

To the Graduate Council:

I am submitting herewith a dissertation written by Manohar S. Madan entitled "A Heuristic and an Exact Solution Algorithm for a Class of Production Planning and Scheduling Problems." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

Kenneth C. Gilbert
Kenneth C. Gilbert, Major Professor

We have read this dissertation
and recommend its acceptance:

Dean S. Fink

Ronald W. Boling
James H. ...

Accepted for the Council:

Cowminkel
Vice Provost
and Dean of The Graduate School

A HEURISTIC AND AN EXACT SOLUTION ALGORITHM FOR
A CLASS OF PRODUCTION PLANNING AND
SCHEDULING PROBLEMS

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Manohar S. Madan

March 1988

To the Great Music of
Jim Morrison and THE DOORS

ACKNOWLEDGMENTS

I would like to express my deep gratitude and appreciation to Dr. Kenneth C. Gilbert, my dissertation director, for his expert help, encouragement, and patience in the completion of this dissertation. A special thanks to Dr. Kenneth C. Gilbert for helping me learn a lot about scientific writing. I would also like to thank sincerely each of the rest of my doctoral committee for his continued interest and help: Dr. Ronald W. Boling, Dr. Oscar F. Fowler, and Dr. James K. Ho. I would especially like to thank them for giving me quick feedback on my dissertation.

Very special thanks to Dr Robert S. Garfinkel for receiving his continuous encouragement and invaluable lessons throughout my graduate program. I acknowledge with pleasure the help received from Don Broach of the University of Tennessee Computing Center.

Other people have helped me in particular ways. Richard K. Laumann did an excellent job of typing and helping to prepare the dissertation manuscript to be published in final form, and I thank him very much for that.

Finally I would like to thank all the funny people-- Marx Brothers, Monty Pythons, Woody Allen etc. for helping me maintain my sense of humor throughout my graduate program.

ABSTRACT

In this dissertation we formulate and solve a class of production planning and scheduling problem. The problem is one of optimally specifying production levels for each product for each period of the planning horizon. The objective is to minimize sum of setup, regular time production, overtime production and inventory holding costs. The problem has been shown to be NP-hard in computational complexity.

We have formulated the production planning and scheduling problem as a variation of fixed charge transportation problem. We have developed an efficient heuristic and an exact solution algorithm based on this formulation.

The performance of the heuristic has been tested on 243 test problems. The problems differ with respect to (a) number of products, (b) number of periods, (c) seasonality of demand, (d) setup cost, and (e) tightness of capacity. The performance of the heuristic has been evaluated by comparing the heuristic solution value with a lower bound value.

The most significant effect on the difference between the heuristic solution and the lower bound is due to seasonality. The second most significant effect on the difference between the heuristic solution and the lower bound is due to setup cost. The effect of number of products, number of periods, and tightness of capacity on the difference between the heuristic solution and the lower bound seems to be insignificant.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION AND OVERVIEW	1
Introduction	1
Overview	4
II. DISCUSSION OF PROBLEM	7
Problem Description and Formulation	7
Tableau Formation of Problem	18
Computational Complexity of the Problem	20
III. LITERATURE SURVEY	22
Capacitated Product Scheduling Problem	22
Fixed Charge Problem	29
IV. HEURISTIC	35
Phase I	35
Phase II	37
Phase III	43
V. EXACT SOLUTION ALGORITHM	51
Description of Algorithm	51
Bounding Procedures	58
VI. LOWER BOUND	65
Procedure for Computing Lower Bound	65
Tightness of Lower Bound	68
Comparison of Lower Bound to Exact Solutions	73
VII. COMPUTATIONAL STUDY	77
Design of Computational Experiments	77
Summary of Results	83
VIII. TOPICS FOR FURTHER RESEARCH	96
BIBLIOGRAPHY	97
VITA	104

LIST OF FIGURES

FIGURE	PAGE
1. Production Capacity Cost Curve	9
2. Tableau Associated with Fixed Charge Transportation Problem	19
3. Nature of Objective Function and Linear Approximation of Objective Function	40
4. Multiplicative Seasonality Factor	80
5. Ranges of Demands and Setup Costs for Products	81
6. The Effect of Seasonality on Average Percent Difference (A.P.D.)	89
7. The Effect of Setup Costs on Average Percent Difference (A.P.D.)	90
8. The Effect of Tightness of Capacity on Average Percent Difference (A.P.D.)	91
9. The Effect of Number of Products on Average Percent Difference (A.P.D.)	92
10. The Effect of Number of Periods on Average Percent Difference (A.P.D.)	93
11. The Effect of Interaction of Setup Costs and Seasonality on Average Percent Difference (A.P.D.)	94

CHAPTER I

INTRODUCTION AND OVERVIEW

1.1 INTRODUCTION

The object of this dissertation is the development of a computationally efficient heuristic for a class of production planning and scheduling problems. The problem is one of a general class of problems commonly referred to as aggregate planning and master scheduling problems.

In the general aggregate planning and master scheduling problem a single production facility is used to produce several different products. The fundamental problem is one of optimally specifying the production levels for each product for each period of the planning horizon. The schedule must meet the forecasted demand for each product in each period. The total production for each period must not exceed the aggregate production capacity for that period. The objective is to minimize the sum of the setup, overtime, inventory holding, backorder, hiring and firing costs.

The capacitated production planning and scheduling problem is of interest for two reasons. First, it is found in several manufacturing settings. Plants with shallow bill of materials are good examples, such as assembly processes, stamping processes, bottling operations, or breweries. Second, a usual approach to constrained MRP (material requirements planning)

problem is to first develop a lot sizing plan called a "master schedule" for the "end products". The "master schedule" is essentially a capacitated product scheduling problem.

As a hypothetical example of such a problem consider that of scheduling production at a sheet metal pressing plant that produces hoods, fenders and body panels for automobiles. Based on the planned production of automobiles a forecasted weekly demand is available for three months into the future for each item produced. The plant operates for two shifts and can press 10,000 parts per day using regular time labor. As many as 5,000 additional parts can be produced in the third shift through the use of overtime. The master scheduler for this plant wishes to meet the forecasted demand with the available capacity at as low a cost as possible. The cost consists of setup costs, inventory holding costs, hiring and firing costs, and labor costs (regular time and overtime costs).

Each production run of a given part requires a very expensive setup of a press. In order to reduce the number of setups, parts can be held in inventory to meet future demand. There is however, a significant inventory holding cost associated with not only the costs of capital and storage but also with the increased difficulty of detecting, tracking and correcting quality problems in the presence of large inventories. The production in any period can be increased by using expensive overtime.

The capacitated product scheduling problem is one of the

most renowned issues in production planning. When extended to multiple products the problem becomes particularly formidable. The available solution techniques for the capacitated multi-product scheduling problem are less than satisfactory. The problem can be formulated as a mixed integer-separable programming problem. The approach is mathematically exact since it models the individual products and takes into account all major costs including setup costs. However, except for problems with very small number of products and planning periods the solution of the resulting mathematical programming problem is impractical (see Barany, Van Roy and Wolsey (1984)). Heuristic approaches have been developed for some special cases.

In practice, an aggregate approach is typically used: the demands for the individual products are aggregated into a common unit e.g. labor hours demanded are lumped into a single aggregate demand for each period transforming the problem a single-product problem. The setup costs for the individual products are ignored. Then a production schedule that specifies the production for each period in aggregate units is developed (typically through trial and error methods). This aggregate schedule is then heuristically "disaggregated" into a schedule specifying production levels for the individual items. Obviously, this inexact process does not yield an optimal master schedule.

1.2 OVERVIEW

The problem addressed here differs from the general aggregate planning and master scheduling problem in one important aspect: it considers the aggregate production capacity for each period as an input, rather than a variable to be optimized. With this one exception, this approach is capable of taking into account the major decision variables of intermediate range production planning e.g. setups, monthly or weekly production levels, inventory levels and shortage levels for each product, overtime, subcontracting and other means for temporarily adjusting capacity. The model is not applicable, for example, in situations where the optimization of the hiring and firing strategy is an important part of the production planning process.

Some authors (e.g. Graves (1982)) have referred to the problem, similar to the one addressed here, as the production planning and scheduling problem, while other authors (e.g. Bahl Ritzman and Gupta (1986)) have called it the single level constrained resource problem. In the rest of this dissertation we will refer to the capacitated product scheduling problem as the single level constrained resource (SLCR) problem.

A basis for this research is our observation that the SLCR problem can be formulated as a type of fixed charge separable transportation problem. A heuristic solution method based on this formulation has been developed. This heuristic employs

some ideas used in solving fixed charged transportation problems arising in other areas of applications e.g. location-allocation problems. It also exploits the special cost structure arising in the SLCR problem.

For purposes of comparison we have also developed an exact implicit enumeration algorithm, also based on the fixed charge separable transportation problem formulation. The development of this exact algorithm in itself posed an interesting research problem. However, since it seemed unlikely that an exact solution method would be practical for real world sized problems, the primary objective in this research was the development of a heuristic that would produce good solutions to large problems.

The computational results for the heuristic are very encouraging. It is fast enough to be considered practical for large problems e.g. problems with nine products and thirty six periods could be solved in an average of approximately 2.00 minutes of c.p.u time on an IBM 4381 computer. The solutions for the heuristic were either optimal or within .25% of optimal solution for all of the test problems that could be solved exactly using the implicit enumeration algorithm. Furthermore for many of the problems which could not be solved via implicit enumeration, it was possible to demonstrate via bounds that the heuristic solution was optimal or near optimal.

On the negative side it was not possible to obtain tight bounds for those test problems having very high setup costs or

highly seasonal demand. Thus for these types of problems it was not possible to determine if the heuristic solutions were near optimal. However even in the worst case it would appear that the heuristic developed in this dissertation would be an improvement over the aggregate approach that is commonly being used.

Chapter II gives a detailed problem description, a mathematical formulation of the SLCR problem and a proof showing the problem to be NP-hard. Chapter III consists of literature review. The chapter discusses literature in the field of capacitated product scheduling as well as the field of the fixed charge problems. Chapter IV describes the heuristic developed for solving the problem. Chapter V describes the exact solution algorithm developed.

Chapter VI describes the procedure for computing a lower bound which is used as bases of comparison of the heuristic solutions. Chapter VII discusses the computational experiments. The chapter consists of the design of computational experiments as well as the summary of results. The chapter discusses in detail the 243 problems used in testing the heuristic. The problems were generated using all combinations of three different levels of five different factors-- number of products, number of time periods, cost of setups, amount of regular time capacity available and seasonality of demand. Chapter VIII discusses topics for further research.

CHAPTER II

DISCUSSION OF PROBLEM

The SLCR problem has traditionally been formulated as a mixed integer linear programming problem. The results developed in this research are based on our observation that, in the particular case we solve, the problem can be formulated as a fixed charge transportation problem. For completeness we first formulate the problem that we solve using the traditional mixed integer linear programming approach. We then show how the problem can be formulated as a type of fixed charge separable transportation problem.

2.1 PROBLEM DESCRIPTION AND FORMULATION

In our problem description and formulation we follow fairly standard conventions (see Lambrecht and Vanderveken (1979) or Karni and Roll (1982)). It is assumed that the planning horizon is divided into M equal time periods (typically weeks). There are J different products. The demand D_{jm} , $j=1, \dots, J$, $m=1, \dots, M$, for product j in period m is assumed to be known.

The products can be produced using k different types of capacity e.g. regular time, and overtime. C_{km} , $k=1, \dots, K$, $m=1, \dots, M$, denotes the number of units of capacity (typically machine hours or labor hours) type k available in period m .

It is assumed that the number of units of capacity used in the production of a given product is directly proportional to the quantity produced. Thus for purposes of exposition we will express all quantities in units of capacity required. For product j in period m , $j=1,\dots,J$, $m=1,\dots,M$, the demand D_{jm} , the production level X_{jm} , and the end of period inventory level I_{jm} , will be expressed in units of capacity, i.e. one unit of a product is the amount that can be produced using one unit of capacity.

The cost of capacity type k , $k=1,\dots,K$, is P_k dollars per unit, $P_1 \leq P_2 \leq \dots \leq P_K$. Thus during a given period m the capacity cost is a piecewise linear function of $\sum_{j=1}^J X_{jm}$, the total production level during period m as shown in Figure 1 (see Hanssman and Hess (1960) and Haehling Von Lanzenhauer (1970)).

Products produced during a given period can be used to meet the demand for the product during that period or in subsequent periods. These products cannot be used to meet the demands of earlier periods i.e. backordering is not permitted. Each product j , $j=1,\dots,J$, has a holding cost of H_j dollars per holding cost is based on the end of period inventory levels. Thus the holding cost for product j in period m , $j=1,\dots,J$,

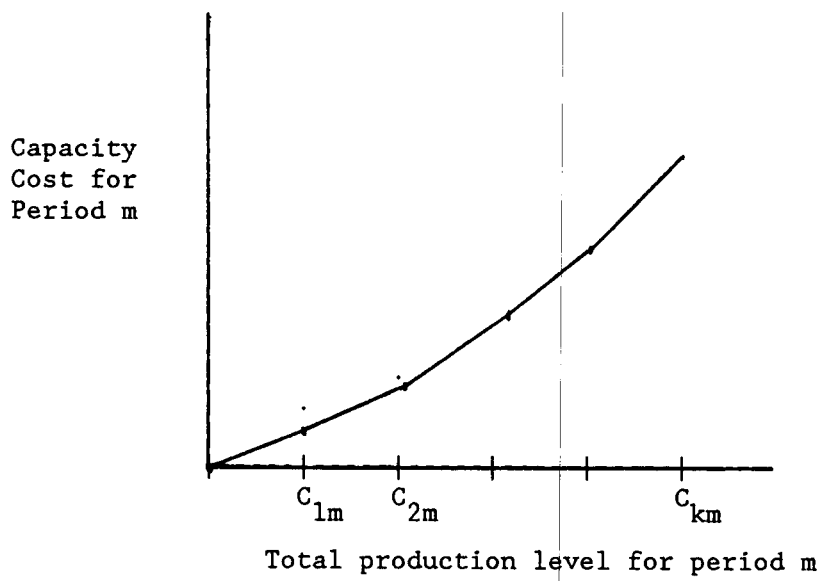


Figure 1. Production Capacity Cost Curve.

$m=1, \dots, M$, is $H_j * I_{jm}$. (The costs of holding inventory can be unit per time period. The computation of the total inventory regarded as being linear over a fairly broad range of inventory levels, with possible discontinuities at very high inventory levels (see Buffa and Miller (1979))).

Each product requires a setup for each period in which it is produced. The cost of a setup for product j is S_j dollars per setup, $j=1, \dots, J$. It is assumed that downtime consumed by the setup operation is negligible.

2.1.1 Mixed Integer Linear Programming Formulation

Let j , k and m index product, type of production capacity and period respectively, $j=1, \dots, J$, $k=1, \dots, K$, $m=1, \dots, M$. Using the following notations the SLCR is formulated as a mixed integer linear programming problem.

