LOWER BOUNDS FOR PRODUCTION/INVENTORY PROBLEMS BY COST ALLOCATION

By

PAUL OMOLEWA IYOGUN

B.Sc. (Petroleum Engineering), University of Ibadan, 1978
M.Eng. (Industrial Engineering), University of Toronto, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
(Commerce and Business Administration)

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July 1987

4 6

Department of Commerce and Business Administration

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date 6th August, 1987

ABSTRACT

This thesis presents a cost allocation method for deriving lower bounds on costs of feasible policies for a class of production/inventory problems. Consider the joint replenishment problem where a group of items is replenished together or individually. A sequence of reorders for any particular item will incur holding, backorder and set-up costs specific to the item. In addition whenever *any* item is replenished a joint cost is incurred. What is required of the total problem is the minimization of a cost function of the replenishment sequence or policy.

The cost allocation method consists of decomposing the total problem into subproblems, one for each item, by allocating the joint cost amongst the items in such a way that every item in the group receives a positive allocation or none. The result is that, for an arbitrary feasible cost allocation, the sum of the minimum costs for the subproblems is a lower bound on the cost of any feasible policy to the total problem. The results for the joint replenishment problem follows:

For the constant and continuous demand case we reproduce the lower bound of Jackson, Maxwell and Muckstadt more easily than they did. For the multi-item dynamic lot-size problem, we generalize Silver–Meal and part-period balancing heuristics, and derive a cost allocation bound with little extra work. For the 'can-order' system, we use periodic policies derived from the cost allocation method and show that they are superior to the more complex (s, c, S) policies. The cost allocation method is easily generalized to pure distribution problems where joint replenishment decisions are taken at several facilities. For example, for the one-warehouse multi-retailer problem, we reproduce Roundy's bound more easily than he did. For the multi-facility joint replenishment problem (a pure distribution system with an arbitrary number of warehouses), we give a lower bound algorithm whose complexity is $dr \log r$ where d is the maximum number of facilities which replenish a particular item and r is the number of items.

Contents

List of Tables

List of Figures

This thesis is dedicated to my mother, Anima Ebe Iyogun.

ACKNOWLEDGEMENT

I would like to thank Professor Derek Atkins, my research supervisor, for introducing me to this area of research, for supporting me both morally and financially and for guiding me throughout the period of this research. His help was invaluable in preparing this document. My thanks also go to Professors Shelby Brumelle and Daniel Granot, members of my research committee, and Maurice Queyranne who were very helpful in correcting my mistakes and for their moral support. I would also like to thank Professors Martin Puterman and Piet de Jong for giving me an opportunity to earn some money when I needed it most, and the Natural Science and Engineering Research Council of Canada and the Faculty of Commerce for their financial support throughout the period of this research.

Finally I must thank my wife Amanele, and my sons Onose and Odalo for their patience and understanding when I could not be with them most of the time because of this research.

INTRODUCTION

This dissertation presents a cost allocation method for deriving lower bounds on costs of feasible policies to joint replenishment problems in inventory management. Consider a group of items whose replenishment generally requires a common process such as having a common supplier or a common production unit, or a common mode of transportation. Replenishment of one or more items in the group involves a major set-up cost resulting from initiating the order or production process. In addition to the major set-up cost, each item to be included in the replenishment group has an associated cost which depends only on the item. This item-dependent cost may include a set-up cost, to be referred to as the minor set-up cost, and a purchase cost for the item.

There is also an inventory related cost rate which depends on the amount of stock and the demand for each item. The particular nature of this cost depends on the context but in general will involve the cost of holding excess inventory in stock when it is not needed or a penalty cost for not having enough stock to meet the immediate demands.

The demand process can be either deterministic (constant or time varying, discrete or continuous) or stochastic. Each of these processes gives rise to a complex problem in joint replenishment inventory management. The methods of solution are quite different, and will therefore be treated in separate sections.

The problem is to schedule replenishment of each item so as to minimize the total set-up and inventory related costs or the average of these costs per unit time over a known horizon. One obvious solution to this problem is to treat each item separately and schedule replenishments using the sum of the item-dependent set-up cost and the major set-up cost as the set-up cost for each item. The solution obtained by this method is feasible and hence its cost is an upper bound on all reasonable solutions. This upper bound is found to be very loose in most practical problems, indicating that there is ample room for improved solutions. Unfortunately, the computational requirements for obtaining an optimum solution to a joint replenishment problem with only a few items are extraordinarily heavy. Moreover the form of the optimum policy, if known, may be very complex and therefore not implementable in practice. These two issues have forced researchers to seek simple solutions to the joint replenishment problem. A solution will be referred to as easy or simple when the parameters specifying the replenishment schedule are easy to compute and the policy induced by these parameters is easy to implement. If also the easy solution results in a substantial cost saving when compared to the upper bound, the solution is "good". Efforts are made to balance computational simplicity and solution quality.

The greater part of this dissertation will discuss simple techniques that give very good solutions - generally better than 105% of a lower bound.

The major motivation for this research is a lower bound for a general class of inventory problems, which has a particularly simple form for the joint replenishment problem. Its derivation also gives some useful insight into possible feasible policies which turn out to be very good.

The dissertation is organized into six chapters. In Chapter one, we survey the literature on the joint replenishment problem and discuss some existing solution techniques. In Chapter two, we introduce a new method referred to as the *cost allocation* method of deriving lower bounds for a class of inventory/ production problems which require coordinating the replenishment of more than one item. This cost allocation method is used to work out in detail the lower bounds for the joint replenishment problem and the one-warehouse, multi-retailer problem both under deterministic, constant and continuous demands. Other problems to which the cost allocation method can be applied are also discussed. In Chapter three we introduce the multi-facility joint replenishment problem which is a pure distribution problem requiring joint replenishment decisions at various facilities. Although the discussion is long, the algorithm for determining the lower bound for this problem is very simple. Again the form of the lower bound is obtained from the cost allocation method. In Chapter four, we discuss the multi-item dynamic lot size problem. The cost allocation method provides a simple lower bound for this problem. This lower bound gives a new insight into the problem which helps in generalizing some of the single item dynamic lot sizing heuristics to the multi-item environment, without losing the quality of the heuristics. For example a series of computational results shows that the generalized Silver-Meal heuristic performs equally well in the multi-item environment as the original heuristic in the single product environment. It is also demonstrated that the effectiveness ratios of the part period balancing heuristic and a variant of it are unchanged when generalized to the multi-item environment. These generalized heuristics can be applied quite easily to the joint replenishment problem with an arbitrary number of products. This is in contrast to the existing heuristics which, because of their complexity, could not be used for problems with more than a few items. In Chapter five, we demonstrate through computational examples that the simple periodic policy obtained using a method derived from the cost allocation bound, performs better than the much written-about s, c, S policy. Finally, Chapter six is a summary and a discussion of topics for further research.

