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL360/ICBYPS,ICDRLS,ICDFRG,ICSBXS,IXPRNO
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL440/OVAVLS,SBCNLS,UPLMWF,WOAVLS,WRGALS,
     1             RGAVLS,DMNLST,NOVLST
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL460/NWORK1(12),NWORK3(12),WORK2(12),WORK4(12),WORK5(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                             WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                             WRGAVL(12)
C
      IF (ICBYPS.EQ.1) GO TO 260
      DO 200 T = 1,BIGT
      ICSBXS = 0
      IF (T.EQ.1) GO TO 20
      GO TO 60
   20 IF (WORK5(T).GT.SBCNLS) GO TO 155
      IF (NWORK3(T).GT.NOVLST) GO TO 170
      IF (NWORK3(T).EQ.NOVLST.AND.WORK4(T).LT.OVAVLS) GO TO 170
      IF (NWORK1(T).GT.NGRBGN) GO TO 170
```

```
      IF (NWORK1(T).EQ.NGRBGN.AND.WORK2(T).LT.RGAVLS) GO TO 170
      GO TO 130
   60 IF (WORK5(T).GT.WORK5(T-1)) GO TO 155
      IF (NWORK3(T).GT.NWORK3(T-1)) GO TO 170
      IF (NWORK3(T).EQ.NWORK3(T-1).AND.WORK4(T).LT.WORK4(T-1)) GO TO 170
      IF (NWORK1(T).GT.NWORK1(T-1)) GO TO 170
      IF (NWORK1(T).EQ.NWORK1(T-1).AND.WORK2(T).LT.WORK2(T-1)) GO TO 170
  130 IF (ICDRLS.EQ.2) GO TO 200
      NHFIR1(2*T-1) = 1
      CALL QUANTY(T,QNTITY)
      HRFIR2(T) = QNTITY
      IF (ICDFRG.EQ.1) GO TO 140
      IF (ICDFRG.EQ.2) GO TO 150
      NHFIR1(2*T) = 0
      IDLE(T) = 0
      GO TO 200
  140 NHFIR1(2*T) = 1
      IDLE(T) = 1
      GO TO 200
  150 NHFIR1(2*T) = 2
      IDLE(T) = 2
      GO TO 200
C
  155 ICSBXS = 1
  170 IDLE(T) = 0
      IF (ICDRLS.EQ.2) GO TO 180
      NHFIR1(2*T-1) = 2
      NHFIR1(2*T) = 0
      CALL QUANTY(T,QNTITY)
      HRFIR2(T) = QNTITY
      GO TO 200
  180 IF (NHFIR1(2*T-1).EQ.0.OR.NHFIR1(2*T-1).EQ.1) GO TO 190
      GO TO 200
  190 NHFIR1(2*T-1) = 3
  200 CONTINUE
C
      IF (ICDRLS.EQ.2) GO TO 240
      WRITE (6,605)
      WRITE (6,610)
      DO 220 T = 1,BIGT
      II = 2*T-1
  220 WRITE (6,640) T,NREG(T),WRGAVL(T),NOVR(T),WOVAVL(T),SUBCON(T),
     1NHFIR1(II),IDLE(T)
      RETURN
  240 WRITE (6,630)
      DO 250 T = 1,BIGT
      II = 2*T-1
  250 WRITE (6,640) T,NRGBAR(T),WRGAVB(T),NOVBAR(T),WOVAVB(T),
     1SBCNBR(T),NHFIR1(II),IDLE(T)
C
C
  260 DFRNCH = 0.0
      DFRNCF = 0.0
      DO 300 T = 1,BIGT
      IF (NHFIR1(2*T-1).EQ.2.OR.NHFIR1(2*T-1).EQ.3)
     1                        DFRNCH = DFRNCH + HRFIR2(T)
      IF (NHFIR1(2*T-1).EQ.1.OR.NHFIR1(2*T-1).EQ.3)
     1                        DFRNCF = DFRNCF + HRFIR2(T)
```

```
  300 CONTINUE
      WRITE (6,650) DFRNCH,DFRNCF
C
      DFRNMX = DFRNCH
      IF (DFRNCF.GT.DFRNCH) DFRNMX = DFRNCF
      DO 330 T = 1,BIGT
      IF (NHFIR1(2*T-1).EQ.1) HRFIR2(T) = DFRNCF
      IF (NHFIR1(2*T-1).EQ.2) HRFIR2(T) = DFRNCH
      IF (NHFIR1(2*T-1).EQ.3) HRFIR2(T) = DFRNMX
  330 CONTINUE
      ICBYPS = 0
C
  605 FORMAT (//,15X,'OPTIONS SET AT THE BEGINING',//)
  610 FORMAT (1H ,/,15X,'PERIOD',8X,'NREG(T)',10X,'NOVR(T)',7X,
     1'SUBCON(T)',2X,'HIRFIR',2X,'IDLE(T)',/)
  630 FORMAT (1H ,15X,'OPTIONS SET AFTER APPLYING',/,
     116X,'RULES OF EFFICIENT UTILIZATION OF RESOURCES',//)
  640 FORMAT (1H ,14X,I4,5X,I4,'(',F7.2,')',4X,I4,'(',F7.2,')',5X,F7.2,
     14X,I4,5X,I4,/)
  650 FORMAT (/,15X,'TOTAL HIRING COULD BE = ',F7.2,/,
     1          15X,'TOTAL FIRING COULD BE = ',F7.2,//)
      RETURN
      END
C
C     ******************************************************************
      SUBROUTINE QUANTY(NT,AMNT)
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL350/DMND(12)
      COMMON/BL360/ICBYPS,ICDRLS,ICDFRG,ICSBXS,IXPRNO
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL440/OVAVLS,SBCNLS,UPLMWF,WOAVLS,WRGALS,
     1             RGAVLS,DMNLST,NOVLST
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                              WRGAVB(12)
      COMMON/BL510/OVRAVL(12),REGAVL(12),SUBCON(12),WOVAVL(12),
     1                                              WRGAVL(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
      ICDFRG = 0
      IF (ICSBXS.EQ.1) GO TO 135
      N1 = NREG(NT)
      IF (ICDRLS.EQ.2) N1 = NRGBAR(NT)
      IF (NT.EQ.1) GO TO 10
      N2 = NREG(NT-1)
      IF (ICDRLS.EQ.2) N2 = NRGBAR(NT-1)
      GO TO 20
   10 N2 = NGRBGN
   20 IF (N1.EQ.N2) GO TO 30
      GO TO 130
   30 W1 = WRGAVL(NT)
      IF (ICDRLS.EQ.2) W1 = WRGAVB(NT)
      IF (NT.EQ.1) GO TO 40
      W2 = WRGAVL(NT-1)
      IF (ICDRLS.EQ.2) W2 = WRGAVB(NT-1)
      GO TO 50
   40 W2 = WRGALS
```

```
   50 IF (W1.EQ.W2) GO TO 60
      GO TO 115
   60 N1 = NOVR(NT)
      IF (ICDRLS.EQ.2) N1 = NOVBAR(NT)
      IF (NT.EQ.1) GO TO 70
      N2 = NOVR(NT-1)
      IF (ICDRLS.EQ.2) N2 = NOVBAR(NT-1)
      GO TO 80
   70 N2 = NOVLST
   80 IF (N1.EQ.N2) GO TO 90
      GO TO 135
   90 W1 = WOVAVL(NT)
      IF (ICDRLS.EQ.2) W1 = WOVAVB(NT)
      IF (NT.EQ.1) GO TO 100
      W2 = WOVAVL(NT-1)
      IF (ICDRLS.EQ.2) W2 = WOVAVB(NT-1)
      GO TO 110
  100 W2 = WOAVLS
  110 IF (W1.EQ.W2) GO TO 120
  115 AMNT = DABS(W1-W2)
      ICDFRG = 1
      GO TO 150
  120 AMNT = 0.0
      ICDFRG = 0
      GO TO 150
  130 ICDFRG = 2
  135 IF (NT.EQ.1) GO TO 140
      AMNT = DABS(DMND(NT)-DMND(NT-1))
      GO TO 145
  140 AMNT = DABS(DMND(NT)-DMNLST)
  145 AMNT = AMNT/REGPRD(NREG(NT),NT)
      IF (ICDRLS.EQ.2) AMNT = AMNT/REGPRD(NRGBAR(NT),NT)
  150 RETURN
      END
C
C     *****************************************************************
      SUBROUTINE STRCTR
C     THIS SUBROUTINE PERFORMS STAGE 2 OF CONSTRUCTION OF THE
C     WORKING MODEL.
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT,T
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL50/M,N,L
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL330/AIZERO,CSTINV,CSTSB,EFCNCY,FACTOR,FRCTN,RGPAY(2)
      COMMON/BL350/DMND(12)
      COMMON/BL355/FRRATE,HRRATE
      COMMON/BL360/ICBYPS,ICDRLS,ICDFRG,ICSBXS,IXPRNO
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL380/ICNSFR(12),ICNSHR(12),ICNSID(12),ICNSOB(12),
     1             ICNSOV(24),ICNSWF(12)
      COMMON/BL385/IDOLFR(12),IDOLHR(12),IDOLID(12),IDOOBJ(12),
     1             IDOLOV(24),IDOLWF(12)
      COMMON/BL390/ICNTFR,ICNTHR,ICNTIC,ICNTIV,ICNTOV,ICNTSB
      COMMON/BL395/IPRDHR(12),IPRDFR(12),IPRDOV(24),IPRDID(12)
      COMMON/BL400/IVARFR(12),IVARHR(12),IVARID(12),IVARIN(12)
      COMMON/BL410/IVAROV(24),IVARPR(12),IVARRG(24),IVARSB(12)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
```