Let,

J = number of products;

K = type of production capacity available;

M = number of time periods in the planning horizon;

D_{jm} = forecasted demand (expressed in units of capacity) of product j in period m , $j=1, \dots, J$, $m=1, \dots, M$;

X_{jm} = production level (expressed in units of capacity) of product j in period m , $j=1, \dots, J$, $m=1, \dots, M$;

I_{jm} = the ending inventory (expressed in units of capacity) of product j in period m , $j=1, \dots, J$, $m=1, \dots, M$;

- I_{j0} - initial inventory level (expressed in units of capacity) of product $j=1, \dots, J$;
- S_j - setup cost for product j $j=1, \dots, J$;
- H_j - holding cost in dollars per unit of product j per period $j=1, \dots, J$;
- P_k - cost per unit of capacity type k , $k=1, \dots, K$;
- C_{km} - units of capacity type k available in period m , $k=1, \dots, K$, $m=1, \dots, M$;
- T_{km} - the total amount of capacity type k used in period m , $k=1, \dots, K$, $m=1, \dots, M$;
- Y_{jm} - is a binary variable where $Y_{jm}=1$ and $Y_{jm}=0$ signify respectively presence and absence of setup for product j in period m , $j=1, \dots, J$, $m=1, \dots, M$.

The mixed integer linear programming formulation of the SCLR problem is given below.

Minimize

$$\sum_{j=1}^J \sum_{m=1}^M [S_j * Y_{jm}] + \sum_{j=1}^J \sum_{m=1}^M [H_j * I_{jm}] + \sum_{k=1}^K \sum_{m=1}^M [P_k * T_{km}] \quad (1.1)$$

subject to:

$$I_{j, (m-1)} + X_{jm} - I_{jm} = D_{jm} \quad \begin{matrix} j=1, \dots, J \\ m=1, \dots, M \end{matrix} \quad (1.2)$$

$$\sum_{j=1}^J X_{jm} - \sum_{k=1}^K T_{km} = 0 \quad m=1, \dots, M \quad (1.3)$$

$$T_{km} \leq C_{km} \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, M \end{matrix} \quad (1.4)$$

$$\begin{aligned} Y_{jm} &= 1 \text{ if } X_{jm} > 0 \\ &= 0 \text{ otherwise} \end{aligned} \quad \begin{matrix} j=1, \dots, J \\ m=1, \dots, M \end{matrix} \quad (1.5)$$

$$I_{jm} \geq 0, X_{jm} \geq 0 \quad \begin{matrix} j=1, \dots, J \\ m=1, \dots, M \end{matrix} \quad (1.6)$$

The objective function (1.1) represents the minimization of total cost. The first term in the objective function represents the sum of setup costs, the second term represents total production capacity cost, and the third term represents the total inventory holding cost.

Constraints (1.2) represent the production-inventory balance equation in which both the quantity of inventory I_{jm} of

product j to be left in stock at the end of period m and the demand D_{jm} for product j in period m are supplied by: the amount $I_{j,(m-1)}$ of product j in stock at the end of period $(m-1)$, and X_{jm} the production quantity of product j in period m .

Constraints (1.3) require that $\sum_{j=1}^J \sum_{m=1}^M X_{jm}$ the sum of all products produced in period m , $m=1, \dots, M$, is equal to $\sum_{k=1}^K T_{km}$ the sum of all types of production capacity used in period m , $m=1, \dots, M$.

Constraints (1.4) require that T_{km} the total amount of capacity of type k , $k=1, \dots, K$, used in period m , $m=1, \dots, M$, does not exceed C_{km} the available capacity type k , $k=1, \dots, K$, in period m , $m=1, \dots, M$.

Constraints (1.5) are logical constraints which account for setups for different products in every period. Setup cost S_j will be incurred once for product j in period m if X_{jm} the production amount of product j , $j=1, \dots, J$, in period m , $m=1, \dots, M$, is positive. Constraints (1.6) are nonnegativity constraints.

Some desirable features of the mixed integer linear programming formulation are its flexibility and its comprehensiveness. It is possible to take into consideration the costs of changing capacity, e.g. hiring and firing costs, by treating the capacity levels as variables rather than constraints (Newson (1975)). Also more than one capacity, e.g.

different machine capacities, can be considered in the formulation by including them as capacity constraints (Billington, McClain and Thomas (1983)).

The obvious disadvantage of this formulation is that it produces a mixed integer linear program with $O(MN)$ integer variables and $O(MN)$ continuous variables. Thus the formulation does not provide a means of solution for most real world problems (Bahl, Ritzman and Gupta (1986)). The largest problem known to be solved exactly using this type of formulation is a special case of eight product eight period problem (see Barany, Van Roy and Wolsey (1984)).

2.1.2 Fixed Charge Transportation Problem Formulation

The SLCR problem can be formulated as a type of fixed charge transportation problem. We have exploited this structure in the development of our solution approach. In addition to the notations used in the previous section we introduce the following.

Let j index product, k index type of production capacity, m index period in which product is produced and n index period in which product is demanded, $j=1, \dots, J$, $k=1, \dots, K$, $m=1, \dots, M$, $n=1, \dots, M$.

We follow the notation of the previous section. We introduce the following variable.

Let X_{jkmn} be the quantity (expressed in units of capacity)

of product j produced using capacity type k in period m used to meet the demand in period n , where, $j=1, \dots, J$, $k=1, \dots, K$, $m=1, \dots, M$, and $n=1, \dots, M$.

The fixed charge transportation formulation of the single level constrained resource problem is given below.

Minimize

$$\begin{aligned} \sum_{j=1}^J \sum_{m=1}^M [S_j * Y_{jm}] + \sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M \sum_{n=m}^M [X_{jkmn} * P_k] \\ + \sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M \sum_{n=m}^M [n-m] * [X_{jkmn} * H_j] \end{aligned} \quad (2.1)$$

subject to:

$$\sum_{k=1}^K \sum_{m=1}^M X_{jkmn} = D_{jn} \quad \begin{array}{l} j=1, \dots, J \\ n=1, \dots, M \end{array} \quad (2.2)$$

$$\sum_{j=1}^J \sum_{n=m}^M X_{jkmn} \leq C_{km} \quad \begin{array}{l} k=1, \dots, K \\ m=1, \dots, M \end{array} \quad (2.3)$$

$$\begin{aligned} Y_{jm} = 1 \text{ if } \sum_{k=1}^K \sum_{n=m}^M X_{jkmn} > 0 \\ = 0 \text{ otherwise} \end{aligned} \quad \begin{array}{l} k=1, \dots, K \\ m=1, \dots, M \end{array} \quad (2.4)$$

$$X_{jkmn} \geq 0 \quad \begin{array}{l} j=1, \dots, J \\ k=1, \dots, K \\ m=1, \dots, M \\ n=1, \dots, M \end{array} \quad (2.5)$$

Problem (2.1)-(2.5) is a variation of fixed charge

transportation problem. It differs from the traditional fixed charge transportation problem in sense that a group of cells in the transportation tableau share a unit of fixed charge. The objective function (2.1) expresses the minimization of total cost. The first term in the objective function represents the sum of setup costs for all products in every period, the second term represents total production capacity cost, and the third term represents total inventory holding cost.

Constraints (2.2) are demand constraints which require that demand, D_{jn} , of product j , $j=1, \dots, J$, during period n , $n=1, \dots, M$, is met by X_{jkmn} , $m=n$, the production quantity of product j in period m used to meet the demand in period n , and by $\sum_{k=1}^K \sum_{m=1}^{n-1} X_{jkmn}$, the quantity of product j produced in periods prior to period n and used to meet the demand in period n (i.e. inventory from previous periods).

Constraints (2.3) are capacity constraints which require J that $\sum_{j=1}^J \sum_{n=m}^M X_{jkmn}$, the total amount of production capacity of type k used in period m to produce different products does not exceed the total amount of capacity type k available in period m , $m=1, \dots, M$, $k=1, \dots, K$.

It can be assumed that the total demand and total supply are equal. If the total demand is less than the supply then an additional column with demand equalling excess supply is added. If the total demand is greater than the supply the problem is infeasible.

If total demand for products over the planning periods is less than the total available capacity, we define $D_{(J+1),(M+1)}$ as follows:

$$D_{(J+1),(M+1)} = \sum_{k=1}^K \sum_{m=1}^M C_{km} - \sum_{j=1}^J \sum_{n=1}^M D_{jn}$$

where $D_{(J+1),(K+1)}$ is demand for dummy column (destination) $((J+1),(M+1))$. We then add the following demand constraint for dummy demand.

$$\sum_{k=1}^K \sum_{m=1}^M X_{(J+1),k,m,(M+1)} = D_{(J+1),(M+1)}$$

Constraints (2.3) will be changed to equality constraints as shown below.

$$\sum_{k=1}^K \sum_{n=m}^M X_{jkmn} + X_{(J+1),k,m,(M+1)} = C_{km} \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, M \end{matrix}$$

Constraints (2.4) are logical constraints which account for setups for product j in period m . As noted earlier a setup cost S_j is incurred once for product j in period m if the product is produced in period m (using any type of capacity), that is if $\sum_{k=1}^K \sum_{n=m}^M X_{jkmn}$ is positive, $j=1, \dots, J$, $m=1, \dots, M$.

Constraints (2.5) are nonnegativity constraints.

This formulation provides computational advantage over the standard mixed integer linear programming formulation, since the structure of the transportation subproblems can be

exploited. One shortcoming of this approach is that it is not as flexible as the mixed integer linear programming formulation. The fixed charge transportation problem formulation cannot be modified to include workforce size as decision variable or multiple capacity constraints without losing the fixed charge transportation problem structure.

2.2 TABLEAU FORMATION OF PROBLEM

As noted earlier the SLCR problem (2.1)-(2.5) is a variation of fixed charge transportation problem. The sources (rows) or origin in the transportation tableau are C_{km} the quantity of type k production capacity available in period m , $k=1, \dots, K$, $m=1, \dots, M$. The sinks (columns) or destinations are D_{jn} quantity of product j demanded in period n , $j=1, \dots, J$, $n=1, \dots, M$ (see Figure 2). The rows are ordered as $(1,1), \dots, (K,1), \dots, (1,M), \dots, (K,M)$. The columns are ordered as $(1,1), \dots, (J,1), \dots, (1,M), \dots, (J,M)$.

The cell in row (k,m) , $1 < k < K$, $1 < m < M$, and column (j,n) , $1 < j < J$, $1 < n < M$, will be denoted $((k,m),(j,n))$. The flow assigned to this cell corresponds to the variable X_{jkmn} , and the value of the setup variables Y_{jm} are implicitly

		Product Demands in Periods											
		1				2							
		1 2 .. J				1 2 .. J				1 2 .. J			
Inadmiss. cell	1												
	2												
	K												
Capac. Avail. in Period	1	/	/	/	/								
	2	/	/	/	/								
	..	/	/	/	/								
	K	/	/	/	/								
		/	/	/	/	/	/	/	/				
	1	/	/	/	/	/	/	/	/				
	2	/	/	/	/	/	/	/	/				
	..	/	/	/	/	/	/	/	/				
	K	/	/	/	/	/	/	/	/				

Figure 2. Tableau Associated with Fixed Charge Transportation Problem.

determined by relationship (2.4). Hence a feasible tableau solution corresponds to a feasible solution to (2.1)-(2.5).

2.3 COMPUTATIONAL COMPLEXITY OF THE PROBLEM

In this section it is shown that the SCLR problem (2.1)-(2.5) is NP-hard, by observing that the problem contains as a special case a known NP-hard problem. The argument is similar to that developed by Bitran and Yanasse (1982).

Proposition:

The capacitated multi-product scheduling problem with no specified pattern of capacity, setup cost, holding cost and capacity cost is NP-hard.

Consider the partition problem: "Given a set of N integers $A_n, n=1, \dots, N$ find $U \subseteq \{1, \dots, N\}$ such that,

$$\sum_{n \in U} A_n = \sum_{n \in \{1, \dots, N\} - U} A_n."$$

The partition problem has been shown to be NP-Complete (Garey and Johnson(1977)). The above proposition is proved by showing that the capacitated multi-product scheduling problem contains the partition problem as a special case.

Assume without loss of generality that $A_n, n=1, \dots, N$ is a set of positive integers. Consider a 2-product problem with the following parameters.

$$D_{1n} = D_{2n} = 0 \quad n=1, \dots, (N-1)$$

$$D_{1N} = D_{2N} = \left\lfloor \frac{1}{2} \right\rfloor \left(\sum_{n=1}^N C_{1n} \right)$$

$$S_1 = S_2 = 1$$

$$P_1 = 0$$

$$H_1 = H_2 = 0$$

$$C_{1n} = A_n \quad n=1, \dots, N$$

Since the problem constructed has $D_{1N} + D_{2N} = \sum_{n=1}^N C_{1n}$ the available capacity C_{1n} , $n=1, \dots, N$ will have to be used. Hence its optimal value is not smaller than N and will be equal to N if and only if the sequence C_{1n} , $n=1, \dots, N$ or equivalently A_n , $n=1, \dots, N$ can be partitioned such that,

$$\sum_{n \in U} C_{1n} = \sum_{n \in \{1, \dots, N\} - U} C_{1n} = \left\lfloor \frac{1}{2} \right\rfloor \left(\sum_{n=1}^N C_{1n} \right)$$

that is, if and only if the partition problem has a solution. Consequently the capacitated multi-product scheduling problem is NP-hard.

CHAPTER III

LITERATURE SURVEY

The survey consists of research in two major areas relevant to this dissertation research:

1. capacitated product scheduling problem;
2. fixed charge problem.

3.1 CAPACITATED PRODUCT SCHEDULING PROBLEM

The capacitated product scheduling problems can be classified into two groups:

1. single level constrained resource problem;
2. multiple level constrained resource problem.

3.1.1 Single Level Constrained Resource Problem

In Chapter II we have shown that the single level constrained resource problem is NP-hard. The problem is also hard in a practical sense since optimal solution algorithms have failed to solve all but very small problems within reasonable computation times (see e.g. Barani, Van Roy and Wolsey (1984)).

There are few researchers who have formulated and solved problems similar to the problem discussed in Chapter II. Most researchers have formulated and attempted to solve special cases of the SLCR problem. Researchers who have formulated the

SLCR problem similar to the formulation in this dissertation ignore overtime capacity. In this dissertation we consider regular time capacity as well as overtime capacity to be constrained.

In the literature two common class of algorithms used to solve the SLCR problem are:

1. linear programming based;
2. dynamic programming based.

3.1.1.1 Linear Programming Based Algorithms

Linear programming based algorithms were initiated by Manne (1958). The algorithms are based on set partitioning, i.e. selecting only one schedule for each product from a set of suggested schedules. The approach is to relax the 0-1 set partitioning problem to a linear programming problem and round off the resulting solutions.

Manne (1958) suggested a linear programming model as an approximation to the SLCR problem. He proposed a formulation in which binary setup variables are replaced by binary variables for each production sequence. Although this significantly increases the number of decision variables the number of constraints is modest. The advantage, however, is that the model can be solved with linear programming rather than mixed integer linear programming. Manne's (1958) approach does not guarantee optimal solutions.

Dzielinski, Baker and Manne (1963) showed through simulation tests that Manne's approach is favorable to simply applying the Economic Order Quantity model. Dzielinski and Gomory (1965) used the Dantzig-Wolfe (1961) decomposition principle (in order to accomplish a reduction in the large number of constraints in the original problem) and improved the computational efficiency of Manne's (1958) formulation. Lasdon and Terjung (1967) developed a column generation technique and handled multiplicity of constraints by the generalized upper bounding procedure of Dantzig and Van Slyke (1967). The technique was shown to be more efficient than Dantzig-Wolfe (1961) decomposition procedure.

Bahl (1983) presented two heuristics based upon Lasdon and Terjung's (1967) column generation algorithm. Instead of using Wagner-Whitin (1958) condition in the column generation he uses a simple lot sizing routine. Bahl and Ritzman (1984) proposed a model that combines fixed-ordering interval techniques (such as lot-for-lot, periodic order quantity or fixed period requirements rule) with the ability of Manne's (1958) formulation to solve SLCR problems with capacity constraints for several work centers.