In summary, this thesis presents a new and very general method - the cost allocation method, and demonstrates its use in determining lower bounds; which in turn can be used to develop new solutions to a wide class of production/inventory problems that require coordinating the replenishment of more than one item.

CHAPTER ONE

The Joint Replenishment Problem: Literature Review

Introduction

Consider an inventory control manager faced with the problem of scheduling the production or procurement of N different items over a known horizon, so that the demands for each product or item over the horizon are met, subject to constraints on backlogging. If there is no restriction on the total amount that can be backlogged at any time, any procurement or production policy that satisfies all demands over the horizon is feasible. Whatever constraints there are on the problem, it is likely that there are an infinite number of, or very many feasible solutions to the problem. The problem is to choose among all the feasible solutions, that policy whose implementation results in the overall minimum cost. The cost of a policy consists of a replenishment cost and an inventory related cost.

The replenishment cost consists of a major set-up cost which must be paid whenever one or more items are replenished and a minor set-up cost associated with each item, which must be paid whenever an item is included in a joint order.

The inventory related cost depends on the current inventory and the demand until the next decision point. There is a holding cost rate charged for excess stock and a backlogging cost rate, when backlogging is allowed, which is charged for each unit of demand not met directly from stock, except in the s, c, S model discussed in Chapter 5 where a cost which depends on the length of a stockout is also charged. Other costs parameters will be introduced as needed.

The demand pattern of each item is important in determining the method of solution to this problem. In most reasonable cases, the computational requirements for finding an optimum solution is prohibitive. Moreover, the form of the policy that is optimal, if known, is usually complex and difficult to implement. Because of these problems, researchers have concentrated more effort on the development of reasonable heuristics. Unfortunately, these heuristics were evaluated using very simple examples because there was no known good lower bound for this problem. There is now available a good lower bound which will be discussed in Chapter 2. This lower bound is useful in that it facilitates the process of developing heuristics. Very simple heuristics can now be tested quite easily as is done in chapters 4 and 5.

For the deterministic demand case, the pattern is usually either the time-varying and periodic demand [78,86,87] or stationary and continuous time demand, see for example [28,27,26,29,30,36,52,53,66,63]. For the stochastic case the pattern is usually either stationary and periodic demand or stationary and continuous time demands. Tan [74] characterized an optimum policy for a two-product problem with periodic demands. For stationary and continuous time demands, see for example [75,69,65,23,39,38,82].

Veinott [76] treated the case of non-stationary demands but no joint set-up cost.

The next sections are devoted to specifying the joint replenishment problem induced by each of the demand patterns above, and discussing some of the associated existing solution methods.

1.2 Deterministic Time-Varying Demand

Demands for the items are known in each period over the horizon but vary from one period to the other. In this case, a finite horizon is always assumed. Whenever a replenishment is made, a major set-up cost A_0 is paid, independent of the number of items in the joint order. In addition, a minor set–up cost A_i is paid if item i is included in the joint order. The unit variable cost of producing or purchasing each of the items is assumed constant for each item. The case with non-constant unit variable cost is not considered here. With this assumption, the total variable purchase or production cost for each item is constant for all feasible solutions which have no shortage, or excess stock at the end of the planning horizon. Since the models of this chapter do not permit shortages or excess stock at the end of the planning horizon, the unit variable procurement or production cost will be ignored in all subsequent analysis. At the end of each period, a holding cost is charged at the rate of h_i per unit, if there is excess stock of item i.

A more general version of this problem has been addressed. For example, Zangwill [86] and Kalymon [40] considered the case with a generalized procurement plus inventory cost; Veinott [78], Kao [41] and Leopoulos et al. [44] considered the case with seperable but concave procurement and inventory cost functions. The restricted version described here where the set-up costs are constant, is most common in practice and has been addressed by Silver [68].

There are many algorithms which find the optimum solution to this problem [40,41,68,78,86]. The common problem with these solution methods is that their computational requirements are enormous. The algorithms of Zangwill [86], Kao [41], Silver [68] and Leopoulos et al. [44] are exponential in the number of products, while those by Veinott [78] and Kalymon [40] are exponential in the number of time periods. None of these methods can be used for a problem with a moderate to large number of items and a long planning horizon.

A brief description of some of the solution methods will now be given.

Some Notations:

π_{it}	=	demand for product i in period t
X_{it}	=	replenishment quantity of item i in period t
X_i	=	$(X_{i1}, X_{i2}, ..., X_{iH})$
X_i^t	=	$(X_{it}, X_{it+1}, ..., X_{iH})$
X^t	=	$(X_{1t}, X_{2t}, ..., X_{Nt})$
I_{it}	=	inventory level of item i at the end of period t
I_t	=	$(I_{1t}, I_{2t}, ..., I_{Nt})$
N	=	number of items
H	=	number of periods in horizon

Model Assumptions:

1. The unit variable acquisition cost of any item is constant and therefore disregarded.

2. There is no backlogging of any item.

3. There is no capacity constraint.

4. Lead time is zero.

The single item problem with these assumptions has been studied extensively. Wagner and Whitin [80] were the first to give a dynamic programming solution to the problem. Their basic model has been expanded, sometimes to relax some of the assumptions above and sometimes to derive new solution properties for the basic problem and for a more general class of problems. For example Zabel [85] introduced planning horizon theorems which reduce the computational requirements of the Wagner-Whitin model.

Some properties of optimum solutions derived for the single item case have been extended to the multi-item problem; examples of such extensions can be seen in [66,86].

The most important of these properties which have been exploited in deriving the solution to the joint replenishment problem are given below.