```
      COMMON/BL440/OVAVLS,SBCNLS,UPLMWF,WOAVLS,WRGALS,
     1             RGAVLS,DMNLST,NOVLST
      COMMON/BL450/OVRTM(12),WIDLE(12),WRKBGN(12),NOVR(12),NREG(12)
      COMMON/BL480/NUMCNS,NUMVAR
      COMMON/BL500/OVAVBR(12),RGAVBR(12),SBCNBR(12),WOVAVB(12),
     1                                 WRGAVB(12)
      COMMON/BL520/OVRPRD(2,12),REGPRD(2,12)
      COMMON/BL550/RHSHIR(2,12),RHSFIR(2,12),RHSOVR(2,24),RHSIDL(2,12),
     1             RHSWFC(2,12),RHSOBJ(2,5)
C
C   ICNTHR = NUMBER OF POSSIBLE OCCURENCE OF HIRING
C     ICNSHR(I) = CONSTRAINT NUMBER CORRESPONDING TO I-TH HIRING
C     IVARHR(I) = VARIABLE WHT CORRESPONDING TO THE ICNSHR(I)
C     IPRDHR(I) = PERIOD CORRESPONDING TO I-TH HIRING
C     IDOLHR(I) = DUAL VARIABLE CORRESPONDING TO ICNSHR(I)
C         FOR OTHER OPTIONS SIMILAR VARIABLES ARE PRESENT
C         SB(I) = RIGHT HAND SIDE OF THE CONSTRAINT I
C A(I,J) = MATRIX A
C       CC(I,J) = J-TH COST COEFFICIENT OF I-TH OBJECTIVE FUNCTION
C
      ICNTHR = 0
      ICNTFR = 0
      ICNTWF = 0
      ICNTOV = 0
      ICNTSB = 0
      ICNTIC = 0
      ICNTIV = 0
      LSTCNS = 1
      LSTVAR = 0
      NEWCNS = 0
      NEWVAR = 0
      NUMCNS = 0
      NUMVAR = 0
C
      DO 5 I = 1,10
      DO 5 J = 1,175
    5 CC(I,J) = 0.0
      DO 7 I = 1,110
      DO 7 J = 1,175
    7 A(I,J) = 0.0
      SB(1) = - AIZERO
C     ----------------------------------------------------------
C
      DO 1000 T = 1,BIGT
      SB(1) = SB(1)+DMND(T)
      A(1,LSTVAR+1) = 1.0
      A(LSTCNS+1,LSTVAR+1) = 1.0
      A(LSTCNS+1,LSTVAR+2) = -REGPRD(1,T)
      A(LSTCNS+1,LSTVAR+3) = - REGPRD(2,T)
      IVARPR(T) = LSTVAR+1
      SB(LSTCNS+1) = 0.0
      A(LSTCNS+2,LSTVAR+2) = 1.0
      CC(1,LSTVAR+2) = RGPAY(1)
      CC(5,LSTVAR+2) = RGPAY(1)
      IF (T.GT.1) GO TO 10
      SB(LSTCNS+2) = WRKBGN(1)
      GO TO 20
   10 A(LSTCNS+2,LSTW1) = -1.0
```

```
      SB(LSTCNS+2) = 0.0
   20 LSTW1 = LSTVAR+2
      IVARRG(2*T-1) = LSTW1
      A(LSTCNS+3,LSTVAR+3) = 1.0
      CC(1,LSTVAR+3) = RGPAY(2)
      CC(5,LSTVAR+3) = RGPAY(2)
      IF (T.GT.1) GO TO 30
      SB(LSTCNS+3) = WRKBGN(2)
      GO TO 40
   30 A(LSTCNS+3,LSTW2) = -1.0
      SB(LSTCNS+3) = 0.0
   40 LSTW2 = LSTVAR+3
      IVARRG(2*T) = LSTW2
      A(LSTCNS+4,LSTVAR+1) = 1.0
      A(LSTCNS+4,LSTVAR+4) = -1.
      IF (T.GT.1) GO TO 50
      SB(LSTCNS+4) = DMND(T)-AIZERO
      GO TO 60
   50 A(LSTCNS+4,LSTINV) = 1.0
      SB(LSTCNS+4) = DMND(T)
   60 LSTINV = LSTVAR+4
      IVARIN(T) = LSTINV
      CC(2,LSTINV) = CSTINV
      CC(5,LSTINV) = CSTINV
         CC(6,LSTINV) = CSTINV
C
      IF (IXPRNO.EQ.1) GO TO 150
C     ----------------------------------------------------------
C     IF FIRING EXTENDS TO ONE CLASS
      IF (NHFIR1(2*T-1).EQ.1.AND.NHFIR1(2*T).EQ.1) THEN
         A(LSTCNS+3,LSTVAR+5) = 1.0
         NEWCNS = LSTCNS+5
         NEWVAR = LSTVAR+5
         A(NEWCNS,NEWVAR) = 1.0
         SB(NEWCNS) = HRFIR2(T)
         CC(1,NEWVAR) = FRRATE
         CC(5,NEWVAR) = FRRATE
         ICNTFR = ICNTFR+1
         ICNSFR(ICNTFR) = NEWCNS
         IVARFR(ICNTFR) = NEWVAR
         IPRDFR(ICNTFR) = T
         RHSFIR(1,ICNTFR) = SB(NEWCNS)
         RHSFIR(2,ICNTFR) = SB(NEWCNS)
         ENDIF
C     ----------------------------------------------------------
C     IF FIRING EXTENDS TO BOTH CLASSES
      IF (NHFIR1(2*T-1).EQ.1.AND.NHFIR1(2*T).EQ.2) THEN
         A(LSTCNS+3,LSTVAR+5) = 1.0
         NEWVAR = LSTVAR+6
         A(LSTCNS+2,NEWVAR) = 1.0
         NEWCNS = LSTCNS+5
         A(NEWCNS,LSTVAR+5) = 1.0
         SB(NEWCNS) = HRFIR2(T)
         CC(1,LSTVAR+5) = FRRATE
         CC(5,LSTVAR+5) = FRRATE
         CC(5,LSTVAR+6) = FRRATE-2.00
         ICNTFR = ICNTFR+1
         ICNSFR(ICNTFR) = NEWCNS
```

```
         IVARFR(ICNTFR) = LSTVAR+5
         IPRDFR(ICNTFR) = T
         RHSFIR(1,ICNTFR) = SB(NEWCNS)
         RHSFIR(2,ICNTFR) = SB(NEWCNS)
         ENDIF
C     -----------------------------------------------------------
C     IF HIRING IS LIKELY
      IF (NHFIR1(2*T-1).EQ.2) THEN
         A(LSTCNS+3,LSTVAR+5) = -1.0
         NEWCNS = LSTCNS+5
         NEWVAR = LSTVAR+5
         A(NEWCNS,NEWVAR) = 1.0
         SB(NEWCNS) = HRFIR2(T)
         CC(1,LSTVAR+5) = HRRATE
         CC(5,LSTVAR+5) = HRRATE
         ICNTHR = ICNTHR+1
         ICNSHR(ICNTHR) = NEWCNS
         IVARHR(ICNTHR) = NEWVAR
         IPRDHR(ICNTHR) = T
         RHSHIR(1,ICNTHR) = SB(NEWCNS)
         RHSHIR(2,ICNTHR) = SB(NEWCNS)
         ENDIF
C     -----------------------------------------------------------
C     IF HIRING AND FIRING BOTH ARE LIKELY AND FIRING EXTENDS TO
C     CLASS ONE ONLY
  100 IF (NHFIR1(2*T-1).EQ.3.AND.NHFIR1(2*T).EQ.1) THEN
         A(LSTCNS+3,LSTVAR+5) = -1.0
         A(LSTCNS+3,LSTVAR+6) = 1.0
         NEWCNS = LSTCNS+5
         NEWVAR = LSTVAR+5
         A(NEWCNS,NEWVAR) = 1.0
         SB(NEWCNS) = HRFIR2(T)
         CC(1,NEWVAR) = HRRATE
         CC(5,NEWVAR) = HRRATE
         ICNTHR = ICNTHR+1
         ICNSHR(ICNTHR)= NEWCNS
         IVARHR(ICNTHR) = NEWVAR
         IPRDHR(ICNTHR) = T
         RHSHIR(1,ICNTHR) = SB(NEWCNS)
         RHSHIR(2,ICNTHR) = SB(NEWCNS)
C
         NEWCNS = NEWCNS+1
         NEWVAR = NEWVAR+1
         A(NEWCNS,NEWVAR) = 1.0
         SB(NEWCNS) = HRFIR2(T)
         CC(1,NEWVAR) = FRRATE
         CC(5,NEWVAR) = FRRATE
         ICNTFR = ICNTFR+1
         ICNSFR(ICNTFR) = NEWCNS
         IVARFR(ICNTFR) = NEWVAR
         IPRDFR(ICNTFR) = T
         RHSFIR(1,ICNTFR) = SB(NEWCNS)
         RHSFIR(2,ICNTFR) = SB(NEWCNS)
         ENDIF
C     -----------------------------------------------------------
C     IF HIRING AND FIRING BOTH ARE LIKELY AND FIRING
C     EXTENDS TO BOTH CLASSES
  150 IF (NHFIR1(2*T-1).EQ.3.AND.NHFIR1(2*T).EQ.2) THEN
```