3.1.1.2 Dynamic Programming Based Algorithms

One of the earlier dynamic programming formulation was presented by Zangwill (1966) for the single stage and

multistage lot sizing problem. He developed a formulation which represents the lot sizing problems as networks. He used the concept of concave cost network analysis to analyze the singlestage and multistage models.

Silver and Meal (1973) presented a heuristic for a single-product problem with capacity constraints. The heuristic essentially combines production for several periods into one lot as long as average cost per period keeps decreasing. Eisenhut (1975) developed a dynamic lot sizing heuristic based on marginal analysis of the holding costs and setup costs. The heuristic determines lot sizes on a period by period basis and allows for fluctuating and uncertain demand patterns. Eisenhut's (1975) heuristic does not always obtain a feasible solution.

Lambrecht and Vanderveken (1979) further extended Eisenhut's (1975) heuristic by improving the marginal cost determination and feasibility assurance procedure. The heuristic provides a feedback mechanism to remove infeasibility (which might result if Eisenhut's (1975) procedure is used) by transferring demand to earlier periods. Dixon and Silver (1981) developed a "greedy" heuristic which always generates a feasible solution, if one exists. The heuristic is based on Silver-Meal (1973) lot sizing technique and a set of feasibility conditions.

The heuristics presented by Lambrecht and Vanderveken (1979) and Dixon and Silver (1981) proceed by constructing a schedule period by period, starting with period 1. A binary knapsack problem is solved to determine which production lot should be scheduled for production.

The algorithm presented by Dogramaci, Panayiotopoulos and Adam (1981) starts with a lot-for-lot schedule for each product. It then involves a number of rather intricate and intertwined feasibility-seeking steps and improvement steps. It looks at all possible lot combinations and merges the two lots which yields the largest absolute savings.

Karni and Roll (1982) presented a heuristic for single-product SLCR problem and then extended the heuristic to multi-product SLCR problem. Karni and Roll (1982) start with the optimal Wagner-Whitin (1958) solution for each product and then attempt to find feasibility and improvement by transferring production from one period to another. Production is shifted left (to an earlier period) or right (to a later period) in order to achieve one of two aims (1) elimination of infeasibility at a minimum cost, or (2) reduction of total setup and carrying cost by changing the solution structure.

Maes and Wassenhove (1986) compare the performance of three heuristics (Lambrecht and Vanderveken (1979), Dixon and Silver (1981), Dogramaci, Panayiotopoulos and Adam (1981)) on a large set of test problems and analyze their difference in

performance for various problem parameters (e.g. cost structure, demand variability, tightness of capacity constraints, etc.).

3.1.2 Multiple Level Constrained Resource Problem

Few papers have considered the multiple level constrained resource problem. Most of the research is for special cases of the multiple level constrained resource problem. Lambrecht and Vanderveken (1978b) proposed two heuristic algorithms for the multiple level constrained resource problem. The formulation assumes constraint on capacity only at the highest level (end item). Rest of the levels in the problem is treated as a separate problem without capacity constraints.

Gabbay (1979) presented an optimizing algorithm for the multiple level constrained resource problem. The model includes a very restrictive assumption about the production time required by each product on each machine. The model assumes that a job must take the same fraction of available time on each machine.

Zahorik, Thomas and Trigeiro (1984) presented an algorithm for the multiple level problem. Capacity is assumed to be constrained only at one level. They proposed an optimizing algorithm for the 3-period problem and a heuristic for the problem with more than three periods based on the optimizing algorithm.

The multiple level constrained resource problem has been discussed in a series of papers by Billington, McClain, Thomas and Weiss. The first of these papers (Billington, McClain and Thomas (1981)) presented a mixed integer linear programming formulation for the problem and addressed the question of product structure compression. The model includes setup costs overtime, undertime and capacity constraints. No algorithm is proposed in this paper to solve the problem.

Billington, McClain and Thomas (1983) proposed a model similar to the one discussed in Billington, McClain and Thomas (1981). The major difference between the models discussed by the authors is that in one paper (1981) a group of similar products are assumed to have a setup and in the other paper (1983) individual products are assumed to have a setup.

The last of this set of papers by Billington, McClain, Thomas and Weiss (1984)) presented a branch and bound procedure for the special case of the problem where there is a bottleneck work center. A heuristic based on Lagrangian relaxation is used to solve the subproblems to determine lot sizes. A smoothing routine is then used to adjust production to fit capacity constraints.

3.2 FIXED CHARGE PROBLEM

In the literature the solution procedure for the fixed charge problem consists of heuristics and exact solution algorithms. Research in the fixed charge problem area assume that a fixed cost is incurred whenever the associated continuous variable becomes positive. There are a few papers which have considered the "traditional" fixed charge transportation problem. In the "traditional" fixed charge transportation problem each cell in the transportation tableau is assumed to have fixed cost as well as variable cost. Some authors have considered the case of fixed charge transportation problem where there is a fixed cost associated with every source.

The problem considered in this dissertation is formulated as a variation of the fixed charge transportation problem. In the SLCR problem formulated in Chapter II a fixed charge is shared by a group of cells in the transportation tableau. None of fixed charge transportation problems discussed in literature have similar property. The existing algorithms for the fixed charge transportation problem do not have a provision for allowing group of cells to share the same fixed charge. Hence none of the existing algorithms for the fixed charge transportation problem can be used to solve the SLCR problem.

The solution procedure for the fixed charge transportation problem consists of heuristics and exact solution algorithms. Most exact algorithms cannot solve very large problems. The most efficient computer code to date (Kennington and Unger (1976)) was used to solve totally dense problems with dimensions of up to 10×20 . Barr et. al. (1981) designed an algorithm for sparse large scale problems (problems where not all cells exist and have fixed charges on only a portion of the existing arcs) which has solved problems with 3,000 constraints and 1,200 fixed charge variables.

3.2.1 Heuristics

One of the initial results on the linear fixed charge problem which appeared in 1954 was by Hirsh and Dantzig. They showed that, for any fixed charge problem, an optimal solution occurs on an extreme point of the constraint set. Since fixed charge transportation is a special case of linear fixed charge problem the result holds and the solution will lie on an extreme point of the set. Balinski (1961) presented an approach to solve the fixed charge transportation problem where he replaced the non-linear fixed charge objective function by an approximate linear objective function and solved the resulting problem using transportation algorithm.

Kuhn and Baumol (1961) presented a heuristic for

the fixed charge transportation problem. The heuristic was designed primarily to handle problems involving small number of origins and destinations. The heuristic obtained a solution by forcing a highly degenerate solution. They suggested an approximate solution by making small adjustments to supplies and demands.

The fact that the optimal solution to the fixed charge problem lies at an extreme point of the convex set has led to the development of heuristics which use adjacent extreme point techniques.

Cooper and Drebes (1967) presented an extreme point heuristic in which the objective function is modified at certain stages in the algorithm (in particular the linear costs are modified). At other stages in the heuristic the algorithm removes from the basis the variable with the largest fixed charge and insert variable with the smallest fixed charge. Cooper and Olson developed a heuristic which is a modification of Cooper-Drebes heuristic.

Denzler (1969) suggested an approach which retains most of the decision rules of simplex method while also taking into account the fixed charges associated with variables entering and the variables leaving the basis at each iteration. The approach omits consideration of degeneracy.

Steinberg (1970) developed several adjacent extreme point heuristics. The algorithm attempts to find better solutions by moving away from the current local minimum in as few iterations as possible. Steinberg (1970) also suggested formulating the

problem as a separable programming problem and developed a branch and bound algorithm for an exact solution of the fixed charge problem.

Walker (1976) proposed a heuristic which was an extension to Denzler's (1969) heuristic. The criteria for selection of arcs to enter the basis differs from Denzler (1969) and Cooper in that it also considers degeneracy. Walker's heuristic identifies an initial local minimum and searches for an improved solution by jumping over adjacent extreme points.

3.2.2 Exact Algorithms

One of the earliest exact algorithm, presented by Murty (1968), was based on searching among the adjacent extreme points of the transportation problem. Murty (1968) suggested that the linear fixed charge problem can be solved by ranking the vertices of the polyhedral constraint set in increasing order of their continuous objective values and then adding fixed charges to determine an optimal solution.

Gray (1968) developed an alternate approach to Murty's (1968) approach. Gray's (1968) procedure decomposes the fixed charge problem into a master integer problem and series of transportation subproblems. To reduce the number of vertices that need to be examined, bounds are established on the maximum and minimum values of the total fixed cost, and feasibility conditions are used extensively.

Steinberg's algorithm (1970) was based on branch and bound techniques. McKeown (1975) presented an improvement on Murty's

(1968) procedure to solve the linear fixed charge problem. Kennington (1976) presented a branch and bound algorithm for the fixed charge transportation problem. He also presented a study of the effect of problem parameters on computational time and core storage requirements. Kennington and Unger (1976) transformed the fixed charge transportation problem into an equivalent problem with a special network interpretation, and developed a branch and bound procedure for solving the fixed charge transportation problems. Simple penalties are constructed from the optimal solution of the associated transportation problem. These penalties are used in the branch and bound tasks of fathoming and guiding separation.

Fisk and McKeown (1977) presented an algorithm for a variation of the fixed charge transportation problem in which there are only fixed costs to be incurred when a route is opened. Their procedure is enhanced by using 0-1 knapsack problems to determine bounds on partial solutions.

McKeown (1981) developed a Branch and Bound procedure where bounds are calculated separately on the sum of fixed costs and on the continuous costs. Barr, Glover and Klingman (1981) proposed a branch and bound algorithm which exploited the absence of full problem density. The fathoming tests are similar to Geoffrion and Marsten (1972). The algorithm employs "up" and "down" penalties proposed by Driebeek (1966) and Tomlin (1971).

Cabot and Erenguc (1984) presented branch and bound

algorithms where Langragian relaxation is applied to develop "up" and "down" penalties for choosing branching variables and fathoming nodes.

CHAPTER IV

HEURISTIC

This chapter presents the heuristic for solving the SLCR problem. The heuristic works with basic feasible solutions to the transportation problem (2.1)-(2.5). (The associated values of the setup variables are determined implicitly by constraints (2.4).) It consists of three phases. The first phase introduces into the basis some variables that are required to meet the demands of the first period. The second phase is a single pass "greedy" algorithm that selects the remaining basic variables to provide a good starting solution. The third phase makes an attempt to improve the solution obtained in the second phase by replacing variables in the basis.

4.1 PHASE I

4.1.1 Description of Phase I

This phase determines how the demands for all products in the first period are to be met. If D_{j1} is positive, then any feasible solution to the problem will have production of product j scheduled in the first period. The demands for the products in the first period are met by using the least expensive type of capacity available.

The output from this phase is either, assignment of values to X_{jkl1} , $j = 1, \dots, J$, $k = 1, \dots, K$, such that $\sum_{k=1}^K X_{jkl1} = D_{j1}$ or indication that the problem is infeasible.

At any point during PHASE I or PHASE II, let $\bar{D}_{jn} = D_{jn} - \sum_{k=1}^K \sum_{m=1}^n X_{jkmn}$ be the remaining unmet demand for product j in period n , $j = 1, \dots, J$, $n = 1, \dots, N$. Similarly let $\bar{C}_{km} = C_{km} - \sum_{j=1}^J \sum_{n=1}^M X_{jkmn}$ represent the unused remaining capacity of type k in period m , $k = 1, \dots, K$, $m = 1, \dots, M$.

4.1.2 Steps of Phase I

The steps of Phase I of the heuristic are as follows:

Step 1: Initialization

Initially $X_{jkmn} = 0$

Set $j = 1$, $k = 1$

Proceed to Step 2.

Step 2: (a) If $j > J$, go to PHASE II.

(b) If $k > K$, STOP. The problem is infeasible.

Otherwise

Assign value to variable associated with cell

$((k,1),(j,1))$

$X_{jkl1} = \text{Minimum } (\bar{D}_{j1}, \bar{C}_{k1})$

Proceed to Step 3.

Step 3: Update demand $\bar{D}_{j1}$ and capacity $\bar{C}_{k1}$.

(a) If $\bar{D}_{j1} \leq \bar{C}_{k1}$

$$\bar{C}_{k1} \leftarrow \bar{C}_{k1} - \bar{D}_{j1}$$

$$\bar{D}_{j1} \leftarrow 0$$

Mark Column (j,1)

$$j = j+1$$

Go to Step 2.

(b) If $\bar{D}_{j1} > \bar{C}_{k1}$

$$\bar{D}_{j1} \leftarrow \bar{D}_{j1} - \bar{C}_{k1}$$

$$\bar{C}_{k1} \leftarrow 0$$

Mark row (k,1)

$$k = k+1$$

Go to Step 2.

4.2 PHASE II

4.2.1 Description of Phase II

The second phase of the heuristic is similar to the Vogel's approximation method. The difference, here, is that the cost coefficients are designed to take into account fixed cost associated with each cell in the tableau. Also, since in our problem a group of cells share a fixed charge, the costs of all the cells which share the fixed charge are updated, whenever one of these cells becomes nonzero.

As in Vogel's approximation method, Phase II of the heuristic bases the selection of cells for its initial feasible solution on the difference between the two lowest cost cells

leaving an origin or entering a destination. The difference (penalty) indicates where departure from lowest cost allocations will bring the highest increase in cost. The maximum possible value is assigned to the variable associated with the cell having the largest penalty.

The cost coefficients used for nonbasic cells are "greedy" costs. The cost coefficient for a cell represents the average rate of increase in the objective function, that will be incurred when the corresponding variable enters the basis.

If cell $((k,m),(j,n))$ is chosen to enter the basis, X_{jkmn} will take on a value equal to minimum of $\bar{D}_{jn}$ or $\bar{C}_{km}$. If a setup has already been incurred for product j in period m , for each unit increase in X_{jkmn} the objective function will increase by,

$$(P_k + (n-m) * H_j) \quad (4.1)$$

On the other hand if the setup has not been incurred for product j in period m then the objective function will increase by,

$$X_{jkmn} * (P_k + (n-m) * H_j) + S_j \quad (4.2)$$

In this case the average rate of increase in the objective function per unit of demand j met in period n is,

$$P_k + (n-m) * H_j + S_j / \text{Min}(\bar{D}_{jn}, \bar{C}_{km}) \quad (4.3)$$

Thus at any stage of Phase II, the cost coefficients of a cell $((k,m),(j,n))$ will be defined by (4.1) if a setup for product j in period m has already been incurred and by (4.3) if such a setup has not been incurred. In the latter case the cost coefficient is simply the slope of the linear approximation of the portion of the objective function involving X_{jkmn} , as shown in Figure 3.

4.2.2 Steps of Phase II

The steps of Phase II of the heuristic are as follows:

Step 1: Initialization of nominal cost for cells in tableau

Compute the nominal cost R_{jkmn} for admissible cells $((k,m),(j,n))$ (i.e. cells with $m < n$).

$$R_{jkmn} = P_k + (n-m) * H_j \quad \begin{matrix} j=1, \dots, J \\ k=1, \dots, K \end{matrix}$$

$$\text{if } D_{jn} = 0 \text{ or } \sum_{k=1}^K X_{jk11} > 0 \quad \begin{matrix} m=2, \dots, M \\ n=m, \dots, M \end{matrix}$$

$$R_{jkmn} = P_k + (n-m) * H_j + S_j / \text{Min}(\bar{D}_{jn}, \bar{C}_{km})$$

$$\text{if } \text{Min}(\bar{D}_{jn}, \bar{C}_{km}) > 0 \text{ and } \sum_{k=1}^K X_{jk11} = 0$$

For inadmissible cells (i.e. cells with $m > n$), define the cost as L , where L is a very large number.