Property 1: $X_{it} = \sum_{j=t}^{n} \pi_{ij} \quad t \leq n \leq H; \quad n$ integer and

$$n \leq \arg\min\{l : (l-t)h_i\pi_{il} > A_0 + A_i\}$$

Property 2: $X_{it} > 0$ implies $I_{it-1} \leq 0; \; I_{it} \geq 0.$

Property 1 says that when any item is replenished, the replenishment quantity must be equal to the sum of demands for an integer number of periods. Also, it is not desirable to purchase the requirement of period n in period t, $t < n$, if the cost to hold this requirement from period t to period n is greater than the total set-up cost. Property 2 says that a replenishment for any item takes place only when that item has zero inventory, and that the replenishment quantity must be sufficient to cover the demand of the period in which the replenishment takes place. The condition on n can also be treated as a property. It is the planning horizon concept which reduces the solution space to consider in searching for an optimum solution.

Let

$$D_t(I_{t-1}) = (X_1^t, X_2^t, ..., X_N^t).$$

where X_i^t is a feasible sequence of replenishments of item i from period t to the end of the planning horizon given that the starting inventory of the item in period t is I_{it-1}. If X_{it} satisfies properties 1 and 2 above, $D_t(I_{t-1})$ is known as the dominant set for period t if the starting inventory in period t is I_{t-1}, Zangwill [86]. This dominant set is simply the set of feasible replenishment schedules, given the starting inventory, which satisfy properties 1 and 2 above and by implication contain an optimum schedule. The construction of an optimum solution to the joint replenishment problem uses dynamic programming, efficiently exploiting the fact that an optimum replenishment schedule is contained in the dominant set. This idea was introduced by Zangwill [86] in the context of a more general problem. The methods of Zangwill, Kao, Leopoulos et al and Veinott, used for obtaining an optimum solution will be described in the next four subsections.

1.2.1 Zangwill's Algorithm

Zangwill devised a dynamic programming method for determining an optimum replenishment schedule from the dominant set of a general class of multi-item, multi-stage inventory problems. The algorithm specializes easily to the joint replenishment problem, and will be described in the context of this chapter.

Let

$$\begin{aligned} \Delta_t &= \sum_{i=1}^{N} X_{it} \\ H_t(X^t) &= \sum_{i=1}^{N} h_i X_{it} \end{aligned}$$

At any time t, let

$$n_{i1} = \max\{k \mid \sum_{j=t}^{k} \pi_{ij} = I_{it-1}\}$$

and

$$I_{it-1} = 0 \qquad \text{implies } n_{i1} = t-1$$

Thus n_{i1} is the time when the current inventory for item i will be fully consumed. For every i, define n_{i2} by

$$\begin{aligned} X_{it} &= \sum_{j=n_{i1}+1}^{n_{i2}} \pi_{ij};\ n_{i2} > n_{i1}; \qquad n_{i2} \text{ integer.} \\ n_1 &= (n_{11}, n_{21}, ..., n_{N1}) \\ n_2 &= (n_{12}, n_{22}, ..., n_{N2}) \\ \overline{H} &= \text{an } N\text{-dimensional vector of H} \end{aligned}$$

From the above definition, n_{i2} is the time when the current replenishment of item i, if any, will be fully consumed. The cost incurred in any period t can now be written as

$$P_t(n_1, n_2) = A_0\delta(\Delta_t) + \sum_{i=1}^{N}(A_i\delta(X_{it}) + h_i(X_{it} + I_{it-1} - \pi_{it})) \qquad (1.1)$$

where $\delta(x) = 0$ if $x \leq 0$ and $\delta(x) = 1$ if $x > 0$. Let $F_t(n_1)$ be the minimum replenishment plus inventory cost from period t to period H given that in period t, the starting inventory will meet the demands from period t to periods n_1. (Recall that n_1 is a vector). It can be shown that $F_t(\cdot)$ satisfies the following recursion.

$$\begin{aligned} F_t(n_1) &= \min_{n_1 \leq n_2 \leq \overline{H}} (P_t(n_1, n_2) + F_{t+1}(n_2)) \\ F_H(\overline{H}) &= 0. \end{aligned}$$

When backlogging is not allowed as in this case,

$$t - 1 \leq n_{i1} \leq H; \quad \text{for} \quad \text{every} \quad i$$

Although this algorithm produces an optimum solution, the state space required is very large. For example the maximum number of ways of selecting n_1 is H^N. For large H and N, this algorithm is clearly impractical.

Example

Let N=2, H=3, $h_1 = h_2 = 1$.

$A_0 = 5$	$\pi_{11} = 3$	$\pi_{21} = 4$
$A_1 = 2$	$\pi_{12} = 5$	$\pi_{22} = 3$
$A_2 = 3$	$\pi_{13} = 1$	$\pi_{23} = 4$

The relevant calculations are shown below.

$$\begin{aligned}
F_3(3,3) &= P_3[(3,3),(3,3)] = 0 \\
F_3(3,2) &= P_3[(3,2),(3,3)] = 8 \\
F_3(2,3) &= P_3[(2,3),(3,3)] = 7 \\
F_3(2,2) &= P_3[(2,2),(3,3)] = 10 \\
F_2(2,3) &= \min(P_2[(2,3),(2,3)] + F_3(2,3); P_2[(2,3),(3,3)] + F_3(3,3)) \\
&= 11 \quad n_2 = (2,3).
\end{aligned}$$

continuing in this way, the other values are:–

$$\begin{aligned}
F_2(3,2) &= 9 \quad \text{with} \quad n_2 = (3,2) \\
F_2(2,2) &= 10 \quad \text{with} \quad n_2 = (2,2) \\
F_2(1,2) &= 15 \quad \text{with} \quad n_2 = (3,3) \\
F_2(2,1) &= 15 \quad \text{with} \quad n_2 = (3,3) \\
F_2(1,3) &= 12 \quad \text{with} \quad n_2 = (3,3) \\
F_2(3,1) &= 13 \quad \text{with} \quad n_2 = (3,3) \\
F_2(1,1) &= 15 \quad \text{with} \quad n_2 = (3,3) \\
F_1(0,1) &= 15 \quad \text{with} \quad n_2 = (3,3) \\
F_1(1,0) &= 15 \quad \text{with} \quad n_2 = (3,3) \\
F_1(0,0) &= 25 \quad \text{with} \quad n_2 = (1,1)
\end{aligned}$$

The solution is to order both items for the first period only, and to order again at the beginning of the second period for the second and third periods. The total cost is 25.