```
          A(LSTCNS+2,LSTVAR+7) = 1.0
          A(LSTCNS+3,LSTVAR+5) = -1.0
          A(LSTCNS+3,LSTVAR+6) = 1.0
          NEWCNS = LSTCNS+5
          NEWVAR = LSTVAR+5
          A(NEWCNS,NEWVAR) = 1.0
          SB(NEWCNS) = HRFIR2(T)
          CC(1,NEWVAR) = HRRATE
          CC(5,NEWVAR) = HRRATE
              CC(6,NEWVAR) = 0.20
              CC(7,NEWVAR) = 0.20
              CC(8,NEWVAR) = 0.20
          ICNTHR = ICNTHR+1
          ICNSHR(ICNTHR) = NEWCNS
          IVARHR(ICNTHR) = NEWVAR
          IPRDHR(ICNTHR) = T
          RHSHIR(1,ICNTHR) = SB(NEWCNS)
          RHSHIR(2,ICNTHR) = SB(NEWCNS)
C
          NEWCNS = NEWCNS+1
          NEWVAR= NEWVAR+1
          A(NEWCNS,NEWVAR) = 1.0
          SB(NEWCNS) = HRFIR2(T)
          CC(1,NEWVAR) = FRRATE
          CC(5,NEWVAR) = FRRATE
                CC(6,NEWVAR) = 0.10
                CC(7,NEWVAR) = 0.10
                CC(8,NEWVAR) = 0.10
          CC(5,LSTVAR+7) = FRRATE -2.00
          ICNTFR = ICNTFR+1
          ICNSFR(ICNTFR) = NEWCNS
          IVARFR(ICNTFR) = NEWVAR
          IPRDFR(ICNTFR) = T
          RHSFIR(1,ICNTFR) = SB(NEWCNS)
          RHSFIR(2,ICNTFR) = SB(NEWCNS)
          ENDIF
C     -----------------------------------------------------------
C         IF IDLE TIME IS LIKELY FOR CLASS ONE ONLY
  200 IF (IDLE(T).EQ.1) THEN
          NEWCNS = NEWCNS+1
          NEWVAR = NEWVAR+1
          A(LSTCNS+1,NEWVAR) = REGPRD(2,T)
          A(NEWCNS,NEWVAR) = 1.0
          SB(NEWCNS) = WIDLE(T)
          CC(3,NEWVAR) = 1.0
               CC(7,NEWVAR) = 1.0
          ICNTIC = ICNTIC+1
          ICNSID(ICNTIC) = NEWCNS
          IPRDID(ICNTIC) = T
          RHSIDL(1,ICNTIC) = SB(NEWCNS)
          RHSIDL(2,ICNTIC) = SB(NEWCNS)
          ICNTIV = ICNTIV+1
          IVARID(ICNTIV) = NEWVAR
          ENDIF
      IF (IXPRNO.EQ.1) GO TO 300
C     -----------------------------------------------------------
C     IF IDLE TIME IS LIKELY FOR BOTH CLASSES
      IF (IDLE(T).EQ.2) THEN
```

```
            NEWCNS = NEWCNS+1
            NEWVAR = NEWVAR+1
            A(LSTCNS+1,NEWVAR) = REGPRD(2,T)
            A(NEWCNS,NEWVAR) = 1.0
            CC(3,NEWVAR) = 1.0
            ICNTIV = ICNTIV+1
            IVARID(ICNTIV) = NEWVAR
            NEWVAR = NEWVAR+1
            A(LSTCNS+1,NEWVAR) = REGPRD(1,T)
            A(NEWCNS,NEWVAR) = 1.0
            SB(NEWCNS) = WIDLE(T)
            ICNTIV = ICNTIV+1
            IVARID(ICNTIV) = NEWVAR
            CC(3,NEWVAR) = 1.0
            ICNTIC = ICNTIC+1
            ICNSID(ICNTIC) = NEWCNS
            IPRDID(ICNTIC) = T
            RHSIDL(1,ICNTIC) = SB(NEWCNS)
            RHSIDL(2,ICNTIC) = SB(NEWCNS)
            ENDIF
C     ------------------------------------------------------------
C           IF OVERTIME IS LIKELY FOR ONE CLASS
      IF (NOVBAR(T).EQ.1) THEN
            NEWCNS = NEWCNS+1
            NEWVAR = NEWVAR+1
            A(1,NEWVAR) = 1.0
            A(LSTCNS+4,NEWVAR) = 1.0
            A(NEWCNS,NEWVAR) = 1.0
            A(NEWCNS,LSTW1) = - EFCNCY*FRCTN*REGPRD(1,T)
            SB(NEWCNS) = 0.0
            CC(4,NEWVAR) = RGPAY(1)*FACTOR/OVRPRD(1,T)
            CC(5,NEWVAR) =   CC(4,NEWVAR)
                  CC(8,NEWVAR) =   CC(4,NEWVAR)
            ICNTOV = ICNTOV+1
            ICNSOV(ICNTOV) = NEWCNS
            IVAROV(ICNTOV) = NEWVAR
            IPRDOV(ICNTOV) = T
            RHSOVR(1,ICNTOV) = SB(NEWCNS)
            RHSOVR(2,ICNTOV) = SB(NEWCNS)
            ENDIF
C     ------------------------------------------------------------
C     IF OVERTIME IS LIKELY FOR TWO CLASSES
  300 IF (NOVBAR(T).EQ.2) THEN
            NEWCNS = NEWCNS+1
            NEWVAR = NEWVAR+1
            A(1,NEWVAR) = 1.0
            A(LSTCNS+4,NEWVAR) = 1.0
            A(NEWCNS,NEWVAR) = 1.0
            A(NEWCNS,LSTW1) = - EFCNCY*FRCTN*REGPRD(1,T)
            SB(NEWCNS) = 0.0
            ICNTOV = ICNTOV+1
            ICNSOV(ICNTOV) = NEWCNS
            IVAROV(ICNTOV) = NEWVAR
            IPRDOV(ICNTOV) = T
            RHSOVR(1,ICNTOV) = SB(NEWCNS)
            RHSOVR(2,ICNTOV) = SB(NEWCNS)
            CC(4,NEWVAR) = RGPAY(1)*FACTOR/OVRPRD(1,T)
            CC(5,NEWVAR) =   CC(4,NEWVAR)
```

```
             CC(8,NEWVAR) =   CC(4,NEWVAR)
C
          NEWCNS = NEWCNS+1
          NEWVAR = NEWVAR+1
          A(1,NEWVAR) = 1.0
          A(LSTCNS+4,NEWVAR) = 1.0
          A(NEWCNS,NEWVAR) = 1.0
          A(NEWCNS,LSTW2) = - EFCNCY*FRCTN*REGPRD(2,T)
          SB(NEWCNS) = 0.0
          CC(4,NEWVAR) = RGPAY(2)*FACTOR/OVRPRD(2,T)
          CC(5,NEWVAR) =    CC(4,NEWVAR)
              CC(8,NEWVAR) =   CC(4,NEWVAR)
          ICNTOV = ICNTOV+1
          ICNSOV(ICNTOV) = NEWCNS
          IVAROV(ICNTOV) = NEWVAR
          IPRDOV(ICNTOV) = T
          RHSOVR(1,ICNTOV) = SB(NEWCNS)
          RHSOVR(2,ICNTOV) = SB(NEWCNS)
          ENDIF
C
      IF (IXPRNO.LT.3) GO TO 400
C     -----------------------------------------------------------
C     IF SUBCONTRACTING IS LIKELY
      IF (SBCNBR(T).GT.0.0) THEN
         NEWVAR = NEWVAR+1
         A(1,NEWVAR) = 1.0
         A(LSTCNS+4,NEWVAR) = 1.0
         ICNTSB = ICNTSB+1
         IVARSB(ICNTSB) = NEWVAR
         CC(5,NEWVAR) = CSTSB
         ENDIF
C     -----------------------------------------------------------
  400 NEWCNS = NEWCNS+1
      A(NEWCNS,LSTVAR+2) = 1.0
      A(NEWCNS,LSTVAR+3) = 1.0
      SB(NEWCNS) = UPLMWF
      ICNSWF(T) = NEWCNS
      RHSWFC(1,T) = UPLMWF
      RHSWFC(2,T) = UPLMWF
      LSTCNS = NEWCNS
      LSTVAR = NEWVAR
 1000 CONTINUE
C     ----------------------------------------------------------
C     ADD SLACK VARIABLES TO INEQUALITY CONSTRAINTS
C     ALSO IDENTIFIES LOCATIONS OF DUAL VARIABLES
C     CORRESPONDING THE INEQUALITY CONSTRAINTS
      WRITE (6,810) LSTCNS,LSTVAR
      NUMVAR = LSTVAR
      DO 500 I = 1,ICNTHR
      LOCATE = ICNSHR(I)
      NUMVAR = NUMVAR+1
      IDOLHR(I) = NUMVAR
      A(LOCATE,NUMVAR) = 1.0
  500 CONTINUE
C
      IF (ICNTFR.EQ.0) GO TO 515
      DO 510 I = 1,ICNTFR
      LOCATE = ICNSFR(I)
```