Proceed to Step 2.

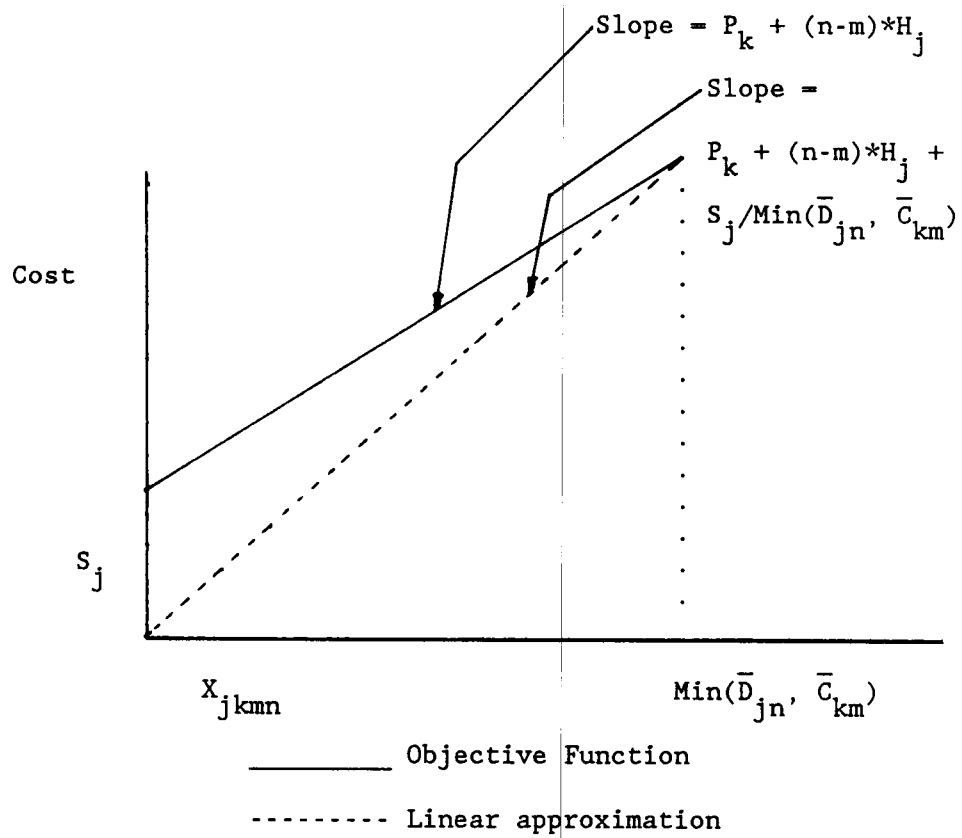


Figure 3. Nature of Objective Function and Linear Approximation of Objective Function.

Step 2: Computation of row penalties

For each unmarked row compute penalty $E_{(k,m)}$.

$$E_{(k,m)} = A'_{(k,m)} - A_{(k,m)}$$

where $A_{(k,m)}$ is the lowest nominal cost in row (k,m) and $A'_{(k,m)}$ is the second lowest cost in row (k,m) .

Proceed to Step 3.

Step 3: Computation of column penalties

For each unmarked column compute penalty $G_{(j,n)}$ as follows:

$$G_{(j,n)} = B'_{(j,n)} - B_{(j,n)}$$

where $B_{(j,n)}$ is the lowest nominal cost in column (j,n) and $B'_{(j,n)}$ is second lowest cost in column (j,n) .

Proceed to Step 4.

Step 4: Selection of cell $((k',m'),(j',n'))$ to enter basis

$$\text{Let } E_{(k',m')} = \text{Max } (E_{(k,m)})$$

$$\text{Let } G_{(j',n')} = \text{Max } (G_{(j,n)})$$

$$\text{If } E_{(k',m')} > G_{(j',n')}$$

$$R_{j'k'm'n'} = \text{Min } (R_{jk'm'n'}) \quad \begin{array}{l} j=1, \dots, J \\ n=1, \dots, M \end{array}$$

where $((k',m'),(j',n'))$ is the cell selected to enter the basis.

$$\text{If } E_{(k',m')} < G_{(j',n')}$$

$$R_{j'k'm'n'} = \text{Min } (R_{j'kmn'}) \quad k=1, \dots, K$$

$m=1, \dots, M$

where $((k', m'), (j', n'))$ is the cell selected to enter the basis.

Proceed to Step 5.

Step 5: Assign value to cell $((k', m'), (j', n'))$ selected to enter the basis

$$X_{j', k', m', n'} = \text{Minimum} (\bar{D}_{j', n'}, \bar{C}_{k', m'})$$

If $X_{j', k', m', n'} = 0$ mark column (j', n') and go to Step 2. Otherwise proceed to Step 6.

Step 6: Update row capacity and column demand

$$\begin{aligned} \text{If } \bar{D}_{j', n'} &\leq \bar{C}_{k', m'} \\ \bar{C}_{k', m'} &\leftarrow \bar{C}_{k', m'} - \bar{D}_{j', n'} \\ D_{j', n'} &\leftarrow 0 \end{aligned}$$

Mark column (j', n') .

$$\begin{aligned} \text{If } D_{j', n'} &> C_{k', m'} \\ \bar{D}_{j', n'} &\leftarrow \bar{D}_{j', n'} - \bar{C}_{k', m'} \\ \bar{C}_{k', m'} &\leftarrow 0 \end{aligned}$$

Mark row (k', m') .

Proceed to Step 7.

Step 7: If all columns are marked, which implies that all demands are met, STOP. Otherwise proceed to Step 8.

Step 8: Recompute nominal cost for cells

Compute nominal costs for cells which share a

fixed charge with the cell $((k',m'),(j',n'))$.

$$R_{j',km'n} = [P_k + (n-m') * H_{j'},] \quad \begin{matrix} k=1,\dots,K \\ n=1,\dots,M \end{matrix}$$

Proceed to Step 2.

Phase II uses a "greedy" approach to add cells to the basis. At each step, the cell which would result in highest penalty if not included in the basis is added to the basis. The underlying logic is that, the approach attempts to add that cell to the basis, which will result in the "best" use of available capacity.

From the definition of R_{jkmn} it can be seen that the least expensive cells will be those for which $m = n$ and D_{jn} is large or those cells for which a setup has already been incurred. Phase II will tend to either schedule new setups for a product in the period of highest demand for that product, or depending upon the magnitude of the holding cost, it may attempt to take advantage of existing setups. In either case it will tend to choose the least expensive type of capacity available.

4.3 PHASE III

Phase III employs a heuristic to improve the solution obtained in Phase II. The algorithm is similar to primal network simplex algorithm. The major difference is that the heuristic takes into consideration fixed costs as well as variable costs in pricing pivots. The algorithm converges to a

local optimum (in sense the no further improvement in the objective function can be achieved by replacing a single variable in the basis).

4.3.1. Review of Primal Simplex Algorithm

In this section we briefly review the primal simplex method. Small bold letters will denote column vectors, and a prime indicates transpose. Large bold letters denote matrices; superscripts denote columns and subscripts denote rows.

Consider

$$\begin{aligned} \text{Min} \quad & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{1}$$

$\mathbf{A}$ is a m by n matrix of technological coefficients. $\mathbf{A}$ may be partitioned $\mathbf{A} = (\mathbf{B}, \mathbf{N})$, with $\mathbf{B}$ an m by m matrix of linearly independent columns which is called a basis.

Given a feasible basis there always exists a unique $\mathbf{x}^*$ such that

$$\mathbf{Bx}^* = \mathbf{b} \tag{2}$$

In terms of this $\mathbf{x}^*$ there is always a current basic solution $\mathbf{x}^0 = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}$ and upon partitioning $\mathbf{c}$ in the same manner

as $\mathbf{A}$, one has

$$\mathbf{c}'\mathbf{x}^0 = (\mathbf{c}'_{\mathbf{B}}, \mathbf{c}'_{\mathbf{N}}) \begin{bmatrix} \mathbf{x}^* \\ 0 \end{bmatrix} = \mathbf{c}'_{\mathbf{B}}\mathbf{x}^* \tag{3}$$

Any generic solution x satisfying the constraints can be rewritten in terms of the basic solution:

$$A_x = (B, N) \begin{bmatrix} x_B \\ x_N \end{bmatrix} = Bx_B + Nx_N = b \quad (4)$$

Further, since B is a basis, there exists a unique transformation Z such that:

$$BZ = N \quad (5)$$

Multiplying (5) by x_N and subtracting from (2) yields:

$$B(x^* - Zx_N) + Nx_N = b \quad (6)$$

Subtracting (6) from (4) yields: $B(x_B - [x^* - Zx_N]) = 0$

Since the columns of B are linearly independent, $x_B = x^* - Zx_N$ and the general solution becomes

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} x^* - Zx_N \\ x_N \end{bmatrix} \quad (7)$$

With this form it is easy to compare the value of x to any current x^0 and identify an improved solution when one exists.

The value of the generic solution (using (3) and (5)) is:

$$\begin{aligned} c'x &= c'_B x_B + c'_N x_N = c'_B (x^* - Zx_N) + c'_N x_N \\ &= c'_B x^* + (c'_N - c'_B Z)x_N \\ &= c'_B x^* + (c'_N - u'N)x_N \end{aligned} \quad (8)$$

where u (often called the dual solution or simplex multiplier) is the solution of

$$u'B = c'_B \quad (9)$$

From (8) it is clear (since $x_N \geq 0$) that a necessary condition

for an improved solution is that there exists a column of N, N^k , such that:

$$c_k - u'N^k < 0 \quad (10)$$

With such a column consider a specific vector obtained from (7) by taking all components of x_N zero except x_k ,

$$x' = (x^* - Z^k x_k, 0, \dots, x_k, \dots, 0). \quad (11)$$

As a function of $x_k \geq 0$, this vector must by its derivation satisfy the explicit constraints

$$Ax = b$$

For components $z_{ik} > 0$ this requires $x_i - z_{ik}x_k \geq 0$

$$\text{or } x_k = x_i^* / z_{ik} \quad (12)$$

Taking $x_k = x_i^* / z_{ik}$ with $z_{ik} > 0$ in (11) leads to the exchange of B^i and N^k in the partition between basic and non-basic columns and the new basic solution

$$\begin{aligned} \tilde{x}_r &= x_r^* - z_{rk}(x_i^* / z_{ik}), \quad r \neq i; \text{ and} \\ \tilde{x}_i &= x_i^* / z_{ik}. \end{aligned}$$

Assume that there is a current basis B , a current solution x^* to $Bx^* = b$, and a current u of $u'B = c'_B$. An iteration of Simplex procedure is summarized:

Step 1: (Pricing) Select a candidate variable to enter the basis with $(c_k - u'N^k) < 0$.

Step 2: (Ratio test) Let i be a variable where i is determined as follows:

$$x_i / z_{ik} = \text{Min} \quad (x_r / z_{rk} : z_{rk} > 0) \\ 1 \leq r \leq m.$$

Step 3: (Pivot) Update the primal solution $\tilde{x} = x^* - x_k N^k$.

Change the basis by exchanging B^i and N^k .

Find the new dual solution $u'B = c'_B$.

The algebraic specialization of the Simplex method for transshipment problems, which is the basis of the Phase III of the heuristic, is given in Bradley, Brown and Graves (1977).

4.3.2 Description of Phase III

Since the SLCR problem has been formulated as a variation of fixed charge transportation problem, if a nonbasic variable is introduced into the basis, the total variable cost as well as the total fixed cost of the solution may be affected. The total fixed cost (setup cost) incurred may be affected because of the following:

1. the setup cost associated with variables which turn positive from zero;
2. the setup cost associated with variables which are reduced to zero from a positive value.

Suppose the setup variable Y_{jm} associated with a production X_{jkmn} is zero. If the pivot turns X_{jkmn} positive from zero it will result in an additional setup. Suppose the setup variable Y_{jm} has value 1. If the pivot reduces all production variables that share this setup to zero it will

result in elimination of the setup.

The major difference in the primal simplex algorithm and Phase III is in the pricing strategy. In Phase III the nonbasic variable selected to enter the basis at each iteration is the one that will result in the maximum reduction of the total cost. Given a basic feasible solution if a nonbasic variable X_{jkmn} enters the basis, the resulting change in the objective function value is given by,

$$\Delta * W_{jkmn} + \sum_{j=1}^J \sum_{m=1}^M S_j * Y_{jm}^* - \sum_{j=1}^J \sum_{m=1}^M S_j * Y_{jm} \quad (13)$$

where

Δ = value at which nonbasic variable X_{jkmn} will enter the basis

W_{jkmn} = the reduced cost of the entering variable X_{jkmn} corresponding to the continuous portion of the objective function (2.1)

$$W_{jkmn} = U_{(k,m)} - V_{(j,n)} + (P_k + (n-m) * H_j)$$

$U_{(k,m)}$ = dual variables associated with source (k,m) considering only the variable cost associated with each cell.

$V_{(j,n)}$ = dual variables associated with destination (j,n) considering only the variable cost associated with each cell.

$P_k + (n-m) * H_j$ = variable cost associated with variable X_{jkmn} .

Y_{jm} is the current value of the setup variable for product j in period m and Y_{jm}^* is the value of the setup variable after

X_{jkmn} is brought into the basis (Y_{jm} and Y_{jm}^* are determined implicitly by constraints (2.4)).

4.3.3 Steps of Phase III

Step 1: Choose a starting basic feasible solution.

Set X^I - the basic feasible solution obtained from Phase II and Z^I - the associated objective function value.

Step 2: Compute dual variables associated with source $U(x,m)$ and destination $V_{(j,n)}$ (considering only variable cost for each cell).

Set $U_{(k,m)} = 0$.

If $U_{(k,m)}$ has been computed, $V_{(j,n)}$ has not been computed, and cell $((k,m),(j,n))$ is a basic cell, then set $V_{(j,n)} = U_{(k,m)} + [P_k + (n-m)*H_j]$.

If $V_{(j,n)}$ has been computed, and cell $((k,m),(j,n))$ is a basic cell, then set $U_{(k,m)} = V_{(j,n)} - [P_k + (n-m)*H_j]$.

Repeat step 2 until all $U_{(k,m)}$, $(k,m) = (1,1), \dots, (K,1), \dots, (1,M), \dots, (K,M)$, and $V_{(j,n)}$, $(j,n) = (1,1), \dots, (J,1), \dots, (1,M), \dots, (J,M)$, have been computed.

Step 3: Pricing

For all nonbasic variables, using (13), compute the change in the objective function that would result if the nonbasic variable is introduced into the basis.

If no entering nonbasic variable will result in a reduction in the objective function go to step 5.

Otherwise select X_{jkmn} , the nonbasic variable which will result in maximum reduction in the objective function.

Step 4: Pivoting

Introduce nonbasic variable X_{jkmn} into the basis.
Update X^I , Z^I , and dual variables.

Go to Step 3.

Step 5: Stop. If the solution X_{jkmn} consists of a positive flow in any of the inadmissible cells (i.e. cells with cost L) the solution obtained is infeasible.

Otherwise the solution obtained is feasible.

CHAPTER V

EXACT SOLUTION ALGORITHM

This chapter describes the exact solution algorithm developed for the SLCR problem. It also discusses different bounding procedures used by the algorithm.

5.1 DESCRIPTION OF ALGORITHM

The exact solution procedure is based on implicit enumeration. In this technique each of the possible assignment of values to the logical variables is considered, either explicitly via computing the minimum value of the objective function associated with the assignment or implicitly by demonstrating that no solution having this assignment can be optimal.