One weakness of this algorithm is that it does not fully utilize property 2 which all optimum solutions have. The state space of the dynamic program is the vector of periods when the entry inventory levels will be fully consumed. Stages are the periods. Property 2 says that no replenishment is needed in time t, if all components of n_1 are strictly greater than t; that is if entry inventories for all items are all strictly positive. Thus, at stage t, it is only necessary to consider states where the vector of entry inventory contains at least one zero, or equivalently, at least one component of n_1 is $t-1$, except when $t = H$. If this fact is incorporated in the algorithm,the states needed will be considerably less. For example, for the problem above it is not necessary to calculate $F_2(3,3)$, $F_2(2,3)$, $F_2(3,2)$, $F_2(2,2)$ and $F_1(1,1)$. Kao [41] noticed this fact and incorporated it into his network algorithm. Apart from the fact that Kao's method uses a network approach, it is the same as Zangwill's algorithm plus property 2 on page 9. A brief description of Kao's method now follows.

1.2.2 Kao's Algorithm

As was indicated in the last subsection, Kao's algorithm is an efficient implementation of Zangwill's algorithm. It uses property 2 of optimal solution to reduce the state space at every stage, and formulated the problem as a network problem. Let M denote the node set of the network and M_t denote the node set that can be generated at stage t. The nodes and arc sets are defined as follows. Any element n_1 of M_t satisfies the following;

$$\begin{aligned} n_1 &= \{n_{11}, n_{21}, \cdots, n_{N1}\} \\ n_{i1} &= \{n | t-1 \le n \le H\} \end{aligned}$$

and

$$n_{j1} = t-1, \text{ for some } j.$$

and there is a directed arc from $n_1 \in M_t$ to $n_2 \in M_{t'}$ if and only if $n_2 > n_1$ and $t' > t$. The length of an arc from n_1 to n_2 at stage t, is given by $P_t(n_1, n_2)$ as defined in 1.1. With these nodes and arc set, the joint replenishment problem is to find the shortest part through the network, starting from the node which represents the initial inventory to the terminal node which is a vector of N components, with each element equal to H. The problem, can thus be stated as follows.

$$F_t(n_1) = \min_{n_1 \leq n_2 \leq H} \{P_t(n_1 n_2) + F_t(n_2)\}$$

Standard solution techniques for this problem can be found in basic texts on networks such as [34]. Although, this method involves a smaller state space than Zangwills method, it is still exponential in the number of products. One solution to the example in section 6 using this method will involve all the steps in Zangwill's algorithm except that $F_2(3,3)$, $F_2(3,2)$, $F_2(2,3)$, $F_2(2,2)$ and $F_1(1,1)$, will not be evaluated. Kao also proposed a heuristic which performs very well on some test examples.

Kao's heuristic

The heuristic is best described using a two product example, say product 1 and product 2.

Initialize: Set R to an N-dimenesional vector of zeroes. $l \leftarrow 1$

Step 1: Choose item 1 and define the set-up cost as:

$$a'_{1t} = \begin{cases} A_1 + A_0 & \text{if } t \not\in R \\ A_1 & \text{otherwise} \end{cases}$$

Step 2: Use Wagner–Whitin's algorithm for item 1 using a'_{1t} as set-up cost.

Let $R_1^l = \{t \mid X_{it} > 0\}$ If $R_1^l = R_1^{l-2}$, STOP; otherwise

$l \leftarrow l + 1$ and go to step 3.

Step 3: Select product 2 and augment the set-up cost as follows:

$$a'_{2t} = \begin{cases} A_2 + A_0 & \text{if } t \not\in R_1^l \\ A_2 & \text{otherwise} \end{cases}$$

Use Wagner–Whitin's algorithm for product 2 with a'_{2t} as set-up cost.

Let

$$R_2^l = \{t | X_{2t} > 0\}.$$

If $R_2^l = R_2^{l-2}$ STOP.

Otherwise $l \leftarrow l + 1$.

$R \leftarrow R_2^l$, and go to step 1.

For more than two products the order of product pairs has to be determined before step 1. For an N-product problem, a typical choice can be $(1,2),(2,3),\ldots,(N-1,N)$, $(N,1)$.

This algorithm can be improved with better starting values for R. One such starting value is described in Variant 1 below [41].

Variant 1

Let

$$a = A_0 + \sum_{i=1}^{N} A_i; \qquad \pi_t = \sum_{i=1}^{N} \pi_{it} h_{it}$$

Use Wagner–Whitin's algorithm with a as set-up cost, π_t as the demand in period t, and unit holding cost per item per period. Let

$$R = \{t | X_t > 0\}.$$

This is then used as initial value for R in step 0 of the heuristic.

Variant 2

Use every item as starting point in the heuristic in conjunction with variant 1. Select the minimum cost solution.

Although, this heuristic is reported to perform well, it is not satisfactory for a problem with a large number of items, considering the number of Wagner–Whitin type problems to be solved. Wagner–Whitin's algorithm is typically not used in practice even in the single item case. Some good heuristics, independent of Wagner–Whitin's algorithm will be discussed in Chapter 4. Another algorithm which is similar to that of Zangwill is that of Loupoulos and Proth, which will be discussed next.

1.2.3 Leopoulos–Proth's Algorithm

The algorithm by Leopoulos and Proth [44] is similar to Zangwill's algorithm in being exponential in the number of products.

Let $F_t(I_{t-1})$ be the minimum cost from period t to the end of the horizon given that the starting inventory in period t is I_{t-1}, $F_t(\cdot)$ satisfies the following recursive equation.

$$F_t(I_{t-1}) = \min_{x \in D_t(I_{t-1})} \{A_0\delta(\Delta_t) + \sum_{i=1}^{N}(A_i\delta(X_{it}) + h_i(X_{it} + I_{it-1} - \pi_{it}) + F_{t+1}(X_t + I_{t-1} - \pi_t)\}$$

$F_{H+1}(\cdot) = 0.$

This dynamic program can be shown to be identical to that of Zangwill, and therefore requires a larger state space than that of Kao. Leopoulos and Proth also gave a heuristic algorithm which depends on the following argument:

For simplicity assume there are only two products. *Suppose $X^* = (X_1^*, X_2^*)$ is an optimum schedule with a given initial starting inventory I_0. The value of X_1 which minimizes $F_t(I_0|X_2^*)$ is X_1^*.*

Leopoulos–Proth's Heuristic

Initialize: set $k \leftarrow 0$; $l \leftarrow 0$ X_1^0 = some feasible policy for product 1.