```
      NUMVAR = NUMVAR+1
      IDOLFR(I) = NUMVAR
      A(LOCATE,NUMVAR) = 1.0
  510 CONTINUE
  515 IF (ICNTIC.EQ.0) GO TO 525
      DO 520 I = 1,ICNTIC
      LOCATE = ICNSID(I)
      NUMVAR = NUMVAR+1
      IDOLID(I) = NUMVAR
      A(LOCATE,NUMVAR) = 1.0
  520 CONTINUE
  525 DO 530 I = 1,ICNTOV
      LOCATE = ICNSOV(I)
      NUMVAR = NUMVAR+1
      IDOLOV(I) = NUMVAR
      A(LOCATE,NUMVAR) = 1.0
  530 CONTINUE
      DO 540 I = 1,BIGT
      LOCATE = ICNSWF(I)
      NUMVAR = NUMVAR+1
      IDOLWF(I) = NUMVAR
      A(LOCATE,NUMVAR) = 1.0
  540 CONTINUE
      LSTVAR = NUMVAR
C     ----------------------------------------------------------
C     ADDS CONSTRAINTS FORMED BY THE OBJECTIVE FUNCTIONS.
C     ALSO IDENTIFIES DUAL VARIABLES CORRESPONDING TO THEM.
C
      NUMOBJ = 5
      DO 560 I = 1,NUMOBJ
      NEWCNS = NEWCNS+1
      ICNSOB(I) = NEWCNS
      DO 550 J = 1,LSTVAR
      A(NEWCNS,J) = CC(I,J)
  550 CONTINUE
      NUMVAR = NUMVAR+1
      IDOOBJ(I) = NUMVAR
      A(NEWCNS,NUMVAR) = 1.0
  560 CONTINUE
C
      SB(LSTCNS+1) = 800000.00
      RHSOBJ(1,1) = 800000.00
      SB(LSTCNS+2) = 90000.0
      RHSOBJ(1,2) = 90000.0
      SB(LSTCNS+3) = 40.0
      RHSOBJ(1,3) = 40.0
      SB(LSTCNS+4) = 100000.00
      RHSOBJ(1,4) = 100000.00
      SB(LSTCNS+5) = 900000.00
      RHSOBJ(1,5) = 900000.00
      NUMCNS = NEWCNS
C     ----------------------------------------------------------
         WRITE (6,610)
         II = 2*BIGT
         WRITE (6,640) (NHFIR1(T),T=1,II)
         WRITE (6,620)
         WRITE (6,640) (IDLE(T),T=1,BIGT)
         WRITE (6,630)
```

```
      WRITE (6,640) (NOVBAR(T),T=1,BIGT)
      WRITE (6,650)
      WRITE (6,660) (SBCNBR(T),T=1,BIGT)
      WRITE (6,670)
      WRITE (6,800) (ICNSHR(T),T=1,ICNTHR)
      WRITE (6,680)
      WRITE (6,800) (IVARHR(T),T=1,ICNTHR)
      IF (ICNTFR.EQ.0) GO TO 565
      WRITE (6,690)
      WRITE (6,800) (ICNSFR(T),T=1,ICNTFR)
      WRITE (6,700)
      WRITE (6,800) (IVARFR(T),T=1,ICNTFR)
565   WRITE (6,710)
      WRITE (6,800) (ICNSOV(T),T=1,ICNTOV)
      WRITE (6,720)
      WRITE (6,800) (IVAROV(T),T=1,ICNTOV)
      IF (ICNTIC.EQ.0) GO TO 570
      WRITE (6,730)
      WRITE (6,800) (ICNSID(T),T=1,ICNTIC)
      WRITE (6,740)
      WRITE (6,800) (IVARID(T),T=1,ICNTIV)
570   IF (IXPRNO.LE.2.OR.ICNTSB.EQ.0) GO TO 580
      WRITE (6,750)
      WRITE (6,800) (IVARSB(T),T=1,ICNTSB)
580   WRITE (6,810) NUMCNS,NUMVAR
C     -----------------------------------------------------
      ILEKHO = 0
      IF (ILEKHO.EQ.1) THEN
         WRITE (6,815)
         IOB = 5
         DO 585 I = 1,IOB
         WRITE (6,825) I
         WRITE (6,830) (CC(I,J),J=1,NUMVAR)
585      CONTINUE
         WRITE (6,820)
         DO 590 I = 1,12
         WRITE (6,835) I
         WRITE (6,830) (A(I,J),J=1,NUMVAR)
590      CONTINUE
         WRITE (6,840)
         WRITE (6,830) (SB(J),J=1,LSTCNS)
         L1 = LSTCNS+1
         L2 = LSTCNS+4
         WRITE (6,845) (SB(J),J=L1,L2)
         ENDIF
      M = NUMCNS
      N = NUMVAR
      DO 600 J = 1,N
600   C(J) = CC(5,J)
C     -----------------------------------------------------
610   FORMAT (15X,'NHFIR1(2T-1) AND NHFIR1(2T)',/)
620   FORMAT (//,15X,'IDLE(T)',/)
630   FORMAT (//,15X,'NOVBAR(T)',/)
640   FORMAT (17X,6I4,/)
650   FORMAT (//,15X,'SBCNBR(T)',/)
660   FORMAT (//,17X,6F6.2,/)
670   FORMAT (//,15X,'CONSTRAINTS NOS. RELATED TO HIRING :',//)
680   FORMAT (//,15X,'VARIABLE NOS. RELATED TO  HIRING :',//)
```

```
  690 FORMAT (//,15X,'CONSTRAINT NOS. RELATED TO FIRING :',//)
  700 FORMAT (//,15X,'VARIABLE NOS. RELATED TO FIRING :',//)
  710 FORMAT (//,15X,'CONSTRAINT NOS. RELATED TO OVERTIME :',//)
  720 FORMAT (//,15X,'VARIABLE NOS. RELATED TO OVERTIME :',//)
  730 FORMAT (//,15X,'COSTRAINT NOS. RELATED TO IDLE TIME : ',//)
  740 FORMAT (//,15X,'VARIABLES RELATED TO IDLE TIME :',//)
  750 FORMAT (//,15X,'VARIABLE NOS. RELATED TO SUBCON :',//)
  800 FORMAT (17X,4I4,/)
  810 FORMAT (//,15X,'TOTAL NUMBER OF CONSTRAINTS :',I4,/,15X,
     1'TOTAL NUMBER OF VARIABLES    :',I4,//)
  815 FORMAT (//,15X,'COST INFORMATION :',//)
  820 FORMAT (//,15X,'MATRIX A(I,J) :',//)
  825 FORMAT (//,15X,'COST COEFF. FOR THE OBJ. FUNCTION : ',I4,/)
  830 FORMAT (4X,16F7.2,/)
  835 FORMAT (//,15X,'ELEMENTS OF THE ROW : ',I4,/)
  840 FORMAT (/,16X,'RIGHT HAND SIDE VALUES : ',/)
  845 FORMAT (/,6X,'OBJECTIVE FUNC. LIMIT',/,6X,4F12.2,/)
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE LNRPRG
C     THIS SUBROUTINE SOLVES THE MODEL BY USING REVISED SIMPLEX METHOD
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL40/IC,IFC,ITN
      COMMON/BL50/M,N,L
      ILEKHO = 0
      IF (ILEKHO.EQ.1) THEN
      WRITE(6,201)
      WRITE (6,200)
      WRITE(6,204)
      WRITE(6,205) (C(I),I=1,N)
      WRITE(6,203)
      WRITE(6,205) (SB(I),I=1,M)
      WRITE(6,200)
      WRITE(6,207) M
      WRITE(6,208) N
      ENDIF
      L=0
      DO 27 J=1,N
   27 IN(J)=1000+J
      CALL PHASE1
      CALL RIVSIM
      IF (IC.EQ.1) GO TO 2000
      CALL PHASE2
      IF (IFC.EQ.1) GO TO 1000
      CALL RIVSIM
      IF (IC.EQ.1) GO TO 2000
      CALL OPTIMA
      RETURN
 1000 WRITE (6,211)
 2000 STOP
  200 FORMAT (//)
  201 FORMAT (1H ,55X,'***  DATA  ***'/)
  203 FORMAT (//1H ,50X,'***  CONSTANT TERMS  ***'//)
```