In the implicit enumeration procedure an assignment of values to the logical variables is represented as a binary matrix $[Y_{jm}]$ of the logical variables. The dimension of this matrix is $J \times M$, where J is the number of products and M is the number of periods. A value of 1 for the row j and column m for example, indicates a setup for product j in period m .

The concept of a partial assignment is used in the enumeration process. A partial assignment is an assignment matrix in which some of the binary variables have been "fixed" at one of the admissible values, 0 or 1, and the remaining

binary variables are "free" to assume a value of either 0 or 1. A complete assignment to the assignment matrix, known as "completion", is constructed by assigning admissible values to all its free variables.

Associated with any completion is a standard transportation problem. Cells in the associated transportation tableau will be admissible or inadmissible depending upon the value of the setup variable Y_{jm} in the completion. Suppose for a given completion Y_{jm} is fixed at a value 1. The group of cells that share the same setup (Y_{jm}) will be admissible. On the other hand if Y_{jm} is fixed at value 0, the cells that share that setup will be inadmissible (the cells in the group are made inadmissible by assigning a very large cost to them). A feasible solution to this associated transportation problem will be one in which there will be flow assigned only to cells which are admissible (i.e a continuous variable X_{jkmn} will be assigned value only if cell $((k,m),(j,n))$ is admissible).

At any point during the enumeration process designated Y^I is a complete assignment of values to the logical variables called the "incumbent". The optimal objective value for the transportation problem associated with the "incumbent" is called Z^I . Z^I is an upper bound on the objective value of the optimal solution to the SLCR problem (2.1)-(2.5).

The implicit character of the algorithm stems from the

ability to determine information pertinent to optimality and feasibility from a specific partial assignment. For example,

1. by demonstrating the nonexistence of any completion of the partial assignment, with the objective value of the problem associated with the completion less than Z^I , or
2. by demonstrating nonexistence of a feasible solution to the problem associated with any completion to the partial assignment.

In either case, the partial assignment is considered to have been fathomed. If fathoming fails, we resort to explicit enumeration by fixing the value of the immediately next free variable (augmentation). If fathoming succeeds, we backtrack to a different partial assignment. If a completion to the binary matrix is constructed the associated transportation problem is solved optimally to determine the values of the continuous variables X_{jkmn} and to determine the objective function value. If the objective value is less than Z^I , then the incumbent Y^I and the objective value of the problem associated with the incumbent Z^I are updated with the improved solution.

In the procedure developed here the augmentation step always proceeds by column in the assignment matrix, starting from element in column 1 row 1 and ending at column M row J. The ordering of the variables for augmentation purposes will be $Y_{11}, \dots, Y_{J1}, \dots, Y_{1M}, \dots, Y_{JM}$. Thus in any partial assignment

the fixed variables will all precede (by column) the free variables. The bottommost fixed variable (of the rightmost column with fixed variables) will always be the most recently fixed variable. In the augmentation step the variable to be fixed is always fixed at 0 first. (The variable takes on a value of 1 in the backtracking step.)

5.1.1 Statement of the algorithm

The following sequence of steps characterize the basic enumerative procedure.

1. Initialization

Initialize Z^I equal to the objective function value of the best known feasible solution to the SLCP problem and Y^I to the corresponding assignment of values to logical variables. If there is no known feasible solution set $Z^I = -\infty$ and fix the first logical variable Y_{11} to 0.

Proceed to step 2.

2. Fathoming

Attempt to fathom the current partial assignment. If successful go to step 3.

Otherwise proceed to step 5.

3. Termination

If all elements of $[Y_{jm}]$ are fixed at 1, stop. Either Z^I

$< \infty$, in which case solution to the problem associated with the incumbent Y^I solves the problem (2.1)-(2.5), or $Z^I = \infty$, in which case no feasible solution exists for problem (2.1)-(2.5).

Otherwise proceed to step 4.

4. Backtrack

Alter the value of the variable Y_{jm} , the variable which was most recently fixed at value 0, to 1. Drop all elements which follow the altered variable. Update $[Y_{jm}]$ and go to step 2.

5. Updating

If the current partial assignment is a completion, solve the problem associated with the completion. If the optimal objective value of the problem associated with the completion is equal to a lower bound Z then STOP. The completion is the optimal setup combination and the optimal solution to the associated problem is the optimal solution to the SLCR problem. (The procedure for computing a lower bound on total cost is described in Chapter VI). If the optimal objective value of the problem associated with the completion is less than Z^I , replace Y^I and Z^I by the improved solution.

Otherwise proceed to step 6.

6. Augmentation

Fix the value of the immediately next free variable to 0.
Update $[Y_{jm}]$, and go to step 2.

The algorithm is both exhaustive and non-redundant; hence, it converges to an optimal solution in a finite number of steps.

The following problem example demonstrates the algorithm.

$$\text{Minimize } Y_{11} + Y_{12} + Y_{21} + Y_{22} \quad (5.1)$$

subject to:

$$X_{111} + X_{121} + X_{112} + X_{122} \leq 4 \quad (5.2)$$

$$X_{211} + X_{221} + X_{212} + X_{222} \leq 4 \quad (5.3)$$

$$X_{111} + X_{211} = 1 \quad (5.4)$$

$$X_{121} + X_{221} = 1 \quad (5.5)$$

$$X_{112} + X_{212} = 1 \quad (5.6)$$

$$X_{122} + X_{222} = 1 \quad (5.7)$$

$$Y_{jm} = \begin{cases} 1 & \text{if } \sum_{n=1}^2 X_{jmn} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} j = 1, 2 \\ m = 1, 2 \end{matrix} \quad (5.8)$$

$$X_{jmn} \geq 0$$

The solution to the above problem example using the implicit enumeration algorithm is shown in Table 1.

Table 1

An Example of the Implicit Enumeration Algorithm.

Stage	$[Y_{jm}]$	Z	Comments
i	0	0	Violates [5.4]. Backtrack.
ii	1	1	Enumerate further.
iii	1 0	1	Violates [5.5]. Backtrack.
iv	1 1	2	Enumerate further.
v	1 0 1	2	Enumerate further.
vi	1 0 1 0	2	Incumbent. Solve optimally problem associated with the completion. Solution is $X_{111} = -1$, $X_{121} = -1$ $X_{112} = 0$, $X_{122} = 0$, $Y_{11} = 1$, $Y_{21} = 1$, $Y_{12} = 0$, $Y_{22} = 0$. Backtrack.
vii	1 0 1 1	3	Inferior to incumbent. Backtrack.
viii	1 1 1	3	Inferior to incumbent. All Y_{jm} which are fixed have a value 1. End of enumeration. Solution is $X_{111} = -1$, $X_{121} = -1$, $X_{112} = 0$, $X_{122} = 0$, $Y_{11} = 1$, $Y_{21} = 1$, $Y_{12} = 0$, $Y_{22} = 0$.

5.2 BOUNDING PROCEDURES

The strength of any implicit enumeration algorithm lies in its fathoming process, since it is by fathoming that solutions are evaluated implicitly rather than explicitly. By exploiting the structure of fixed cost transportation problems we have developed efficient fathoming tests.

Fathoming tests are based on optimality and feasibility considerations relative to the problem of determining a completion to the partial assignment being enumerated and solution to its associated problem.

The algorithm developed here is for the case of SLCR problem in which there are two modes of production capacity (i.e. regular time and overtime). Backordering is not permitted. The modifications to handle additional modes of production and/or backordering are very straight forward. We have chosen the less general case for reasons of notational simplicity and because this case corresponds to the one which we, as well as most researchers in the field, have chosen for computational experiments.

5.2.1 Fathoming Procedure 1

If $\underline{Z}$ is a lower bound on the optimal objective values of the problems associated with all completions of the partial assignment being enumerated, then the partial assignment can be fathomed if $\underline{Z} > Z^I$, where Z^I is the optimal objective value of the problem associated with the "incumbent".

Let $Y_{j'm'}$ be the most recently fixed logical variable.

Let $\underline{S}$ be a lower bound on the setup cost, $\underline{H}$ be a lower bound on the holding cost and $\underline{P}$ be a lower bound on the production cost for the problems associated with all completions of the partial assignment being enumerated.

$\underline{Z}$, a lower bound on the optimal objective value for the problems associated with all completions of the partial assignment being enumerated. $\underline{Z}$ is given by,

$$\underline{S} + \underline{H} + \underline{P}.$$

5.2.1.1 Lower Bound on the Setup Cost

A lower bound on the setup cost for all completions to the partial assignment being enumerated can be determined by computing the setup cost that will be incurred by that partial assignment. $\underline{S}$ is given by,

$$\sum_{j=1}^J \sum_{m=1}^{m'-1} S_j * Y_{jm} + \sum_{j=1}^{j'} S_j * Y_{jm'}, \quad (5.9)$$

5.2.1.2 Lower Bound on the Holding Cost

A lower bound on holding cost that will be incurred by the problems associated with all completions of the partial assignment being enumerated can be determined by computing minimum holding cost that will be incurred by the problem associated with that partial assignment. Since backordering is not permitted, if setup for a product is not planned in any period, the demand for the product in that period has to be met by using capacity in periods prior to that period. This will result in holding cost for products carried over to the period

in which they are demanded. Hence if $Y_{jm} = 0$, $j = 1, \dots, J$, $m = n$, D_{jn} the demand for product j in period n has to be met by using C_{km} , the production capacity of type k in period m , where $k = 1, 2$ and $m < n$.

Suppose Y_{jn} is fixed at value zero the minimum holding cost that will be incurred in meeting demand D_{jn} will be equal to $H_j * D_{jn}$ (since demand for the products in period n will have to be satisfied by products which are carried forward by at least one period). Therefore a lower bound on holding cost H that will be incurred by the problems associated with all completions of the partial assignment being enumerated can be computed as shown below.

$$H = \sum_{j=1}^J \sum_{n=1}^{m'-1} D_{jn} [1-Y_{jn}] * H_j + \sum_{j=1}^{j'} D_{jm'} [1-Y_{jm'}] * H_j, \quad (5.10)$$

(Suppose $Y_{jn} = 0$ and $Y_{(j),(n-1)} = 0$, D_{jn} the demand for product j in period n has to be met by using C_{km} , the production capacity of period m , where $k = 1, 2$ and $m < (n-1)$. In this case the demand for product j in period n will have to be satisfied by the inventory of product j which is carried forward by atleast two periods. Hence a better lower bound on holding cost can be computed if information regarding setups for products in consecutive periods is stored. This can be done without much difficulty by using "efficient" data structures.)

5.2.1.3 Lower Bound on the Production Cost

A lower bound on the production cost for the problems

associated with all completions of the partial assignment being enumerated can be determined by assuming that the product demands are met by using the cheapest mode of production capacity available i.e. all regular time capacity available is used before any overtime capacity is used. It is clear that the production capacity in any period can be utilized only if setup for at least one product is planned during that period i.e. if $\sum_{j=1}^J Y_{jm} = 0$, the capacity available in period m cannot be utilized.

Let k=1 and k=2 index regular time and overtime modes of production capacity respectively.

Let C_{1m} denote regular time capacity available in period m.

Let C_{2m} denote overtime capacity available in period m.

Let P_1 denote cost per unit of regular time capacity.

Let P_2 denote cost per unit of overtime capacity.

Let $\bar{C}$ = maximum regular time capacity that can be utilized.

$$\bar{C} = \sum_{m=1}^{m'} C_{1m} * A_m + \sum_{m=m'+1}^M C_{1m} \quad (5.11)$$

$$\text{where } A_m = \begin{cases} 1 & \text{if } \sum_{j=1}^J Y_{jm} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$P = \left(\sum_{j=1}^J \sum_{n=1}^M D_{jn} - \bar{C} \right) * P_2 + \bar{C} * P_1 \quad \text{if } \sum_{j=1}^J \sum_{n=1}^M D_{jn} > \bar{C}$$

$$P = \sum_{j=1}^J \sum_{n=1}^M D_{jn} * P_1 \quad \text{otherwise}$$

$$\underline{Z} = \underline{S} + \underline{H} + \underline{P}$$

5.2.2 Fathoming Procedure 2

If $\underline{S}$ is a lower bound on the setup cost for the problems associated with all completions of the partial assignment being enumerated then that partial assignment can be fathomed if $\underline{S} > \bar{S}$ where $\bar{S}$ is an upper bound on setup cost for the optimal solution to the SLCR problem.

5.2.2.1 Upper Bound on Setup Cost

In problem (2.1)-(2.5) if fixed costs are equal to zero the problem reduces to an ordinary transportation problem. Hence if setup costs are set to zero, the objective function of the reduced problem becomes one of minimizing variable (production and holding) costs.

Let Z_V be the optimal objective function value of the reduced problem. Z_V will be a lower bound on the variable (sum of production and holding costs) costs for any feasible solution to the SLCR problem.

Let $\bar{S}$ be the upper bound on total setup cost incurred by the optimal solution to the SLCR problem.

$\bar{S}$ is given by,

$$Z^I - Z_V \quad (5.13)$$

where Z^I is the optimal objective value of the problem associated with the incumbent completion Y^I .

Theorem 1:

$\bar{S}$ is an upper bound on the setup cost for the optimal solution to the SLCR problem.

Let Z^0 be the optimum objective value for the SCLR problem

$$Z^0 = S^0 + Z_V^0$$

where S^0 is the total fixed (setup) cost incurred by the optimal solution and Z_V^0 is the total variable cost (production and holding costs) incurred by the optimal solution.

$$Z^I = S + Z_V \quad (\text{from 5.13})$$

Since Z^0 is optimal value $Z^I \geq Z^0$

$$\text{Therefore } \bar{S} + Z_V \geq S^0 + Z_V^0$$

$$\text{Therefore } \bar{S} \geq S^0 + (Z_V^0 - Z_V)$$

$$Z_V^0 - Z_V \geq 0 \quad (\text{since } Z_V \text{ is a lower bound on the variable costs for any feasible solution to the SLCR problem})$$

$$\text{Therefore } \bar{S} \geq S^0$$

Therefore $\bar{S}$ is an upper bound on the setup cost for the optimal solution to the SCLR problem.

5.2.3 Fathoming Procedure 3

A partial assignment matrix $[Y_{jm}]$ can be fathomed during the process of enumeration if it can be shown that all completions of $[Y_{jm}]$ will violate some demand constraint of the SCLR problem.

If $Y_{j',m'}$ is the most recently fixed setup variable, it may be possible to fathom by observing that demand for product j' in period m' cannot possibly be met. A partial assignment matrix $[Y_{jm}]$ can be fathomed if the cumulative capacity available in periods 1 through m' is less than the cumulative demand for product j' in periods 1 through m' .

It is clear that the production capacity in any period can be utilized only if setup for at least one product is planned during that period i.e. if $\sum_{j=1}^J Y_{jm} = 0$, C_{km} the production capacity available in period m cannot be utilized.

Therefore a partial assignment matrix $[Y_{jm}]$ can be fathomed if

$$\sum_{k=1}^K \sum_{m=1}^{m'} C_{km} * A_m - \left(\sum_{j=1}^J \sum_{n=1}^{m'-1} D_{jn} + \sum_{j=1}^{j'} D_{jm'} \right) < 0 .$$

$$\text{where } A_m = 1 \text{ if } \sum_{j=1}^J Y_{jm} > 0$$

$$= 0 \text{ otherwise}$$

CHAPTER VI

LOWER BOUND

This chapter describes a procedure for computing the lower bound on total cost for the SLCR problem. The lower bound is used for fathoming in the implicit enumeration algorithm and also for analyzing the performance of the heuristic. The chapter also discusses the effect of the problem parameters on the lower bound. Finally the chapter presents a comparison of the lower bound to the optimal solutions for those problems which were solved exactly.