Step 1: Define the set-up cost as follows:

$$a'_{2t} = \begin{cases} A_2 + A_0 & \text{if } X_{1t} = 0 \\ A_2 & \text{otherwise} \end{cases}$$

Use Wagner–Whitin's algorithm for item 2 using a'_{2t} as set-up cost, to obtain X_2^l.

If $X_2^l = X_2^{l-1}$ STOP, otherwise go to step 2.

Step 2: $k \leftarrow k + 1$ and let

$$a'_{1t} = \begin{cases} A_1 + A_0 & \text{if } X_{2t}^l = 0 \\ A_1 & \text{otherwise} \end{cases}$$

Use Wagner–Whitin's algorithm for product 1 with a'_{1t} as set-up cost, to obtain X_1^k.

If $X_1^k = X_1^{k-1}$ STOP, otherwise

$l \leftarrow l + 1$ and go to 1.

This algorithm is applied using different starting values of X_1^0, randomly generated. The heuristic policy is the policy which gives the minimum cost.

This heuristic is similar to Kao's heuristic. The only noticeable difference is that Kao uses a systematic method to generate a starting feasible solution while Leopoulos–Proth randomly generates the starting solution. Since at least one of the randomly generated starting solutions will correspond to an optimum set-up times as the number of trials go to infinity, Leopoulos–Proth's method will eventually find an optimum solution. However there is no reason to believe that Kao's method of generating initial set-up times will give an optimum solution even for large number of trials.

The algorithms described thus far are based on Zangwill's method and therefore exponential in the number of products. In practice, the number of products involved in a joint replenishment problem is large while the number of periods may be small. This is partly due to the fact that demand forecasts far into the future are not very reliable. Veinott gave this as a justification for proposing a different solution to the joint replenishment problem which is exponential in the number of time periods and linear in the number of products.

1.2.4 Veinott's Algorithm

The algorithm is based on the fact that in any period, there is either a major set-up or there is none. All possible major set-up patterns can therefore be enumerated. The optimum replenishment schedule must correspond to at least on of these patterns. The algorithm is simply to generate a major set-up pattern, and determine using Wagner–Whitin's algorithm for each item an optimum replenishment schedule within the major set-up pattern. The cost for a pattern is the sum of the costs for all the items. The replenishment schedule for each item, corresponding to the pattern that gives the minimum total cost is the optimum replenishment policy. This algorithm is simple, but enumerating all the possible set-up patterns, which are at most 2^{H-1} patterns is obviously not an attractive thing to do in practice, unless the number of periods, H is less than about 15. Veinott also showed that when the minor set-up cost for each item is zero for all items, the replenishment schedule satisfies:

$$I_{it-1}\left(\sum_{i}^{N} X_{it}\right) = 0 \qquad \text{for every } i \text{ and } t$$

This means that no item is ordered unless the total inventory level is zero; all items are out of stock simultaneously.

1.3 Deterministic Stationary Demand Case

We now turn to continuous time models. The problem parameters are constant and stationary in time. Specifically, the demand rate for item i is π_i per period, which for convenience will be taken as one year. The holding cost rate per unit of item per unit of time is h_i. No backlogging of demand is allowed for any item. The major and minor set-up costs are as specified in the previous section. The problem is to determine an ordering policy which minimizes the average set-up plus inventory holding cost per unit time over the infinite horizon. This problem arises when a number of items, each of which has a constant demand per period, can benefit from a centralized replenishment policy. Such is the case if the items share a common production unit or if they have a common supplier or a common mode of transportation. The amount of each item to replenish each time a major replenishment process is initiated becomes important, as well as the average number of replenishments to be made per unit time.

A number of authors [20,26,28,27,36,46,53,64,66] have addressed this problem. The common denominator of their solution methods is that they use equally spaced replenishment epochs. These policies are not necessarily optimal as is argued in [5]. The optimum referred to in all these papers where 'optimum' is specifically mentioned should be qualified by the word 'periodic'. It is still an open problem what problem parameters ensure the existence of periodic optimum policies, except two cases. The first case is obvious and this is the case when all items have equal parameter values. This fact is easily seen from the lower bound for this class of problems, to be discussed in the next chapter. The second case which is slightly different from the model in this chapter has item-dependent set-up costs but with a fixed saving when all items are replenished together. For this problem Andres and Emmons [4] showed that an optimum policy has equally spaced replenishments epochs for each item. Even the problem of finding optimal periodic policies is not a trivial task, [27,66]. Because of this, researchers have focussed mainly on developing approximate methods for finding good periodic policies. Recently efforts at finding good solutions to this problem have shifted to specific periodic policies such as the power-of-two policies, introduced by Roundy [57] and also used in [36]. In this class of policies, it is assumed that a base period exists for operational reasons or otherwise, or that a convenient one can be found by optimization methods. All replenishments are then restricted to 2^l of this base period, with $l \geq 0$, a positive integer. This approach is interesting because it generally yields good solutions and in any case, has a worst case performance of less than 6% above the unknown optimum cost. In the next subsections, two methods, Silver

[66] and Goyal [27] which give good periodic policies and another one by Jackson et al [36] for finding optimum power-of-two policy will be discussed.

Assumptions and Notations:

Let

$$\begin{aligned}
g_i &= \frac{\pi_i h_i}{2} \\
T_i &= \text{reorder interval in years for product } i \\
T_0 &= \text{the major replenishment interval in years.} \\
N_i &= \text{the number of replenishments per year for item } i \\
N_0 &= \text{the number of major replenishments per year.} \\
\frac{\pi_i}{N_i} &= \text{replenishment quantity per replenishment epoch for item } i \\
k_i &= \frac{N_0}{N_i}
\end{aligned}$$

Note that if the period is 1 year,

$$T_0 = \frac{1}{N_0} \quad \text{and} \quad T_i = \frac{1}{N_i}$$

Problem Formulation.