```
  204 FORMAT (1H ,50X,'***  COST CO-EFFICIENTS  ***'//)
  205 FORMAT (1H ,20X,7F10.2)
  207 FORMAT (1H ,15X,'***  NUMBER OF CONSTRAINTS  ***',I4//)
  208 FORMAT (1H ,15X,'***  NUMBER OF  VARIABLES  ***',I4//)
  211 FORMAT (1H ,20X,'***  INFEASIBLE  ***'//)
      END
C
C     *****************************************************************
      SUBROUTINE PHASE1
C     THIS SUBROUTINE PREPARES FOR THE PHASE 1 OF REVS. SIMPLX. METHOD.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      EPS=1.0E-4
      NL=N+L
      DO 12 I=1,M
      IF (SB(I).GT.-EPS) GO TO 12
      SB(I)=-SB(I)
      DO 11 J=1,NL
      A(I,J)=-A(I,J)
   11 CONTINUE
   12 CONTINUE
      DO 13 I=1,M
      DO 13 J=1,M
   13 BI(I,J)=0.00
      DO 14 I=1,NL
   14 WM(2,I)=0.00
      WRITE (6,100)
      DO 15 I=1,M
      I1=I+N+L
      IB(I)=I1
      BI(I,I)=1.00
      BE(I)=SB(I)
      WM(3,I)=1.00
   15 CONTINUE
  100 FORMAT (///,10X,'***  PHASE ONE  ***'//)
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE PHASE2
C     THIS SUBROUTINE PREPARES FOR THE PHASE 2 OF REVS. SIMPLX. METHOD
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL40/IC,IFC,ITN
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      IFC=0
      NL=N+L
C       FEASIBILITY TEST
      DO 10 I=1,M
      IF (IB(I).LT.1000) GO TO 1000
   10 CONTINUE
```

```
      WRITE (6,602)
      WRITE (6,603)
      DO 11 I=1,NL
      WM(2,I)=C(I)
   11 CONTINUE
      DO 14 I=1,M
      DO 12 J=1,NL
      IF (IB(I).EQ.IN(J)) GO TO 13
   12 CONTINUE
   13 WM(3,I)=WM(2,J)
   14 CONTINUE
      RETURN
 1000 WRITE(6,601)
      DO 15 I=1,M
   15 WRITE (6,604) IB(I),BE(I)
      IFC=1
  601 FORMAT (///25X,'***  INFEASIBLE  ***'/)
  602 FORMAT (///25X,'***  FEASIBLE  ***'/)
  603 FORMAT (///10X,'***  PHASE 2  ***'/)
  604 FORMAT (/1H ,25X,2HX(,I4,2H)=,F10.5)
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE RIVSIM
C     THIS SUBROUTINE CHECKS THE OPTIMALITY CONDITION, THE PRESENCE
C     OF UNBOUNDED SOLUTION ETC.  IT ALSO DETERMINES THE VARIABLE
C     TO LEAVE THE CURRENT BASIS AND THE VARIBLE TO ENTER THE NEW BASIS.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL40/IC,IFC,ITN
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      EPS=1.0E-4
      ITN=200
      IC=0
      NL=N+L
C     CALL BASIVA
      IT = 1
 1200 CONTINUE
C BOX 1
      DO 10 J=1,M
      WM(1,J)=0.00
      DO 10 I=1,M
      WM(1,J)=WM(1,J)+WM(3,I)*BI(I,J)
   10 CONTINUE
C  BOX 2      W(J)=C BAR(J)
      DO 12 J=1,NL
      W(J)=0.00
      Y1(J)=0.00
      DO 11 I=1,M
      W(J)=W(J)+WM(1,I)*A(I,J)
   11 CONTINUE
      Y1(J)=W(J)
      W(J)=W(J)-WM(2,J)
   12 CONTINUE
```

```
      DO 13 J=1,NL
   13 Y2(J)=W(J)
C BOX 3: DETERMINES THE VARIABLE TO ENTER THE NEW BASIS
      HIGH=-1.0E5
      DO 15 J=1,NL
      DO 14 I=1,M
      IF (IB(I).EQ.IN(J)) GO TO 15
   14 CONTINUE
      IF (W(J).LE.HIGH) GO TO 15
      HIGH=W(J)
      JS=J
   15 CONTINUE
      IF (HIGH.LE.EPS) GO TO 1000
C  BOX 5     Y(I)=AS(I,JS)
      DO 16 I=1,M
      Y(I)=0.00
      DO 16 J=1,M
      Y(I)=Y(I)+BI(I,J)*A(J,JS)
   16 CONTINUE
C  BOX  6
      DO 17 I=1,M
      IF (Y(I).GE.EPS) GO TO 18
   17 CONTINUE
      GO TO 1100
C     DETERMINES THE VARIABLE TO LEAVE THE CURRENT BASIS
   18 ISTRT = I
      THTAMN = BE(I)/Y(I)
      IR = I
      DO 19 I = ISTRT,M
      IF (Y(I).LT.EPS) GO TO 19
      TH=BE(I)/Y(I)
      IF (THTAMN.LE.TH) GO TO 19
      THTAMN=TH
      IR=I
   19 CONTINUE
      IVER = 0
      IF (IVER.EQ.1) WRITE (6,606) IT,IR,JS,Y(IR),BE(IR)
  606 FORMAT (/,16X,'ITER=',I3,'IR=',I3,' JS=',I3,' Y(IR)=',F15.11,
     1' BE(IR) = ',F7.2,/)
      CALL PIVOT(IR,JS)
C  BOX 11 ITERATION AND PRINT
      IT=IT+1
      IF (IT.EQ.ITN) GO TO 1300
      GO TO 1200
 1000 CONTINUE
      DO 100 I=1,M
      IF (IB(I).GE.1000) GO TO 100
      IF (DABS(BE(I)).GE.EPS) RETURN
      IR=I
      DO 101 J=1,NL
      DO 200 JJ=1,M
      IF (IB(JJ).EQ.IN(J)) GO TO 101
  200 CONTINUE
      SUM=0.00
      DO 102 IJ=1,M
  102 SUM=SUM+BI(IR,IJ)*A(IJ,J)
      IF (DABS(SUM).GT.EPS) GO TO 103
  101 CONTINUE
```

```
      RETURN
  103 CONTINUE
      JS=J
      CALL PIVOT(IR,JS)
      IT=IT+1
      IF(IT.EQ.ITN) GO TO 1300
  100 CONTINUE
      RETURN
 1100 WRITE (6,601)
      WRITE (6,604)
      WRITE (6,605) (Y(J),J=1,M)
      IC=1
      RETURN
 1300 WRITE (6,603)
      STOP
  601 FORMAT (///25X,'***  INFINITE  ***'/)
  603 FORMAT (///25X,'***  ITERATION OVER  ***'/)
  604 FORMAT (1H ,25X,'***  Y(J)  ***'//)
  605 FORMAT (1H ,25X,7F10.5)
      END
C
C     *****************************************************************
      SUBROUTINE PIVOT(IR,JS)
C     THIS SUBROUTINE PERFORMS THE PIVOT OPERATION
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      EPS4=1.0E-8
      DO 120 I = 1,M
      Y(I) = 0.0
      DO 120 J = 1,M
      Y(I) = Y(I)+BI(I,J)*A(J,JS)
  120 CONTINUE
      IF (Y(IR).EQ.0.0.OR.DABS(Y(IR)).LE.1.0E-7) THEN
                WRITE (6,610) (Y(I),I=1,M)
                WRITE (6,610) (BE(I),I=1,M)
                ENDIF
  610 FORMAT (/,3X,8F15.10)
      DO 12 I=1,M
      IF (I.EQ.IR) GO TO 12
      WM(4,I)=-Y(I)/Y(IR)
   12 CONTINUE
      WM(4,IR)=1.00/Y(IR)
      DO 13 I=1,M
      IF (I.EQ.IR) GO TO 13
      BE(I)=BE(I)+WM(4,I)*BE(IR)
   13 CONTINUE
      BE(IR)=WM(4,IR)*BE(IR)
      EPS2=0.000005
      DO 14 I=1,M
      IF (BE(I).LT.EPS2) BE(I)=0.000000
   14 CONTINUE
      DO 16 J=1,M
      DO 15 I=1,M
      IF (I.EQ.IR) GO TO 15
```