6.1 PROCEDURE FOR COMPUTING LOWER BOUND

The objective function (2.1) of the SLCR problem (2.1)-(2.5) is to minimize the sum of setup, holding and production costs. A lower bound on total cost for the problem, is computed by solving two separate relaxations of the problem. In the first a lower bound on the sum of setup and holding costs and in the second a lower bound on production costs are found. The lower bound on the sum of setup and holding costs is determined by solving a relaxation of the SLCR problem with setup and holding costs in the objective function. The second computes a lower bound on the production costs solving a relaxation of the problem with only production costs in the objective function. Z , a lower bound on the objective value for the SLCR problem, can be obtained by summing the lower

bound on the setup and holding costs and the lower bound on the production costs.

6.1.1 Lower Bound on Setup and Holding Costs

This is found by solving a relaxation of the SLCR problem (2.1)-(2.5). If cost of production is set to zero the objective of the problem becomes one of minimizing the sum of setup and holding costs. The problem then reduces to a multi-product capacitated lotsizing problem. The SLCR problem with zero production cost is shown below.

Minimize Z^1 where $Z^1 =$

$$\sum_{j=1}^J \sum_{m=1}^M [S_j * Y_{jm}] + \sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M \sum_{n=m}^N [n-m] * [X_{jkmn} * H_j] \quad (6.1a)$$

subject to:

$$\sum_{k=1}^K \sum_{m=1}^M X_{jkmn} = D_{jn} \quad \begin{matrix} j=1, \dots, J \\ n=1, \dots, M \end{matrix} \quad (6.2a)$$

$$\sum_{j=1}^J \sum_{n=m}^M X_{jkmn} \leq C_{km} \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, M \end{matrix} \quad (6.3a)$$

$$Y_{jm} = 1 \text{ if } \sum_{k=1}^K \sum_{n=m}^M X_{jkmn} > 0 \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, M \end{matrix} \quad (6.4a)$$

= 0 otherwise

$$X_{jkmn} \geq 0 \quad \begin{matrix} j=1, \dots, J \\ k=1, \dots, K \\ m=1, \dots, M \\ n=1, \dots, M \end{matrix} \quad (6.5a)$$

If the production capacity constraints (6.3a) are relaxed

the problem reduces to several single-product uncapacitated lotsizing problem. The Wagner-Whitin algorithm (1958) can be used to solve these single-product uncapacitated lot sizing problems optimally. A lower bound (Z^1) on the objective value for the above relaxation can be obtained by minimizing the sum of setup and holding costs separately for each product using Wagner-Whitin algorithm and summing the individual objective values. The lower bound (Z^1) so obtained will be a lower bound ($\underline{Z}$) to the objective of problem (6.1a) and also a lower bound on the sum of setup and holding costs of the SLCR problem (2.1)-(2.5).

6.1.2 Lower Bound on Production Cost

If the setup costs and holding costs are set to zero in the SLCR problem (2.1)-(2.5), it reduces to a standard transportation problem. The objective of the reduced problem is to minimize the production costs. The reduced problem can be stated as shown below.

Minimize Z^2 where $Z^2 =$

$$\sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M \sum_{n=1}^M [X_{jkmn} * P_k] \quad (6.1b)$$

subject to:

$$\sum_{k=1}^K \sum_{m=1}^M X_{jkmn} = D_{jn} \quad \begin{matrix} j=1, \dots, J \\ n=1, \dots, M \end{matrix} \quad (6.2b)$$

$$\sum_{j=1}^J \sum_{n=1}^M X_{jkmn} \leq C_{km} \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, M \end{matrix} \quad (6.3b)$$

$$Y_{jm} = 1 \text{ if } \sum_{k=1}^K \sum_{n=1}^M X_{jkmn} > 0 \quad k=1, \dots, K \quad (6.4b)$$

$$= 0 \text{ otherwise} \quad m=1, \dots, M$$

$$X_{jkmn} \geq 0 \quad j=1, \dots, J \quad (6.5b)$$

$$k=1, \dots, K$$

$$m=1, \dots, M$$

$$n=1, \dots, M$$

The optimal objective value for the reduced problem (6.1b) provides a lower bound ($\underline{Z''}$) on the production costs for the SLCR problem.

Hence the summation of the lower bounds on the objective values of problem (6.1a)-(6.5a) and problem (6.1b)-(6.5b) will give a lower bound on the objective of the SLCR problem, i.e.

$$\underline{Z} = \underline{Z'} + \underline{Z''}$$

6.2 TIGHTNESS OF LOWER BOUND

In this section we give arguments that the tightness of lower bound will be affected significantly by the following parameters:

1. seasonality;
2. setup costs;
3. tightness of capacity.

These conjectures are supported by the computational results.

6.2.1 Effect of Seasonality

The lower bound procedure used to obtain a lower bound on the sum of setup and holding cost for each product ignores capacity constraints. If there is no seasonality in demand the Wagner-Whitin algorithm will tend to result in setups distributed rather uniformly over the planning horizon. It is likely that the lot sizes for this setup combination will be small and hence will not be hard to fit into the available capacity. For problems with low seasonality the lot sizes obtained by the Wagner-Whitin algorithm will be very similar (depending upon the amount of production capacity available) to the lot sizes in the optimal solution. Hence in problems with low seasonality the lower bound on the sum of setup and holding cost will tend to be very close to the sum of setup and holding costs in the optimal solution.

As the seasonality increases the Wagner-Whitin algorithm will tend to schedule large lot sizes, mainly during peak periods. When capacity constraints are imposed, the lot sizes obtained by the Wagner-Whitin procedure may violate some capacity constraints. (As capacity constraints become tighter the number of setups in the optimal solution will increase and the lot sizes will decrease.) The Wagner-Whitin algorithm will result in fewer setups and larger lot sizes than the optimal solution. Hence as seasonality increases the lower bound on the sum of setup and holding cost will tend to be much lower than the sum of setup and holding costs in the optimal solution.

The procedure for lower bound on production cost assumes that all regular time capacity will be used before overtime capacity is used. As seasonality increases the optimal solution may have lot sizes which might necessitate use of overtime capacity in some (peak) periods of demand eventhough there may be some unused regular time capacity in other periods. Hence as seasonality increases the lower bound on production cost will tend to be much lower than the production cost in the optimal solution.

Hence the lower bound will tend to be looser as seasonality increases.

6.2.2 Effect of Setup Cost

When the setup cost for the products is low the Wagner-Whitin algorithm, in order to balance the setup and holding costs, will tend to result in setups in many periods. It is very likely that the lot sizes so obtained will be small enough to fit into available capacity. The lot sizes will be similar (depending upon tightness of capacity) to the lot sizes in the optimal solution. Hence for problems with low setup costs the lower bound on the sum of setup and holding costs will tend to be very close to the sum of setup and holding costs in the optimal solution.

As the setup costs increase the Wagner-Whitin algorithm will result in fewer setups and larger lot sizes. When capacity constraints are imposed the lot sizes obtained by the

procedure may become infeasible. (As capacity constraints become tighter the number of setups in the optimal solution will increase and the lot sizes will decrease.) The lower bound procedure will tend to result in fewer setups and larger lot sizes than the optimal solution. Therefore as setup costs increase the lower bound on the sum of setup and holding costs will tend to be much lower than the sum of setup and holding costs in the optimal solution.

As setup costs increase the number of setups in the optimal solution will decrease. As a result, to make use of expensive setups, overtime capacity might be used in certain periods even though there may be some unused regular time capacity in other periods. Hence as setup costs increase the lower bound on production cost obtained will tend to be much lower than the production cost in the optimal solution.

Hence the lower bound will tend to be looser as setup costs increase.

6.2.3 Effect of Tightness of Capacity

When available regular time capacity is loose (e.g. 120% of the average demand) the lot sizes obtained by Wagner-Whitin algorithm will be more likely to fit into the available capacity. The lot sizes obtained by the lower bound procedure may be near optimal. The lower bound on the sum of setup and holding costs will tend to be very close to the sum of setup and holding costs in the optimal solution.

As capacity constraints become tighter the lot sizes obtained by the Wagner-Whitin algorithm will become increasingly difficult to fit into the available capacity. The optimal solution in these cases will tend to have far more setups and smaller lot sizes compared to the Wagner-Whitin solution. Hence as tightness of capacity increases the lower bound on the sum of setup and holding costs will tend to be increasingly lower than the sum of setup and holding costs.

When the tightness of capacity is 120% of the average demand the procedure for computing lower bound on production cost will result in a value which assumes that all the demands are met by using only regular time capacity. Since regular time capacity available is 120% of the average demand it is likely that all the demands will be met without using any overtime capacity. The lower bound on production cost will tend to be very close to the production cost in the optimal solution.

Hence in case of 120% tightness of capacity the lower bound on total cost will be tight.

When tightness of capacity is 80% the procedure for computing lower bound on production cost will result in a value which assumes that 80% of the demands are met by using regular time capacity and the rest 20% is met by using overtime capacity. Since capacity constraint is very tight it is likely that the optimal solution will result in setups distributed in

many periods and hence will have a production plan which uses all regular time to meet 80% of the demands and overtime to meet the other 20% of the demands. Here again the lower bound on production cost will tend to be very close to the production cost in the optimal solution.

Hence in case of 80% tightness of capacity the lower bound on total cost will be tight but not as tight as in the case of 120% tightness of capacity.

When tightness of capacity is 100% the procedure for computing lower bound on production cost will result in a value which will assume that all the demands are met by using regular time capacity. This may not be true especially for certain types of problems (e.g. for problems with high setup costs and for problems with extreme seasonality the optimal solution might result in larger lot sizes at the expense of overtime). Hence, in these cases, the lower bound on production cost will tend to be much lower than the production cost in the optimal solution.

Hence in case of 100% tightness of capacity the lower bound obtained will be worse than the 80% tightness of capacity case as well as in the case of 120% tightness of capacity.

6.3 COMPARISON OF LOWER BOUND TO THE EXACT SOLUTIONS

Table 2 presents the results of those test problems which were solved optimally by the exact algorithm. As conjectured

Table 2

Comparison of Solution Value to the Lower Bound Value
for problems^a solved optimally
(3 product 12 period cases).

Sea.	Set.	Cap.	Opt. Soln. Value	Lower Bound Value	Difference	% Difference
No	Med	100%	536448	533028	3402	.50
No	Med	80%	585492	583188	2304	.60
Mod	Low	100%	518147	514136	4011	.80
Mod	Low	80%	560080	560016	64	.02
Mod	Med	80%	587144	584508	2636	.55

^aAll problems with low setup and no seasonality were solved optimally. In these cases the heuristic solution and the lower bound were equal to the optimal solution.

the lower bound seems to be significantly affected by problem parameters like seasonality of demand, setup costs and tightness of capacity.

All problems with low setup costs and no seasonality were solved exactly. As expected the lower bound for problems with low setup cost and no seasonality was equal to the optimal solution value.

For problems with low setup cost and moderate seasonality the difference between the lower bound and the optimal solution ranged from 0.02% to 0.80%.

For problems with medium setup and no seasonality the difference between the lower bound and the optimal solution ranged from 0.50% to 0.60%. Finally one problem with medium setup costs and moderate seasonality was solved optimally. The difference between the lower bound and the optimal solution was 0.55%.

From the results of problems solved exactly it can be seen that for problems with low seasonality, and low setup cost the lower bound obtained was tight. The lower bound obtained for problems with moderate seasonality was looser than lower bound for problems with no seasonality. Similarly the lower bound for problems with medium setup costs was looser than the lower bound for problems with low setup costs. Finally the lower bound for problems with 80% tightness of capacity was as tight or tighter (in most cases) compared to the problems with 100% tightness of capacity.

Since the exact solution algorithm could not solve any problems with high setup costs and problems with extreme seasonality it was not possible to say how loose the lower bound will be in these cases. Based on the discussion on tightness of the lower bound it appears that for problems with high setup costs and problems with high seasonality the lower bound will tend to be loose.

CHAPTER VII

COMPUTATIONAL EXPERIMENTS

This chapter presents the design of computational experiments and the summary of the results of the computational experiments. The summary of results includes analysis of the performance of the heuristic by comparing the heuristic solutions with the optimal solutions for all problems solved exactly and also by comparing the heuristic solutions with the lower bound for all test problems.

7.1 DESIGN OF COMPUTATIONAL EXPERIMENTS

The computational study consists of the solution of 243 test problems. The test problems differ with respect to (1) the number of products, (2) the number of time periods, (3) the seasonality of demand, (4) the available regular time capacity (as a percentage of average demand), and (5) the setup. Each of these five problem parameters were varied over three values to generate $3^5 = 243$ different parameter combinations on which the problems were based. The test design is similar to that of Graves (1982), although the problem formulated in this research is different from the problem formulated by Graves (1982). (One major difference is that Graves formulation assumes unlimited overtime capacity as opposed to constrained overtime capacity.) The test problems in this research range in size from 3 product and 12 period

problem to 9 product and 36 period problem.

The number of products for any problem can be 3, 6 or 9. The number of time periods can be 12, 24, or 36. The product demands for any problem can either have no seasonality, moderate seasonality or extreme seasonality. The setup cost for each product can assume three values-- low, medium and high. The level of regular time capacity can assume three values such that it can cover 80%, 100% or 120% of total demand over the planning horizon.

For each problem the amount of regular time available is constant across the periods over the planning horizon. For convenience the initial inventories and the ending inventories for all products is constrained to zero.

7.1.1 Specifications of test problems (3 product and 12 period case).

Let q index problem set, where $q = 1$ indexes problem set with no seasonality, $q = 2$ indexes problem set with moderate seasonality and $q = 3$ indexes problem set with extreme seasonality.

Product Demands

For problem set q , demand for product j in period n is given by,

$$D_{jn}^q = \sum_{p=1}^P r_p * b_{jn}^q$$

where b_{jn}^q is a multiplicative seasonality factor (see Figure 4) for product j in period n belonging to problem set q , and r_p is the p^{th} random draw from a uniform distribution over the range $(\underline{d}_p, \overline{d}_p)$ (see Figure 5). P is equal to five for products 1 and 2 and P is equal to ten for product 3.

Setup Costs

For each product j the setup cost S_j is given by,

$$S_j = \sum_{p=1}^P t_p$$

where t_p is the p^{th} random draw from a uniform distribution over the range $(\underline{s}_p, \overline{s}_p)$ (see Figure 5). S_j is the "medium" setup cost for product j for all time periods for all test problems. To get the "low" setup cost we divide S_j by five, while to get the "high" setup cost we multiply S_j by five.

Production costs

The regular time cost is 40.00 per unit and overtime cost is 60.00 per unit for all periods for all problems.

Holding costs

The inventory holding cost H_j is 4.00 for product 1, 7.00 for product 2 and 6.00 for product 3 for all time periods for all problems.

PRODUCT #1

$\frac{d_p}{d_p}$	20	40	15	25	80
	-----	-----	-----	-----	-----
	40	60	25	65	120

$\frac{s_p}{s_p}$	50	50	50	50	50
	-----	-----	-----	-----	-----
	150	150	150	150	150

PRODUCT #2

$\frac{d_p}{d_p}$	80	15	40	40	55
	-----	-----	-----	-----	-----
	120	25	60	60	85

$\frac{s_p}{s_p}$	100	100	100	100	100
	-----	-----	-----	-----	-----
	200	200	200	200	200

PRODUCT #3

$\frac{d_p}{d_p}$	20	20	30	30	20	30	30	50	60	40
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
	40	40	50	50	40	70	50	100	80	80

$\frac{s_p}{s_p}$	20	20	30	30	20	30	30	50	60	40
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
	40	40	50	50	40	70	50	100	80	80

Figure 5. Ranges of Demands and Setup Costs for Products.