This formulation assumes equally spaced replenishments. Let $C(N_0, \overline{N})$ denote the cost per year given N_0 and $\overline{N} = (N_1, N_2, \cdots, N_N)$

$$\begin{aligned}
C(N_0, \overline{N}) &= A_0 N_0 + \sum_{i=1}^{N} (A_i N_i + \frac{g_i}{N_i}) && (1.2) \\
&= A_0 N_0 + \sum_{i=1}^{N} (\frac{N_0 A_i}{k_i} + \frac{g_i k_i}{N_0}) && (1.3) \\
&= \frac{A_0}{T_0} + \sum_{i=1}^{N} (\frac{A_i}{T_0 k_i} + T_0 g_i k_i) && (1.4)
\end{aligned}$$

Letting $\overline{k} = \{k_0, k_1, \ldots, k_N\}$, equation 1.3 can be written as

$$C(N_0, \overline{k}) = N_0 \sum_{i=0}^{N} \frac{A_i}{k_i} + \frac{1}{N_0} \sum_{i=0}^{N} g_i k_i \qquad (1.5)$$

and equation 1.4 can be written as

$$C(T_0, \bar{k}) = \sum_{i=0}^{N} \left(\frac{A_i}{k_i T_0} + g_i k_i T_0\right) \tag{1.6}$$

where $k_0 = 1$ and $g_0 = 0$.

The problem is to determine N_0 and k_i which minimize 1.5 or equivalently T_0 and k_i which minimize 1.6 with k_i restricted to the positive integers for every i.

1.3.1 Goyal's Method

Goyal's approach is to choose N_0 and k_i for every i, to minimize the cost function expressed 1.5. Define the *natural cost* of an item as the single item cost, if replenishment could be made at the minor set-up cost. For every item i, let

$$\begin{aligned} n_i' &= \text{natural order frequency for item } i \\ C^i &= \text{the natural cost for item } i \end{aligned}$$

C^i is given by:

$$C^i = A_i n_i' + \frac{g_i}{n_i'}$$

and this is minimized when:

$$n_i' = \left(\frac{g_i}{A_i}\right)^{1/2}$$

Let $C_i(N_0, k_i)$ denote the cost of replenishing item i independently of all other items, given the major replenishment frequency is N_0.

$$C_i(N_0, k_i) = \frac{A_i N_0}{k_i} + \frac{g_i k_i}{N_0} \tag{1.7}$$

For a given N_0 let $k_i(N_0)$ minimize 1.7.

$$k_i(N_0) = \left(\frac{A_i N_0^2}{g_i}\right)^{1/2}$$

Since $k_i(N_0)$ must be an integer, it must satisfy:

$$C_i(N_0, k(N_0)) \leq C_i(N_0, k(N_0)+1) \tag{1.8}$$

and

$$C_i(N_0, k(N_o)) < C_i(N_0, k(N_0)-1) \tag{1.9}$$

Equations 1.8 and 1.9 are respectively equivalent to

$$\frac{N_0^2 A_i}{g_i} \leq k_i(N_0)[k_i(N_0)+1] \tag{1.10}$$

and

$$\frac{N_0^2 A_i}{g_i} \leq k_i(N_0)[k_i(N_0)-1] \tag{1.11}$$

which both imply:

$$n_i' k_i(N_0)[k_i(N_0)-1] \leq N_0 \leq n_i' k_i(N_0)[k_i(N_0)+1] \tag{1.12}$$

A simple justification of 1.12 is given in [81]. For any value of N_0, it is easy to determine $k_i(N_0)$ from 1.12. Thus a search over all reasonable values of N_0 will give the desired result from the point of view of minimizing the cost function given in 1.5.

Goyal [27] used the following two conditions to restrict the range of N_0 values to search.

Let $L = \arg\max_i n_i' = \{i | \max_i n_i' = n_L'\}$.

Condition one:

$$N_0 \leq N_L$$

This condition states that the major replenishment frequency cannot be greater than the natural replenishment frequency of the item with the largest replenishment frequency.

Andres and Emmons [5] pointed out that this condition is not true in general. It is only true when the policy space is restricted to the class of policies for which the major set-ups are equally spaced. Because of this and the fact that $k_i(N_0)$ are restricted to integers, the 'optimum' found by Goyal can at best be optimum in this class of policies.

Condition Two:

Set $k_i = 1$ for every i. This implies that all items are ordered at every major set-up. Let N_0' denote the ordering frequency under this rule. The best value of N_0' can be found from 1.6, since the cost function is a convex function of N_0. The value is:

$$N_0' = \frac{\sum_{i=0}^{N} g_i}{\sum_{i=0}^{N} A_i} \tag{1.13}$$

A necessary condition for N_0 to be optimum is that:

$$N_0 \geq N_0'$$

Goyal's Algorithm

Step 1: Find n_L' and N_0' from 1.12 and 1.13 respectively. Arrange all the items in descending order by the value of n_i'.

Step 2: Find $\theta_i \geq 0$ for every i such that

$$k_i(N_0' + \theta_i) = k_i(N_0') + 1.$$

Let $N_i' =: N_0' + \theta_i$. Because of step 1, $N_i' \leq N_j'$ for every $j > i$, with the restriction that $N_0' + \theta_i \leq n_L'$ for every i.

Let m be the number of distinct N_i'.

Step 3: For N_0 in each range, $N_0' - N_1', N_1' - N_2', \ldots, N_m' - n_L'$, determine an optimum value of k_i for every i, and the associated cost using 1.5. Let k_i^* be the value of k_i which gives the overall minimum cost after searching all the ranges.

Step 4: Determine an optimum major replenishment frequency by

$$N_0^* = \frac{\sum_{i=0}^N g_i k_i^*}{\sum_{i=0}^N (A_i/k_i)}$$

This algorithm obviously requires a lot of computations. Roundy [57], used an algorithm similar to this, but restricted attention only to power-of-two policies, and showed that the computational requirement not only reduces considerably, but the solution found within the class of power-of-two policies is at most 2% above the unknown optimum. Silver devised a rather simple but relatively good heuristic for this problem.

1.3.2 Silver's Method

Silver [66] gave a very simple closed form equation for selecting values of k_i for every i, and T_0 which give close to the minimum value of the average replenishment plus inventory holding cost given in equation 1.6

Consider 1.6

$$C(T_0, \overline{k}) = \frac{1}{T_0}\sum_{i=0}^N \frac{A_i}{k_i} + T_0 \sum_{i=0}^N g_i k_i$$

with $k_0 = 1$ and $g_0 = 0$.