```
      IF (DABS(WM(4,I)).LE.EPS4) WM(4,I)=0.00
      BI(I,J)=BI(I,J)+WM(4,I)*BI(IR,J)
   15 CONTINUE
      BI(IR,J)=WM(4,IR)*BI(IR,J)
   16 CONTINUE
      IB(IR)=IN(JS)
      WM(3,IR)=WM(2,JS)
      RETURN
      END
C
C     ******************************************************************
      SUBROUTINE OPTIMA
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      WRITE (6,601)
      CALL BASIVA
  601 FORMAT (///45X,'***  OPTIMAL SOLUTION  ***'/)
      RETURN
      END
C
C     ******************************************************************
      SUBROUTINE BASIVA
C     THIS SUBROUTINE PRINTS THE DETAILED RESULTS OF LINEAR PROGRAMMING
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      DO 20 I=1,M
      IF (IB(I).LT.1000) GO TO 10
      GO TO 20
   10 I2=IB(I)-N-L
      IB(I)=I2
   20 CONTINUE
      WRITE (6,600)
      WRITE (6,601) (IB(I),I=1,M)
      WRITE (6,602)
      WRITE (6,603) (BE(I),I=1,M)
      F=0.00
      DO 30 I=1,M
   30 F=F+WM(3,I)*BE(I)
      WRITE (6,605) F
  600 FORMAT (46X,'***  BASIC VARIABLES  ***'//)
  601 FORMAT (41X,7I5)
  602 FORMAT (//,40X,'***  VALUES OF THE BASIC VARIABLES  ***'//)
  603 FORMAT (16X,7F12.3)
  605 FORMAT (///,10X,'OBJECTIVE FUNCTION VALUE = ',F13.2/)
      RETURN
      END
C
C     ******************************************************************
      SUBROUTINE CSTCMP
C     THIS SUBROUTINE COMPUTES DIFFERENT COST COMPONENTS
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      INTEGER BIGT
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL20/BE(110),BI(110,110)
      COMMON/BL30/IB(110),IN(175)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL370/IDLE(12),NHFIR1(24),HRFIR2(12)
      COMMON/BL380/ICNSFR(12),ICNSHR(12),ICNSID(12),ICNSOB(12),
     1             ICNSOV(24),ICNSWF(12)
      COMMON/BL390/ICNTFR,ICNTHR,ICNTIC,ICNTIV,ICNTOV,ICNTSB
      COMMON/BL395/IPRDHR(12),IPRDFR(12),IPRDOV(24),IPRDID(12)
      COMMON/BL400/IVARFR(12),IVARHR(12),IVARID(12),IVARIN(12)
      COMMON/BL410/IVAROV(24),IVARPR(12),IVARRG(24),IVARSB(12)
      COMMON/BL430/NOVBAR(12),NRGBAR(12)
      COMMON/BL550/RHSHIR(2,12),RHSFIR(2,12),RHSOVR(2,24),RHSIDL(2,12),
     1             RHSWFC(2,12),RHSOBJ(2,5)
      COMMON/BL600/OPTVAL(8,12),OPTCST(8,12)
C     ------------------------------------------------------------
      DO 1000 II = 1,8
      DO 1000 JJ = 1,BIGT
      OPTVAL(II,JJ) = 0.0
      OPTCST(II,JJ) = 0.0
 1000 CONTINUE
C     ------------------------------------------------------------
C     COMPUTATION OF COST OF REGULAR PAYROLL OVER THE HORIZON
      COSTRG = 0.0
      KK = 0
      DO 30 II = 1,BIGT
      JJ = 0
    5 JJ = JJ + 1
      KK = KK + 1
      KWNTED = IVARRG(KK) + 1000
      DO 10 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 20
   10 CONTINUE
      VAL1 = 0.0
      CST1 = 0.0
      GO TO 25
   20 VAL1 = BE(I)
      KWNTED = KWNTED - 1000
      CST1 = VAL1 * CC(5,KWNTED)
   25 OPTVAL(6,II) = OPTVAL(6,II) + VAL1
      OPTCST(6,II) = OPTCST(6,II) + CST1
      IF (JJ.EQ.2) GO TO 28
      GO TO 5
   28 COSTRG = COSTRG + OPTCST(6,II)
   30 CONTINUE
C     ------------------------------------------------------------
C     COMPUTATION OF COST DUE TO HIRING OVER HORIZON
      COSTHR = 0.0
      DO 60 II = 1,ICNTHR
      KWNTED = IVARHR(II)+1000
      DO 40 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 50
   40 CONTINUE
      GO TO 60
   50 OPTVAL(1,IPRDHR(II)) = BE(I)
```

```
      KWNTED = KWNTED - 1000
      OPTCST(1,IPRDHR(II)) = BE(I) * CC(1,KWNTED)
      COSTHR = COSTHR + OPTCST(1,IPRDHR(II))
   60 CONTINUE
C     ----------------------------------------------------------
C     COMPUTATION OF COST DUE TO FIRING OVER HORIZON
      COSTFR = 0.0
      DO 90 II = 1,ICNTFR
      KWNTED = IVARFR(II)+1000
      DO 70 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 80
   70 CONTINUE
      GO TO 90
   80 OPTVAL(2,IPRDFR(II)) = BE(I)
      KWNTED = KWNTED - 1000
      OPTCST(2,IPRDFR(II)) = BE(I) * CC(1,KWNTED)
      COSTFR = COSTFR + OPTCST(2,IPRDFR(II))
   90 CONTINUE
C     ----------------------------------------------------------
      CSTTT1 = COSTRG+COSTHR+COSTFR
      CSTTT2 = COSTHR+COSTFR
C     ----------------------------------------------------------
C     COMPUTATION OF COST DUE TO OVERTIME OVER HORIZON
      COSTOV = 0.0
      IF (ICNTOV.EQ.0) GO TO 140
      II = 0
  110 II = II + 1
      NPRD = IPRDOV(II)
      OPTVAL(4,NPRD) = 0.0
      OPTCST(4,NPRD) = 0.0
      JJ = 1
      IF (NOVBAR(NPRD).EQ.2) JJ = 2
      DO 130 J = 1,JJ
      KWNTED = IVAROV(II)+1000
      DO 120 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 125
  120 CONTINUE
      GO TO 128
  125 VAL1 = BE(I)
      KWNTED = KWNTED - 1000
      CST1 = BE(I) * CC(4,KWNTED)
      OPTVAL (4,NPRD) = OPTVAL (4,NPRD) + VAL1
      OPTCST (4,NPRD) = OPTCST (4,NPRD) + CST1
  128 IF (J.EQ.2) GO TO 130
      IF (JJ.EQ.2) II = II + 1
  130 CONTINUE
      COSTOV = COSTOV + OPTCST(4,NPRD)
      IF (II.LT.ICNTOV) GO TO 110
C     ----------------------------------------------------------
C     COMPUTATION OF NUMBER OF IDLE WORKERS OVER HORIZON
  140 WRKIDL = 0.0
      IF (ICNTIC.EQ.0) GO TO 165
      IJK = 0
      DO 160 II = 1,ICNTIC
      NPRD = IPRDID(II)
      IJK = IJK + 1
      JJ = 1
      IF (IDLE(NPRD).EQ.2) JJ = 2
```

```
      DO 155 J = 1,JJ
      KWNTED = IVARID(IJK) + 1000
      DO 145 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 150
  145 CONTINUE
      GO TO 153
  150 VAL1 = BE(I)
      KWNTED = KWNTED - 1000
      CST1 = VAL1 * CC(3,KWNTED)
      OPTVAL(3,NPRD) = OPTVAL(3,NPRD) + VAL1
      OPTCST(3,NPRD) = OPTCST(3,NPRD) + CST1
  153 IF (J.EQ.2) GO TO 155
      IF (JJ.EQ.2) IJK = IJK + 1
  155 CONTINUE
      WRKIDL = WRKIDL + OPTVAL(3,NPRD)
  160 CONTINUE
C     ----------------------------------------------------------
C     COMPUTATION OF INVENTORY COST OVER HORIZON
  165 COSTNV = 0.0
      DO 190 II = 1,BIGT
      KWNTED = IVARIN(II)+1000
      DO 170 I = 1,M
      IF (IB(I).EQ.KWNTED) GO TO 180
  170 CONTINUE
      GO TO 190
  180 OPTVAL(7,II) = BE(I)
      KWNTED = KWNTED - 1000
      OPTCST(7,II) = BE(I) * CC(2,KWNTED)
      COSTNV = COSTNV + OPTCST(7,II)
  190 CONTINUE
C     ----------------------------------------------------------
C     COMPUTATION OF TOTAL COST OVER HORIZON
      CSTTTL = CSTTT1 + COSTOV + COSTNV
      WRITE (6,610) CSTTT1,COSTNV,WRKIDL,COSTOV,CSTTTL
  610 FORMAT (//,10X,'COST OF REG. PAY, HIRING, AND FIRING :',F12.2,/,
     110X,'COST OF INVENTORY :',F12.2,/,10X,'TOTAL IDLE WORKER :',
     1F7.2,/,10X,'COST OF OVERTIME  :',F9.2,/,
     110X,'TOTAL PRODUCTION COST :',F12.2,//)
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE DUALS
C     THIS SUROUTINE FINDS THE VALUES OF THE DUAL VARIABLES
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT
      COMMON/BL10/A(110,175),C(175),CC(10,175),SB(110)
      COMMON/BL50/M,N,L
      COMMON/BL60/W(175),WM(5,175),Y(175),Y1(175),Y2(175)
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL390/ICNTFR,ICNTHR,ICNTIC,ICNTIV,ICNTOV,ICNTSB
      COMMON/BL395/IPRDHR(12),IPRDFR(12),IPRDOV(24),IPRDID(12)
      COMMON/BL380/ICNSFR(12),ICNSHR(12),ICNSID(12),ICNSOB(12),
     1             ICNSOV(24),ICNSWF(12)
      COMMON/BL385/IDOLFR(12),IDOLHR(12),IDOLID(12),IDOOBJ(12),
     1             IDOLOV(24),IDOLWF(12)
      COMMON/BL545/VALNEW,IOPTON,NEWOPT,NNNCNS
      COMMON/BL550/RHSHIR(2,12),RHSFIR(2,12),RHSOVR(2,24),RHSIDL(2,12),
```