Available regular time

The available regular time in period m for problem set q is given by,

$$C_{1m}^q = \sum_{j=1}^J \sum_{n=1}^M D_{jn}^q / M$$

and overtime in period m for problem set q is given by,

$$C_{2m}^q = 1.5 * C_{1m}^q.$$

We multiply C_{1m}^q by .80 for problems with 80% tightness of capacity, by 1.00 for problems with 100% tightness of capacity and by 1.2 for 120% tightness of capacity.

The same problem generating algorithm is used for the 6 product and 9 product problems. The 6 product problem is treated as two 3 product problems. Similarly the 9 product problem is treated as three 3 product problems.

The setup costs for the 6 product and 9 product problem is obtained by replicating the setup costs for the 3 product problem. Similarly the holding costs for the 6 product problem and 9 product problem is obtained by replicating the holding costs for the 3 product problem. Production costs for the 6 product and the 9 product problem are same as the production costs for the 3 product problem.

The demands for the 6 product problem is obtained by replicating the demands for the 3 product problem. Similarly the demands for the 9 product problem is also obtained by replicating the demands for the 3 product problem. The capacities are computed exactly like the 3 product problem.

The 24 period problem is treated as two 12 period problems. The demands for the 24 period problem is obtained by replicating the 12 period problem demands. Similarly the 36 period problem is treated as three 12 period problems. The capacities are computed exactly like the 12 period problem.

7.2 SUMMARY OF RESULTS

The test problems were solved on the IBM 4381 system. The computation time for our procedure ranged from .94 c.p.u. seconds to 189.01 c.p.u. seconds with the average time per problem being approximately 30 c.p.u. seconds. We discuss the performance of our heuristic with respect to (1) seasonality in demand, (2) setup costs, (3) tightness of capacity, (4) number of products, and (5) number of periods.

To analyze the performance of the heuristic the heuristic solution value is compared with the following:

1. optimal solution value for the problems solved exactly;
2. the lower bound value for all problems.

7.2.1 Comparison of the heuristic solution with the exact solution

All problems with low setup costs and no seasonality were solved optimally by the exact algorithm. Two problems with low setup costs and moderate seasonality were solved optimally. Two problems with medium setup costs and no seasonality were solved optimally. Finally one problem with medium setup costs

and moderate seasonality was solved optimally. The comparison of the heuristic solution value with the optimal solution for those problems solved optimally is shown in Table 3.

For all problems that were solved exactly it can be seen that the objective function value of the heuristic solutions exceeded the optimal value by an amount that is less than the amount by which the optimal value exceeded the lower bound. That is, the heuristic solution deviated from optimality by less than half of the amount by which it exceeded the lower bound.

For all problems with low setup costs and no seasonality the heuristic solution for these problems were the same as the optimal solution. For problems with low setup costs and moderate seasonality the difference between the heuristic solution and the optimal solution was very close to 0.00%.

For the problems with moderate setup costs and no seasonality solved optimally the difference between the heuristic solution and the optimal solution ranged from 0.00% to 0.18%. Finally for one problem with medium setup costs and moderate seasonality solved optimally the difference between the heuristic solution and the optimal solution was 0.30%.

Since the problems with high setup costs or high seasonality could not be solved exactly it was not possible to determine the deviation from optimality of the heuristic solution.

Table 3

Comparison of Solution Value to the Heuristic Solution
Value for problems^a solved optimally
(3 product 12 period cases).

Sea.	Set.	Cap.	Opt. Soln. Value	Heur. Soln. Value	Difference	% Difference
No	Med	100%	536448	536448	0	0.00
No	Med	80%	585492	586608	1116	0.18
Mod	Low	100%	518147	518264	117	0.02
Mod	Low	80%	560080	560112	32	0.00
Mod	Med	80%	587144	588932	1788	0.30

^aAll problems with low setup and no seasonality were solved optimally. In these cases the heuristic solution and the lower bound were equal to the optimal solution.

7.2.2 Comparison of the heuristic solution with lower bound

To analyze the performance of the heuristic the percentage difference between the heuristic solution and the lower bound (discussed in Chapter VI) for the SLCR problem is computed. The percentages are the difference between the heuristic solution value and the lower bound value divided by the heuristic solution value.

The difference between the heuristic solution and the lower bound could be due to either the performance of the heuristic or looseness of the lower bound. (As was pointed out earlier, in those problems for which the optimal solution was known, less than half of this difference was attributed to the difference between the heuristic solution and the optimal solution.)

The difference between heuristic solution and the optimal solution and lower bound and the optimal solution for the problems which were solved optimally by the exact algorithm is shown in Table 4. The difference between heuristic solution and the lower bound ranged from 0.00% to 0.66%. In cases where the difference was greater than 0.00% the difference was mainly due to looseness of the lower bound rather than performance of the heuristic. The looseness of the lower bound accounted for 0.0% to 100% of the difference in those problems which were solved exactly.

Table 4

Comparison of Heuristic Solution Value and the Lower Bound Value for problems^a solved optimally
(3 product 12 period cases).

Sea.	Set.	Cap.	Diff. Betn. Heur. and Lower Bound	Diff. Betn. Heur. and Opt. Soln.	Diff. Betn. Lower Bound Opt. Soln.
No	Med	100%	3420	0	3420
No	Med	80%	3420	1116	2304
Mod	Low	100%	4128	117	4011
Mod	Low	80%	96	32	64
Mod	Med	80%	4424	1788	2636

^aAll problems with low setup and no seasonality were solved optimally. In these cases the heuristic solution and the lower bound were equal to the optimal solution.

The average percentage difference (a.p.d.) between the heuristic solution value and the lower bound is computed for each problem parameter. The a.p.d. across all 243 problems was 4.12%. The percentage difference ranged from 0.00% to 14.67%.

The most significant effect on the difference between the heuristic solution and the lower bound was due to the seasonality parameter. The second most significant effect on the difference between the heuristic solution and the lower bound was due to the setup cost parameter. The effect of the number of products, number of periods and the tightness of capacity on the difference between the heuristic solution and the lower bound seems to be insignificant.

The a.p.d. increased as seasonality of the problem increased (see Figure 6). The a.p.d. also increased as the setup costs increased (see Figure 7). The a.p.d. was the lowest (3.12%) for 120 % tightness of capacity, was second lowest (4.01%) for 80% capacity and the highest (5.21%) for 100% capacity (see Figure 8). The number of products did not seem to have a significant effect on the a.p.d. (see Figure 9). Finally the number of periods did not seem to have significant effect on the a.p.d. (see Figure 10).

The effect of the interaction between seasonality parameter and the setup cost parameter on the a.p.d. is shown in Figure 11. The results show that the effect of interaction

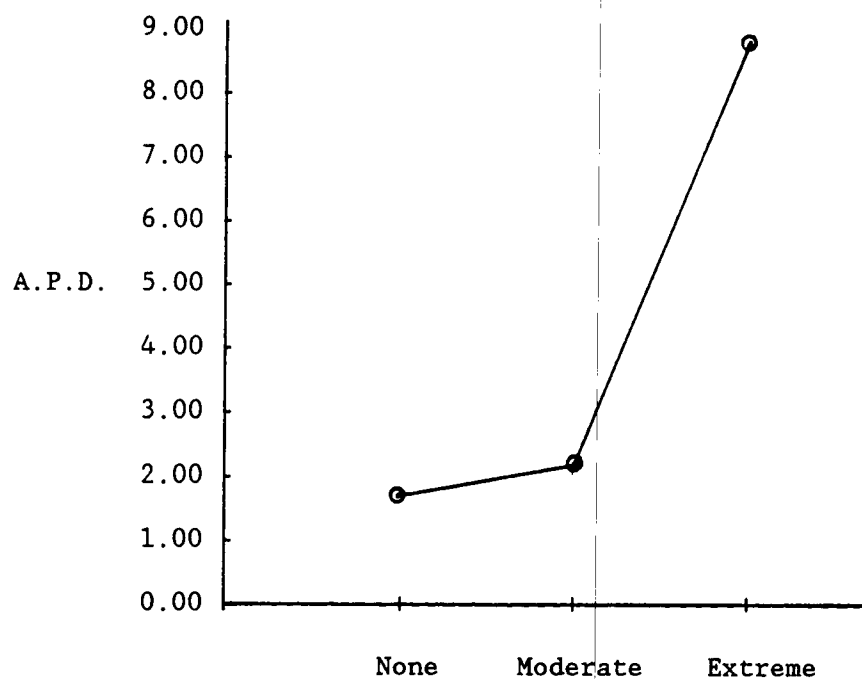


Figure 6. The Effect of Seasonality on Average Percent Difference (A.P.D.).

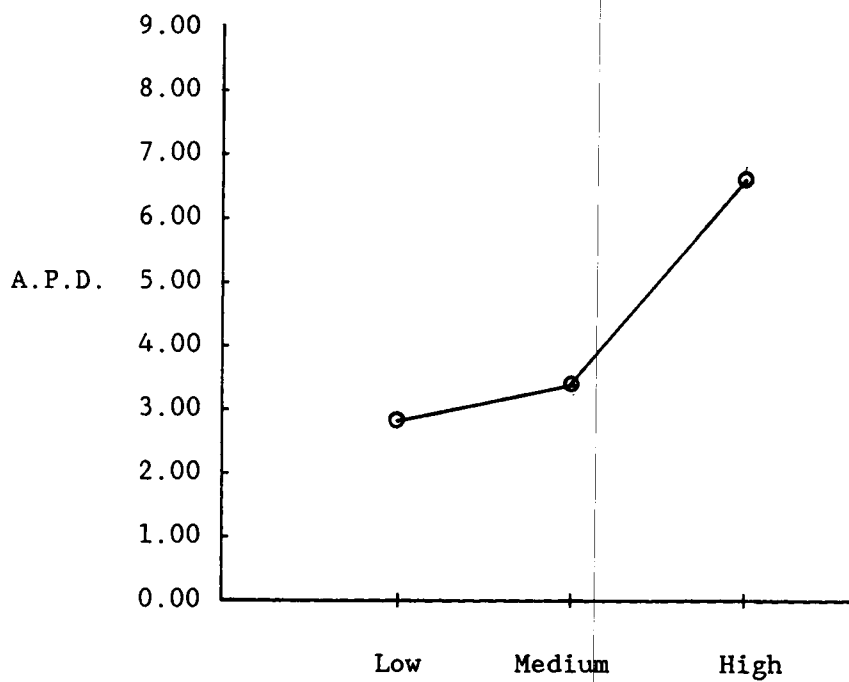


Figure 7. The Effect of Setup Costs on Average Percent Difference (A.P.D.).

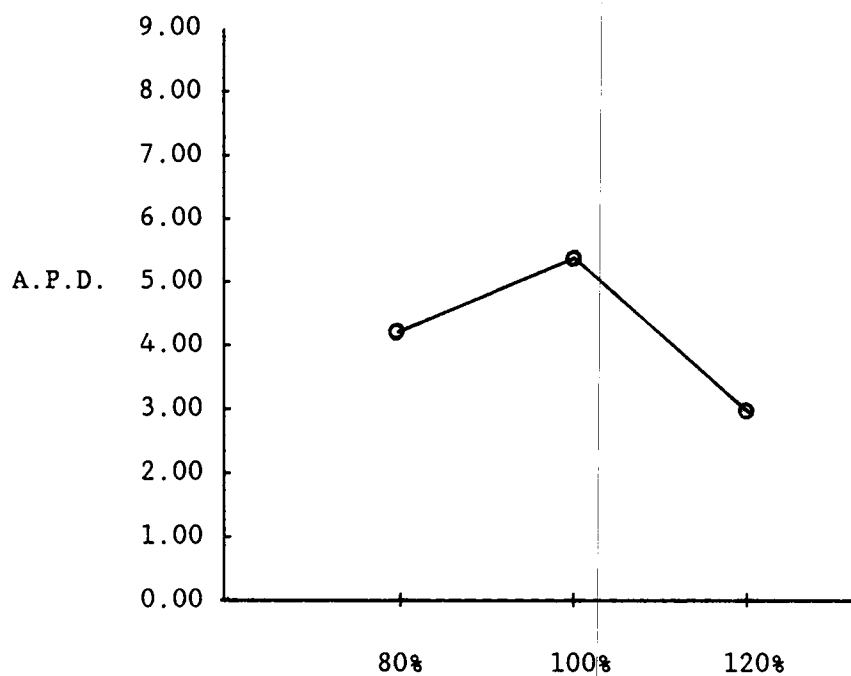


Figure 8. The Effect of Tightness of Capacity on Average Percent Difference (A.P.D.).

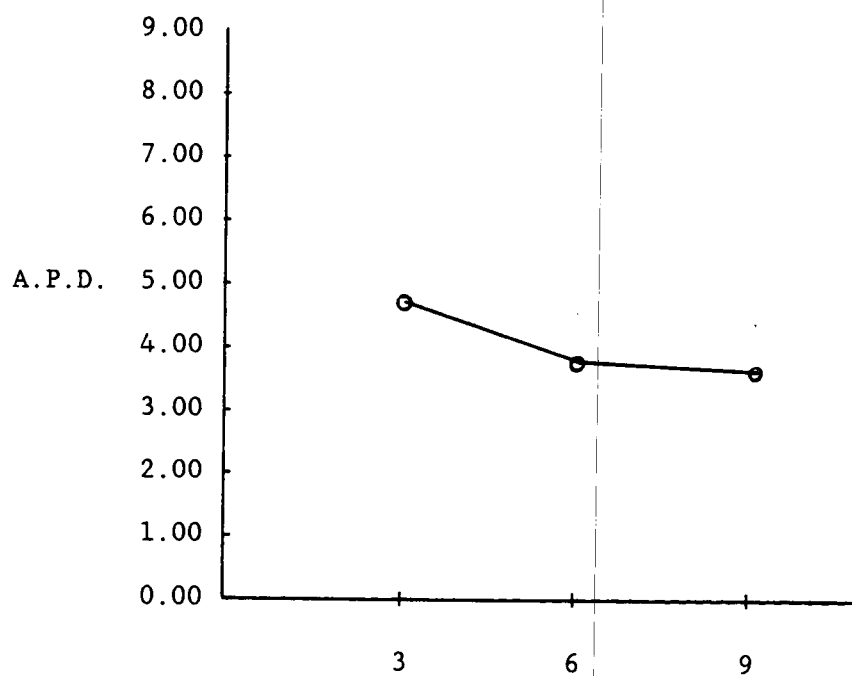


Figure 9. The Effect of Number of Products on Average Percent Difference (A.P.D.).