For a given $\overline{k}$, the cost is convex in T_0. Minimizing with respect to T_0 gives

$$T_0^*(\overline{k}) = \left(\frac{\sum_{0=1}^N A_i/k_i}{\sum_{i=0}^N g_i k_i}\right)^{1/2} \qquad (1.14)$$

and

$$C(T_0^*, \overline{k}) = 2\left(\sum_{i=0}^N (\frac{A_i}{k_i}) \sum_{i=0}^N (g_i k_i)\right)^{1/2} \qquad (1.15)$$

The problem is to select k_i for every i to minimize 1.15. The approach used in [66] is first to ignore the constraint that every k_i be restricted to the positive integers, and minimize 1.15 with respect to $\overline{k}$. This is done by taking the partial derivatives with respect to each k_i and setting the result equal to zero. This gives

$$-\frac{A_i}{k_i^2}\sum_{i=1}^{N} g_i k_i + g_i \sum_{i=0}^{N}\frac{A_i}{k_i} = 0, \quad i = 1, 2, \ldots, N \tag{1.16}$$

As is shown in Schweitzer and Silver [63] this system has no solution when $A_0 > 0$. To show this multiply 1.16 by k_i and sum over all i to obtain

$$A_0 \sum_{i=1}^{N} g_i k_i = 0$$

which implies that $A_0 = 0$.

Call the item whose A_i/g_i is minimum over all i item 1. Despite the fact that system 1.16 does not have a solution, if the constraint $k_1 = 1$ is added to the system, the solution obtained by this method gives the correct relative values of the k_i's. Solving system 1.16 with this restriction gives;

$$\begin{aligned} k_j &= \left(\frac{A_j}{g_j}\right)^{1/2} c; \qquad j = 2, \cdots, N \\ k_1 &= 1 \end{aligned} \tag{1.17}$$

where

$$c = \left(\frac{\sum_{i=1}^{N} k_i g_i}{\sum_{i=0}^{N} A_i/k_i}\right)^{1/2} \tag{1.18}$$

From 1.17

$$\sum_{i=1}^{N} k_i g_i = g_1 + c\left(\sum_{i=2}^{N} A_i g_i\right)^{1/2} \tag{1.19}$$

and

$$\sum_{i=1}^{N}\frac{A_i}{k_i} = \frac{\left(\sum_{i=2}^{N} A_i g_i\right)^{1/2}}{c} + A_1 \tag{1.20}$$

Substituting 1.19 and 1.20 into 1.18 gives

$$c = \left(\frac{g_i}{A_0 + A_1}\right)^{1/2} \tag{1.21}$$

From which

$$k_j = \left[(\frac{A_i}{g_i})(\frac{g_i}{A_0 + A_1}) \right]^{1/2} \qquad j = 2, \ldots, N. \tag{1.22}$$

The k_j obtained will not necessarily be integer. Simply rounding to the nearest integer is found to be satisfactory, although a search routine can be used.

Goyal and Belton [29] suggested a modification to this method. They pointed out that the item to be denoted as item 1 should be that item with the minimum value of $(A_0 + A_i)/g_i$. Computational results with this modification show some improvement over Silver's basic method which is ideal for the case with $A_0 = 0$. Silver's method is simple and performs relatively well when compared to Goyal's method. Silver showed that for the case with two items, the algorithm gives a solution that is always better than 1% above the optimum.

Attention has recently shifted to a special class of periodic policies, called power-of-two policies. The next subsection will describe the method of Jackson et al for computing an optimum power-of-two policy for the joint replenishment problem.

1.3.3 Jackson-Maxwell-Muckstadt's Method

The method of Jackson-Maxwell-Muckstadt [36] is to determine an optimum power-of-two policy for a given base planning period. The basic cost equation used is similar to equation 1.6.

The problem is to minimize the average cost given by:

$$C(\overline{k}) = \sum_{i=0}^{N} \left(\frac{A_i}{k_i T_0} + g_i k_i T_0 \right)$$

with the restriction that;

$$k_i \in \{2^l;\ l = 0, 1, \ldots\}, i = 0, 1, 2, \cdots, N.$$

and

$$k_i \geq k_0$$

In this formulation T_0 is assumed fixed for operational reasons or otherwise. If the second constraint is relaxed and k_0 is fixed, the problem becomes separable, and an optimum $k_i's$ are given by:

$$k_i^* = 2^{l_i^*}; \qquad \text{for every } i$$

where l_i^* is a non-negative integer that minimizes-

$$\frac{A_i}{2^{l_i}T_0} + g_i 2^{l_i} T_0$$

If the minimum is unique, l_i^* is the smallest integer that satisfies:

$$\frac{A_i}{2^{l_i}T_0} + g_i 2^{l_i} T_0 \leq \frac{A_i}{2^{l_i+1}T_0} + g_i 2^{l_i+1} T_0$$

This is equivalent to:

$$2^{l_i} \geq \left(\frac{A_i}{2g_i T_0^2}\right)^{1/2}$$

Take log to base 2 of both sides to obtain,

$$l_i^* = log_2 \left\lfloor \left(\frac{A_i}{2g_i T_0^2}\right)^{1/2} \right\rfloor_2$$

where $\lfloor x \rfloor_2$ is the smallest integer power of two greater than or equal to x.

If the minimum is not unique, the other minimizing value of l_i is the above quantity plus 1.

The Algorithm

Step 1: Sort items in ascending order of A_i/g_i. Let

$$\aleph = \{0, 1, \cdots, i^*\}$$

where

$$\frac{\sum_{j=0}^{i^*} A_j}{\sum_{j=1}^{i^*} g_j} \geq \frac{A_i}{g_i} \qquad \text{for} \quad i \leq i^*$$

and

$$\frac{\sum_{j=0}^{i^*} A_j}{\sum_{j=1}^{i^*} g_j} < \frac{A_i}{g_i}; \qquad \text{for} \quad i > i^*$$

Step 2:

$$k_0 = 2^{l_0} = \left\lfloor \left(\frac{\sum_{j=0}^{i^*} A_j}{2T_0 \sum_{j=0}^{i^*} g_j} \right)^{1/2} \right\rfloor_2$$

Step 3: Set $k_j = k_0$; for $j \in \aleph$ and $k_j = \left\lfloor \left(\frac{A_j}{2T_0^2 g_j} \right)^{1/2} \right\rfloor_2$; for $j \notin \aleph$.

This algorithm is accomplished in $O(N \log N)$ time and is proved to have a worst case performance of at most 6% above the unknown optimum.

The mathematical justification for this algorithm, and especially for step 2 is somewhat complex as given by the authors. A very simple method for justifying step 2 will be given in Chapter 2.