```
     1              RHSWFC(2,12),RHSOBJ(2,5)
C
      NL = N + L
 1000 WRITE (6,610)
      WRITE (6,665)
      DO 30 I = 1,ICNTHR
      II = IDOLHR(I)
      DO 10 J = 1,NL
      IF (J.EQ.II) GO TO 20
   10 CONTINUE
   20 SHDPRC = -W(J)
      WRITE (6,670) IPRDHR(I),ICNSHR(I),RHSHIR(2,I),SHDPRC
   30 CONTINUE
C     ---------------------------------------------------------
 2000 WRITE (6,620)
      WRITE (6,665)
      DO 60 I = 1,ICNTFR
      II = IDOLFR(I)
      DO 40 J = 1,NL
      IF ( J.EQ.II) GO TO 50
   40 CONTINUE
   50 SHDPRC = - W(J)
      WRITE (6,670) IPRDFR(I),ICNSFR(I),RHSFIR(2,I),SHDPRC
   60 CONTINUE
C     -----------------------------------------------------------
 3000 IF (ICNTIC.EQ.0) GO TO 3500
      WRITE (6,630)
      WRITE (6,665)
      DO 90 I = 1,ICNTIC
      II = IDOLID(I)
      DO 70 J = 1,NL
      IF (J.EQ.II) GO TO 80
   70 CONTINUE
   80 SHDPRC = - W(J)
      WRITE (6,670) IPRDID(I),ICNSID(I),RHSIDL(2,I),SHDPRC
   90 CONTINUE
      GO TO 4000
C     ----------------------------------------------------------
 3500 WRITE (6,600)
  600 FORMAT (10X,'NO CONSTRAINT FOR IDLE TIME EXISTS',//)
C     ----------------------------------------------------------
 4000 WRITE (6,640)
      WRITE (6,665)
      DO 120 I = 1,ICNTOV
      II = IDOLOV(I)
      DO 100 J = 1,NL
      IF (J.EQ.II) GO TO 110
  100 CONTINUE
  110 SHDPRC = - W(J)
      WRITE (6,670) IPRDOV(I),ICNSOV(I),RHSOVR(2,I),SHDPRC
  120 CONTINUE
C     ----------------------------------------------------------
 5000 WRITE (6,650)
      WRITE (6,665)
      DO 150 I = 1,BIGT
      II = IDOLWF(I)
      DO 130 J =1,NL
      IF (J.EQ.II) GO TO 140
```

```
  130 CONTINUE
  140 SHDPRC = - W(J)
      WRITE (6,670) I,ICNSWF(I),RHSWFC(2,I),SHDPRC
  150 CONTINUE
C     ----------------------------------------------------------
 6000 WRITE (6,660)
      WRITE (6,667)
      DO 180 I = 1,5
      II = IDOOBJ(I)
      DO 160 J = 1,NL
      IF (J.EQ.II) GO TO 170
  160 CONTINUE
  170 SHDPRC = - W(J)
      WRITE (6,670) I,ICNSOB(I),RHSOBJ(1,I),SHDPRC
  180 CONTINUE
C     ------------------------------------------------------------
C     WRITE (6,680)
C     WRITE (6,690) (WM(1,I),I=1,M)
  610 FORMAT (///,10X,'SHADOW PRICE FOR HIRING OPTION :',//)
  620 FORMAT (///,10X,'SHADOW PRICE FOR FIRING OPTION :',//)
  630 FORMAT (///,10X,'SHADOW PRICE FOR IDLE TIME OPTION :',//)
  640 FORMAT (///,10X,'SHADOW PRICE FOR OVERTIME OPTION :',//)
  650 FORMAT (///,10X,'SHADOW PRICE FOR UPPER LIMT OF WORKFORCE :',//)
  660 FORMAT (///,10X,'SHADOW PRICE FOR OBJ. FUNC. VALUE :',//)
  665 FORMAT (15X,'PERIOD',5X,'CNSTRN NO',11X,'RHSVAL',9X,'SHDPRC',//)
  667 FORMAT (13X,'FUNC NO',3X,'CNSTRN NO',13X,'RHSVAL',9X,'SHDPRC',//)
  670 FORMAT (13X,I4,9X,I4,10X,F12.2,5X,F9.2,/)
C 680 FORMAT (/,10X,'CB B INV : ',//)
C 690 FORMAT (15X,8F10.2,/)
      RETURN
      END
C
C     *****************************************************************
      SUBROUTINE RESULT
C     THIS SUBROUTINE PRINTS THE DETAILED FINAL RESULTS
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER BIGT
      COMMON/BL300/ISBRCG,NGRBGN,BIGT
      COMMON/BL600/OPTVAL(8,12),OPTCST(8,12)
C
      WRITE (6,600)
  600 FORMAT (//,10X,'DETAILED RESULTS :',//)
      WRITE (6,602)
  602 FORMAT (10X,'PERIOD    REGULAR ',6X,' HIRING  ',4X,
     1'  FIRING  ',2X,' OVERTIME ',1X,' INVENTORY',/,
     116X,'    PAYROLL ',7X,' COST ',8X,' COST ',5X,'  COST  ',
     15X,' COST ',//)
      DO 10 I = 1,BIGT
      WRITE (6,604) I,OPTCST(6,I),OPTCST(1,I),OPTCST(2,I),
     1                          OPTCST(4,I),OPTCST(7,I)
  604 FORMAT (12X,I3,3X,F9.1,5X,F8.1,6X,F8.1,4X,F8.1,4X,F8.1,/)
   10 CONTINUE
      WRITE (6,605)
  605 FORMAT (//,10X,'PERIOD     REGULAR ',6X,'  HIRING  ',3X,
     1'  FIRING  ',2X,' OVERTIME ',3X,' INVENTORY',/,
     120X,' WRKFORC ',7X,'QUANTITY',5X,'QUANTITY',5X,'PRODCTN',//)
      DO 20 I = 1,BIGT
      WRITE (6,606) I,OPTVAL(6,I),OPTVAL(1,I),OPTVAL(2,I),
```