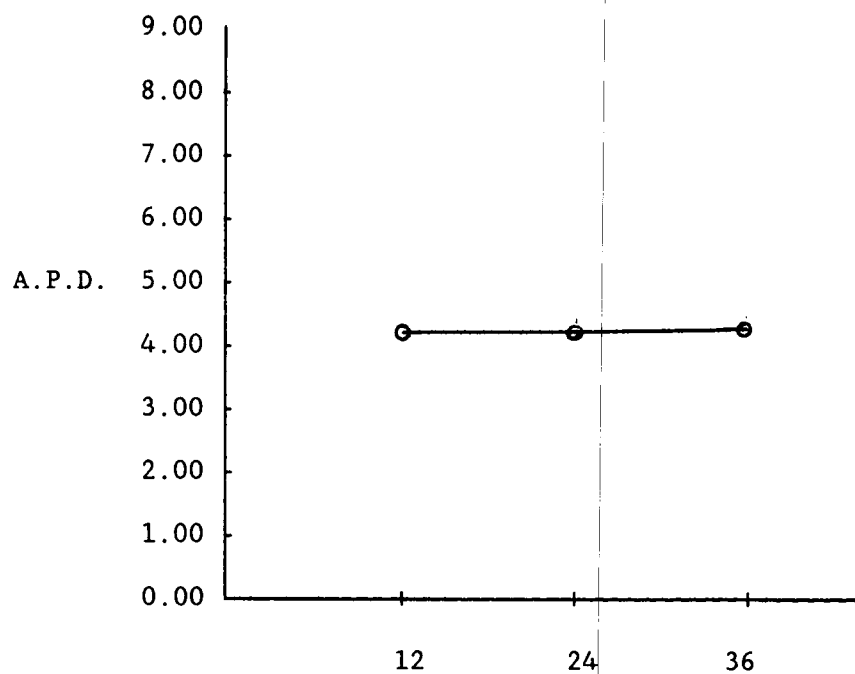


Figure 10. The Effect of Number of Periods on Average Percent Difference (A.P.D.).

Setup Cost	High	4.40	4.76	10.49
	Medium	0.46	0.84	7.83
	Low	0.00	0.39	8.00
		None	Moderate	Extreme
		Seasonality		

Figure 11. The Effect of Interaction of Setup Costs and Seasonality on Average Percent Percent Difference (A.P.D.).

between seasonality parameter and setup cost parameter on the a.p.d. is not significant. The a.p.d. was the lowest for the problems with low setup costs and low seasonality, and it was the highest for the problems with high setup costs and extreme seasonality.

For problems with low seasonality, the a.p.d. increased as setup costs increased. The same was true for problems with moderate seasonality and problems with extreme seasonality. Similarly, for problems with low setup costs, the a.p.d. increased as seasonality in product demands increased. The results were consistent for problems with medium setup costs and problems with high setup costs.

The a.p.d. for problems with no seasonality and moderate seasonality is very small. Similarly for problems with low setup costs and medium setup costs the a.p.d. is very small. The a.p.d. increased considerably for problems with extreme seasonality for problems with high setup costs.

CHAPTER VIII

TOPICS FOR FURTHER RESEARCH

A useful extension to this dissertation would be to develop other efficient exact solution algorithms. Different orders of enumeration in the implicit enumeration algorithm could be explored to see if the solution procedure could be made more efficient.

The heuristic developed in this dissertation could be compared with existing heuristics in solving "traditional" fixed charge transportation problems. Also research can be done to identify other practical problems in business which have structure similar to the problem discussed in this research.

Another useful extension to this research would be to explore other techniques for computing a "better" lower bound. This would help in doing a better evaluation of the heuristic developed in this dissertation and other heuristics used in solving similar class of problems.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Aras, Omar A. and Swanson, L. G., "A Lot Sizing and Sequencing Algorithm for Dynamic Demands upon a Single Facility," Journal of Operations Management, 2 3, 177-185 (May 1982).
- Bahl, H. C., "Column Generation Based Heuristic for Multi-Item Scheduling," AIIE Transactions, 15 2, 136-141 (June 1983).
- Bahl, H. C. and Ritzman, L. P., "A Cyclical Scheduling Heuristic for Lot Sizing with Capacity Constraints," International Journal of Production Research, 22 5, 791-800 (1984).
- Balinski, M. L., "Fixed Cost Transportation Problem," Naval Research Logistics Quarterly, 8 1, 41-54 (1961).
- Barany, I., Van Roy, T. J. and Wolsey, L. A., "Strong Formulations for Multi-Item Capacitated Lot Sizing," Management Science, 30 10, 1255-1261 (October 1984).
- Barr, R. S., Glover, F. and Klingman, D., "A New Optimization Method for Large Scale Fixed Charge Transportation Problems," Operations Research, 29 3, 448-463 (May-June 1981).
- Berry, W. L., "Lot Sizing Procedures for Requirements Planning Systems: A Framework for Analysis," Production and Inventory Management, 13 2, 19-34 (1972).
- Billington, P. J., McClain, J. O., and Thomas, L. J., "Mathematical Programming Approaches to Capacity Constrained MRP Systems: Review, Formulation, and Problem Reduction," Management Science, 29 10, 1126-1141 (October 1983).
- Billington, P. J., McClain, J. O., Thomas, L. J. and Weiss, E. N., "MRP Lotsizing with a Bottleneck Work Center," Working Paper. Graduate School of Business and Public Administration, Cornell, (1984).
- Bitran, G. R. and Yanasse, H. H., "Computational Complexity of the Capacitated Lot Sized Problem," Management Science, 28 10, 1174-1186 (October 1982).
- Blackburn, J. D. and Millen, R. A., "Heuristic Lot-Sizing Performance in a Rolling Schedule Environment," Decision Sciences, 11 4, (October 1980).
- Blackburn, J. D. and Millen, R. A., "Improved Heuristics for Multi-Stage Requirements Planning Systems," Management Science, 28 1, 44-56 (January 1982).

- Bradley, G. H., Brown, G. G., and Graves, G. W., "Design and Implementation of Large Scale Primal Transshipment Algorithms," Management Science, 24 1, 1-33 (September 1977).
- Bradley, G. H., Brown, G. G., and Graves, G. W., "GNET, A Primal Capacitated Network Program," Copyright 1975, 1977.
- Buffa, E. S. and Miller, J. G., Production-Inventory Systems: Planning and Control, Richard D. Irwin, Homewood, Illinois (1979).
- Cabot, A. V. and Erenguc, S. S., "Some Branch-and-Bound Procedures for Fixed-Cost Transportation Problems," Naval Research Logistics Quarterly, 31, 145-154 (1984).
- Cooper, L. and Drebes, C., "An Approximate Solution for the Fixed Charge Problem," Naval Research Logistics Quarterly, 14 1, 101-113 (1967).
- Daniels, V. L., "Lot Sizing and Sequencing in Process Industries," unpublished dissertation, Ohio State University, (1983).
- Dantzig, G. B. and Wolfe, P., "A Decomposition Algorithm for Linear Programs," Econometrica, 29, 767-778 (1961)
- Denzler, D. R., "An Approximate Algorithm for the Fixed Charge Problem", Naval Research Logistics Quarterly, 16 3, 411-416 (1969).
- Dixon, P. S. and E. A. Silver, "A Heuristic Solution Procedure for the Multi-Item, Single-Level, Limited Capacity, Lot-Sizing Problem," Journal of Operations Management, 2 1, 23-29 (October 1981).
- Dogramaci, A., Panayiotopoulos J. E. and Adam, N. R., "The Dynamic Lot Sizing Problem for Multiple Items Under Limited Capacity," AIIE Transactions, 13 4, 294-326 (December 1981).
- Driebeek, N. J., "An Algorithm for the Solution of Mixed Integer Programming Problems," Management Science, 12, 576-587 (1966).
- Dzielinski, B. P., Baker, C. T. and Manne, A. S., "Simulation Tests of Lot Size Programming," Management Science, 9 2, 229-258 (January 1963).
- Dzielinski, B. P. and Gomory, C. T., "Optimal Programming of Lot Sizes, Inventory and Labor Allocations," Management Science, 2 9, 874-890 (July 1965).

- Eisenhut, P. S., "A Dynamic Lot-Sizing Algorithm with Capacity Constraints," AIIE Transactions, 7 2, 170-176 (June 1975).
- Elmaghraby, S. E., "The Economic Lot Scheduling Problem (ELSP): Review and Extension," Management Science, 24 6, 587-598 (1978).
- Fisk, J., "An Initial Bounding Procedure for use with 0-1 Single Knapsack Algorithms," Operations Research, 14, 88-98 (1977).
- Fisk, J. and McKeown, P., "The Pure Fixed Charge Transportation Problem," Naval Research Logistics Quarterly, 26, 631-641 (1979).
- Florian, M., Klein, M., "Deterministic Production Planning with Concave Costs and Capacity Constraints," Management Science, 18 1, 12-20 (September 1971).
- Florian, M., Lenstra, J. K. and Rinnooy Kan, A. H. G., "Deterministic Production Planning: Algorithms and Complexity," Management Science, 26 7, 12-20 (July 1980).
- Gabbay, H., "Multi-Stage Production Planning," Management Science, 25 11, 1138-1148 (November 1979).
- Garey, M. R., and Johnson, D. S., Computations and Tractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, California, (1979).
- Garfinkel, R. S., and Nemhauser, G. L., Integer Programming, John Wiley and Sons, New York, (1972).
- Graves, S. C., "Multi-Stage Lot Sizing: An Iterative Procedure," in Schwartz, L. B., (ed.), Multi-Level Production/Inventory Control System: Theory and Practice, North-Holland, 95-109 (1981).
- Graves, S. C., "Using Lagrangean Techniques to Solve Hierarchical Production Planning Problems," Management Science, 28 3, 260-275 (March 1982).
- Gray, P., "Exact Solution of the Fixed-Charge Transportation Problem," Operations Research, 19 6, 1629-1538 (August 22 1968).
- Haehling von Lanzenhauer, C., "Production and Employment Scheduling in Multi-Stage Production Systems," Naval Research Logistics Quarterly, 17 2, 193-198 (June 1970).

- Haessler, R. W., "An Improved Extended Basic Period Procedure for Solving the Economic Lot Scheduling Problem," AIIE Transactions, 2, 336-340 (1979).
- Hirsch, W. M., and Dantzig, G. B., "The Fixed Charge Problem," Naval Research Logistics Quarterly, 15 3, 413-424 (1968).
- Karni, R., "Maximum Part-Period Gain: A Lot-Sizing Procedure for Unconstrained and Constrained Requirement Planning Systems," Production and Inventory Management, 22 2 (1981).
- Karni, R. and Roll, Y., "A Heuristic Algorithm for the Multi-Item Lot-Sizing Problem with Capacity Constraints," AIIE Transactions, 14 4, 249-256 (1982).
- Kennington, J., "The Fixed-Charge Transportation Problem: A Computational Study with a Branch-and-Bound Code," AIIE Transactions, 8 2, 241-247 (1976).
- Kennington, J. and Unger, E., "A New Branch-and-Bound Algorithm for the Fixed-Charge Transportation Problem," Management Science, 22 10, 1116-1126 (1976).
- Kuhn, H. W. and Baumol, W. J., "An Approximate Algorithm for the Fixed-Charges Transportation Problem," Naval Research Logistics Quarterly, 9 1, 1-15 (July 1961).
- Lambrecht, M. R. and Vander Eecken, J., "A Facilities in Series Capacity Constrained Dynamic Lot-Sized Model," European Journal of Operational Research, 2 1, 42-49 (1978a).
- Lambrecht, M. R. and Vander Eecken, J., "Capacity Constrained Single-Facility Dynamic Lot-Sizing Model," European Journal of Operational Research, 2 2, 132-136 (1978b).
- Lambrecht, M. R. and Vanderveken, H., "Heuristic Procedures for the Single Operation, Multi-Item Loading Problem," AIIE Transactions, 11 4, 319-326 (December 1979).
- Lasdon, L. S. and Terjung, R. C., "An Efficient Algorithm for Multi-Item Scheduling," Operations Research, 19, 892-914 (1967).
- Manne, A. S., "Programming of Economic Lot Sizes," Management Science, 4 2, 115-135 (January 1958).
- McKeown, P. G., "A Branch-and-Bound Algorithm for Solving Fixed Charge Problems," Naval Research Logistics Quarterly, 28 4, 607-617 (December 1981).

- Meal, H. C. and Silver, E. A., "A Simple Modification of the EOQ for the Case of a Varying Demand Rate," Production and Inventory Management, (Fourth Quarter 1969).
- Murty, K. G., "Solving the Fixed Charge Problem by Ranking Extreme Points," Operations Research, 16 2, 268-279 (March-April 1968).
- Newson, E. F. P., "Multi-Item Lot Size Scheduling by Heuristic-Part 1: with Fixed Resources and Part 2: with Variable Resources," Management Science, 21 10, 1186-1203 (June 1975).
- Posner M. E. and Szwarc, W., "A Transportation Type Aggregate Production Model with Backordering," Management Science, 29 2, 188-199 (February 1983).
- Ross, G. J. and Soland, R., "A Branch and Bound Algorithm for the Generalized Assignment Problem," Mathematical Programming, 891-903 (1975).
- Schweitzer, P. J. and Silver, E. A., "Mathematical Pitfalls in the One Machine Multiproduct Economic Lot Scheduling Problem," Operations Research, 31 2, 401-405 (March-April 1983).
- Silver, E. A. and Meal, H. C., "A Heuristic for Selecting Lot Size Quantities for the Case of a Deterministic Time Varying Rate and Discrete Opportunities for Replenishment," Production and Inventory Management, 14 2, 64-74 (1973).
- Steinberg, D. I., "The Fixed Charge Problem," Naval Research Logistics Quarterly, 17 2, 217-235 (1970).
- Steinberg, E. and Napier, H. A., "Optimal Multi-Level Lot Sizing for Requirement Planning Systems," Management Science, 26 12, 1258-1271 (1980).
- Thizy, J. M. and Van Wassenhove, L. N., "Decomposition Algorithms for the Multi-Product Lot-Sizing Problem with Capacity Constraints," Report 82-15, Dept. of Industrial Engineering, Katholieke Universiteit Leuven, Belgium, (1982).
- Thizy, J. M. and Van Wassenhove, L. N., "A Subgradient Algorithm for the Multi-Item Capacitated Lot Sizing problem," AIIE Transactions, 17 4, 308-313 (December 1985).

- Tomlin, J. A., "An Improved Branch-and-Bound Method for Integer Programming," Operations Research, 19 4, 1070-1075 (November-December 1971).
- Trotter, L. E., and Shetty, C. M., "An Algorithm for the Bounded Variable Integer Programming Problem," Journal of the Association for Computing Machinery, 21 3, 505-513 (July 1974).
- Wagner, H. M., and Whitin, T. M., "Dynamic Version of the Economic Lot Size Model," Management Science, 5 1, 89-96 (October 1958).
- Walker, W. E., "A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem," Management Science, 22 5, 587-596 (January 1976).
- Zahorik, A., Thomas, L. J. and Trigeiro, W. W., "Network Programming Models for Production Scheduling in Multi-Stage Multi-Item Capacitated Systems," Management Science, 30 3, 308-325 (1984).
- Zangwill, W. I., "A Deterministic Multi-Product, MultiFacility, Production and Inventory Model," Operations Research, 14 3, 486-507 (May-June 1966).
- Zangwill, W. I., "A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot-Size Production System - A Network Approach," Management Science, 15 9, 506-597 (May 1969).
- Zangwill, W. I., "Minimum Concave Cost Flows in Certain Networks," Management Science, 14 7, 429-450 (March 1968).

VITA

The author was born Manohar S. Madan in Madras, India, on May 22, 1955.

He earned a Bachelor of Engineering degree in Mechanical Engineering from M.S. University, India, in June 1977. He came to the United States in August 1978. He earned a Master's degree in Engineering Management from the University of Detroit, Detroit, in December 1979. He will receive the Doctor of Philosophy degree at the University of Tennessee, Knoxville, in 1987 with a major in Business Administration.

He has been working as an Assistant Professor in the Information Systems and Operations Management Department at the University of Toledo since January 1985.