```
  1                             OPTVAL(4,I),OPTVAL(7,I)
606 FORMAT (12X,I3,3X,F9.1,6X,F8.1,5X,F8.1,6X,F8.1,5X,F8.1,/)
 20 CONTINUE
    RETURN
    END
```

APPENDIX D

OUTPUT OF A COMPUTER RUN

WORKING MODEL (CONSTANT PRODUCTIVITY) :

NUMBER OF PERIODS IN THE HORIZON = 10

NO OF WORKER CLASSES IN THE BEGINNING = 2

WORKER IN THE CLASS 1 = 10.00

WORKER IN THE CLASS 2 = 71.00

OVERTIME DUR/RGLR. TIME DUR = 0.50

THE EFFICIENCY DURING OVERTIME = 1.00

OVERTIME PAY/REGULAR PAY = 1.50

SUBCONTRACTING IN THE LAST PERIOD = 0.00

NO. OF CLASSES WORKING OVERTIME LAST PERIOD = 0

NO. OF WORKERS (IN LOWEST CLASS THROUGH WHICH
OVERTIME EXTENDED IN THE LAST PERIOD) NOT UTILIZED = 0.00

DEMAND IN THE LAST PERIOD = 400.00

INITIAL INVENTORY = 263.00

HIRING COST PER WORKER = 180.00

FIRING COST PER WORKER = 360.00

REGULAR PAYROLL PER WORKER OF CLASS 1 = 340.00

REGULAR PAYROLL PER WORKER OF CLASS 2 = 340.00

UPPER LIMIT OF REGULAR WORKFORCE = 95.00

UNIT INVENTORY CARRYING COST = 20.00

UNIT SUBCONTRACTING COST = 200.00

DEMAND IN PERIOD 1 = 430.00

DEMAND IN PERIOD 2 = 447.00

DEMAND IN PERIOD 3 = 440.00

DEMAND IN PERIOD 4 = 316.00

DEMAND IN PERIOD 5 = 397.00

DEMAND IN PERIOD 6 = 375.00

DEMAND IN PERIOD 7 = 292.00

DEMAND IN PERIOD 8 = 458.00

DEMAND IN PERIOD 9 = 400.00

DEMAND IN PERIOD 10 = 350.00

NO OF ALLOWABLE IDLE WORKER AT ANY PERIOD = 20.00

REGULAR TIME PRODUCTION RATE OF CLASS 1 = 5.67

ESTIMATED OVERTIME PRODUCTIVITY OF THIS CLASS = 5.67

REGULAR TIME PRODUCTION RATE OF CLASS 2 = 5.67

ESTIMATED OVERTIME PRODUCTIVITY OF THIS CLASS = 5.67

OPTIONS SET AT THE BEGINING

PERIOD	NREG(T)	NOVR(T)	SUBCON(T)	HIRFIR	IDLE(T)
1	2(51.55)	0(0.00)	0.00	1	1
2	2(2.16)	0(0.00)	0.00	2	0
3	2(3.40)	0(0.00)	0.00	1	1
4	2(25.27)	0(0.00)	0.00	1	1
5	2(10.98)	0(0.00)	0.00	2	0
6	2(14.86)	0(0.00)	0.00	1	1
7	2(29.50)	0(0.00)	0.00	1	1
8	2(0.22)	0(0.00)	0.00	2	0
9	2(10.45)	0(0.00)	0.00	1	1
10	2(19.27)	0(0.00)	0.00	1	1

NO OVERTIME IF ALL WORKERS ARE KEPT.
THIS MIGHT CAUSE IDLE TIME TO EXIST.

SINCE OVERTIME IS A FEASIBLE OPTION
THIS IS INCLUDED IN THE MODEL WHEN HIRING
MIGHT TAKE PLACE. THE NEW OVERTIME SCHE-
DULE IS AS FOLLOWS.

0 1 0 0 1 0

0 1 0 0

TOTAL HIRING COULD BE = 92.95
TOTAL FIRING COULD BE = 112.22

TOTAL NUMBER OF CONSTRAINTS : 71
TOTAL NUMBER OF VARIABLES : 60

NHFIR1(2T-1) AND NHFIR1(2T)

1 1 2 0 1 1

1 1 2 0 1 1

1 1 2 0 1 1

1 1

IDLE(T)

1 0 1 1 0 1

1 0 1 1

NOVBAR(T)

0 1 0 0 1 0

0 1 0 0

SBCNBR(T)

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00

CONSTRAINTS NOS. RELATED TO HIRING :

13 34 55

VARIABLE NOS. RELATED TO HIRING :

11 29 47

CONSTRAINT NOS. RELATED TO FIRING :

6 20 27 41
48 62 69

VARIABLE NOS. RELATED TO FIRING :

5 17 23 35
41 53 59

CONSTRAINT NOS. RELATED TO OVERTIME :

14 35 56

VARIABLE NOS. RELATED TO OVERTIME :

12 30 48

COSTRAINT NOS. RELATED TO IDLE TIME :

7 21 28 42
49 63 70

VARIABLES RELATED TO IDLE TIME :

6 18 24 36
42 54 60

TOTAL NUMBER OF CONSTRAINTS : 76
TOTAL NUMBER OF VARIABLES : 95

*** PHASE ONE ***

*** FEASIBLE ***

*** PHASE 2 ***

*** OPTIMAL SOLUTION ***

*** BASIC VARIABLES ***

1003	1001	1002	1088	1021	1064	1071
1081	1007	1061	1009	1004	1092	1078
1008	1047	1014	1082	1010	1065	1074
1083	1072	1020	1026	1029	1066	1073
1084	1058	1062	1027	1022	1085	1079
1015	1031	1032	1060	1013	1067	1019
1086	1037	1038	1044	1025	1068	1075
1087	1046	1063	1045	1043	1039	1048
1033	1005	1050	1051	1049	1069	1076
1089	1055	1056	1057	1040	1070	1077
1090	1091	1028	1093	1094	1095	

*** VALUES OF THE BASIC VARIABLES ***

51.964	351.333	10.000	28.570	51.964	93.180	20.000
33.036	351.333	92.945	51.964	184.333	91706.500	28.350
10.000	1.468	10.000	33.036	88.667	112.217	20.000
33.036	20.000	10.000	10.000	2.998	112.217	20.000
33.036	0.000	89.947	54.962	35.333	30.038	28.350
51.964	368.333	10.000	4.702	351.333	112.217	351.333
30.038	368.333	10.000	10.000	368.333	112.217	20.000
30.038	23.342	91.477	56.430	376.658	54.962	28.350
54.962	19.036	10.000	56.430	376.658	112.217	20.000
28.570	350.000	10.000	56.430	76.333	112.217	15.298
28.570	774052.819	6.667	25.298	97450.000	5763209.319	

OBJECTIVE FUNCTION VALUE = 236790.68

COST OF REG. PAY, HIRING, AND FIRING : 225947.18
COST OF INVENTORY : 8293.50
TOTAL IDLE WORKER : 4.70
COST OF OVERTIME : 2550.00
TOTAL PRODUCTION COST : 236790.68

DETAILED RESULTS :

PERIOD	REGULAR PAYROLL	HIRING COST	FIRING COST	OVERTIME COST	INVENTORY COST
1	21067.6	0.0	6853.1	0.0	3686.7
2	21067.6	0.0	0.0	0.0	1773.3
3	21067.6	0.0	0.0	0.0	0.0

4	21067.6	0.0	0.0	0.0	706.7
5	22087.0	539.7	0.0	0.0	133.3
6	22087.0	0.0	0.0	0.0	0.0
7	22087.0	0.0	0.0	0.0	1526.7
8	22586.2	264.3	0.0	2550.0	466.8
9	22586.2	0.0	0.0	0.0	0.0
10	22586.2	0.0	0.0	0.0	0.0

PERIOD	REGULAR WRKFORC	HIRING QUANTITY	FIRING QUANTITY	OVERTIME PRODCTN	INVENTORY
1	62.0	0.0	19.0	0.0	184.3
2	62.0	0.0	0.0	0.0	88.7
3	62.0	0.0	0.0	0.0	0.0
4	62.0	0.0	0.0	0.0	35.3
5	65.0	3.0	0.0	0.0	6.7
6	65.0	0.0	0.0	0.0	0.0
7	65.0	0.0	0.0	0.0	76.3
8	66.4	1.5	0.0	28.4	23.3
9	66.4	0.0	0.0	0.0	0.0
10	66.4	0.0	0.0	0.0	0.0

SHADOW PRICE FOR HIRING OPTION :

PERIOD	CNSTRN NO	RHSVAL	SHDPRC
2	13	92.95	-0.00
5	34	92.95	-0.00
8	55	92.95	-0.00

SHADOW PRICE FOR FIRING OPTION :

PERIOD	CNSTRN NO	RHSVAL	SHDPRC
1	6	112.22	-0.00

3	20	112.22	-0.00
4	27	112.22	-0.00
6	41	112.22	-0.00
7	48	112.22	-0.00
9	62	112.22	-0.00
10	69	112.22	-0.00

SHADOW PRICE FOR IDLE TIME OPTION :

PERIOD	CNSTRN NO	RHSVAL	SHDPRC
1	7	20.00	-0.00
3	21	20.00	-0.00
4	28	20.00	-0.00
6	42	20.00	-0.00
7	49	20.00	-0.00
9	63	20.00	-0.00
10	70	20.00	-0.00

SHADOW PRICE FOR OVERTIME OPTION :

PERIOD	CNSTRN NO	RHSVAL	SHDPRC
2	14	0.00	-0.00
5	35	0.00	-0.00
8	56	0.00	5.87

SHADOW PRICE FOR UPPER LIMT OF WORKFORCE :

PERIOD	CNSTRN NO	RHSVAL	SHDPRC
1	8	95.00	-0.00
2	15	95.00	0.00
3	22	95.00	-0.00
4	29	95.00	-0.00

5	36	95.00	0.00
6	43	95.00	-0.00
7	50	95.00	-0.00
8	57	95.00	0.00
9	64	95.00	-0.00
10	71	95.00	-0.00

SHADOW PRICE FOR OBJ. FUNC. VALUE :

FUNC NO	CNSTRN NO	RHSVAL	SHDPRC
1	72	1000000.00	-0.00
2	73	100000.00	-0.00
3	74	30.00	-0.00
4	75	100000.00	-0.00
5	76	6000000.00	-0.00

VITA

Muhammad Abdulhalim Shaikh

Candidate for the Degree of

Doctor of Philosophy

Thesis: INCORPORATING DECISION MAKING INTO A MULTICRITERIA AGGREGATE PRODUCTION PLANNING MODEL.

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Ullapara, Bangladesh, March 1, 1948, the second son of M. A. Jabbar Shaikh, and M. Nurun Nahar. Married to M. Sabequn Nahar on July 11, 1971.

Education: Graduated from Pabna Edward College, Bangladesh in May, 1964; received Bachelor of Science in Mechanical Engineering degree from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in November, 1968; received Master of Engineering (Systems Engineering) degree from Hiroshima University, Hiroshima, Japan, in March, 1979; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1984.

Professional Experience: Lecturer, Mechanical Engineering Department, Rajshahi Engineering College, Rajshahi, Bangladesh, 1969-1972; Assistant Professor (Rajshahi Engineering college), 1972-1975; Graduate Teaching and Research Assistant, Oklahoma State University, 1981-1983; Visiting Assistant Professor, teaching Nonlinear Programming, and Discrete and Continuous Systems Simulation (SLAM), Industrial Quality Control, Engineering Economic Analysis, Clemson University, Clemson, South Carolina, August, 1983, to present.

Professional Organization: Institue of Industrial Engineers.