1.4 Random Demand Case

The joint replenishment problem under random demands has been studied by many authors, [6,11,38,39,65,69,23,76]. Not much is known about optimum policies for the joint replenishment problem with both major and minor set-up costs. Tan [74] characterized the optimum policy for a two-product problem under random demands and periodic review. In some cases, for example when there is no major set-up cost, Veinott [76] showed that an optimum policy is of the (s, S) type. When there is a major set-up cost but no minor set-up cost, Johnson [38] and Kalin [39] have shown that an optimum policy is of the (σ, S) type, where σ is a subset, referred to as the 'do-not-order' set, of the possible inventory position of the system, when an order is not desirable. The characterization of the set σ is complex and poses computational problems, rendering the (σ, S) policy impractical. In the face of these difficulties, researchers have tended to focus on heuristic policies like the (s, c, S) policy where an item i is ordered whenever its inventory position reaches its 'must-order' point s_i, and any other item j whose inventory position is at or below its 'can-order' point c_j, is also ordered. The amount ordered in either case is enough to raise the inventory level of the item i (resp., j) to its 'order-up-to' point S_i (resp., S_j). This policy was proposed by Balintfy [6]. Silver and other authors have developed good procedures for computing the values of the control parameters s, c and S [23,65,68,69,75]. The (s, c, S) policy is not an optimal policy, as was shown by Ignall [35] but it is judged to be a reasonable policy and has been shown empirically to give cost savings of sometimes up to 20% below the cost of an uncoordinated policy. The uncoordinated policy uses the major set-up plus the item-dependent set-up cost as the set-up cost for each item. The demand patterns used in evaluating the s, c, S policy have so far been Poisson demands [65,68] and compound Poisson demands [75,69,23]. Assumptions common to all the papers are that the opportunities for an item to be ordered at its 'can-order' point are independently distributed according to the Poisson process with rate equal to the expected number of major replenishments per period that are not caused by the item and that the demands are uncorrelated. These assumptions simplify the required mathematics and the solution still shows considerable cost savings over the independent policy. The basic method used consists of a decomposition procedure and an iterative routine. Silver [65] developed a simple sequential method of determining the values of s, S that solve a single item problem under Poisson demand. The values of c for the items are then selected iteratively until convergence in the cost is achieved. Thompstone and Silver [75] also developed a method for compound Poisson demands. The paper by Federgruen et al [23] differs from previous work in one essential aspect: the method for solving the decomposed problem, that is, the single item problem, is a specialized policy iteration method.

Other interesting work in this area include those of Miltenburg and Silver [48,49]. They determined the probability distribution of the residual stock of items jointly replenished at the time of replenishment by modelling the inventory process as a diffusion process in both periodic and continuous review environments. Heuristics were then developed for ordering policies which take proper account of residual stock. They showed that these heuristics perform better than the IBM's IMPACT inventory control package. Naddor [52] compared different policies e.g, periodic, s,S and other policies for single item problem and developed a simple method of determining good periodic policies for the joint replenishment problem.

1.5 Other Related Models

Clark [15] gave a general survey of multi-echelon inventory models. An interesting survey of general physical distribution problems in inventory management is given by Schwarz [61]. A particular model of interest is the dynamic multi-location inventory model. In this model there are many locations supplied by a central warehouse that may or may not hold stock. When the warehouse does not hold stock, it acts merely as a distribution center. For this problem where the emphasis is on the distribution of a fixed amount of stock to minimize inventory cost, Simpson [70] showed that when the penalty costs are equal across locations, the optimum rule is to distribute the stock so as to equalize the probability of shortage in all locations. When the penalty costs are different the distribution rule is to equalize the weighted probability of shortage across locations, the weights being the penalty costs. Other early studies of this problem can be found in [3]. Recently Eppen [21] showed the exact effect of centralization on the expected inventory cost for a one-period model. Eppen and Schrage [22] later considered centralized ordering with lead time in a multi-period setting with normal demand distribution. Following this lead, Federgruen and Zipkin [24,25] developed good approximate allocation rules by aggregating demands across locations and over time. They used normal, exponential and gamma demand distributions all with constant coefficients of variations.

When the warehouse can hold stock, the problem becomes a one-warehouse multi-retailer problem. This problem and the generalization has been studied widely. Veinott [78] and Kalymon [40] gave exponential running time algorithms for this problem with discrete and deterministic demand under periodic review. Schwarz [60] gave some properties which any optimum solution of the deterministic demand, continuous review version of this problem must possess. Roundy [57] exploited these properties along with restricting the policy to power-of-two policies and found an algorithm which runs in $N \log N$ time, if there are N end facilities called retailers, and showed that such a solution has effectiveness of 98% in the worst case. This result in part motivates the research reported in this thesis.

A 98% effective power-of-two solution for the general multi-stage, multi-facility production/inventory problem under continuous review, with a fixed set-up cost depending on the group of items replenished at a facility has also been given by Roundy [59]. This result was generalized by Queyranne [55] to include submodular set-up cost functions. Other multi-echelon inventory models include the assembly systems in MRP setting. Early work in this area include Crowston et al [17,18], Schwarz and Schrage [62] Graves [31] and Blackburn and Millen [9]. Others can be found in [56]. Recently Afentakis et al [1] used a Langragean relaxation method to develop a lower bound on the cost for an assembly system under periodic review and used a branch and bound integer programming method for solution. Roundy [56] developed a 94% effective power-of-two solution to the assembly problem under continuous review in the infinite horizon.

CHAPTER TWO

A Lower Bound on a Class of Production /Inventory Problems

Introduction

A cost allocation method is presented for determining lower bounds for a class of production/inventory problems. We have N products which are replenished together or individually. A sequence of reorders for any particular item will incur holding, backorder and set-up costs specific to the item. In addition whenever *any* item is replenished a joint cost A_0 is incurred. For instance, if several items are replenished simultaneously, the joint cost is incurred. But if only one item is replenished, the joint cost is also incurred.

What is required of the total problem is to minimize a cost function of the replenishment sequence. This might be a long run average or a total cost function. It should be clear that any feasible replenishment sequence for the total problem would also give a feasible replenishment sequence to each item individually. If, at any instance of joint replenishment, the joint cost A_0 is shared amongst all items so that each item i gets a fraction $\alpha_i A_0$ of the cost, and

$$\sum_{i=1}^{N} \alpha_i = 1; \quad \alpha_i \geq 0 \tag{2.1}$$

then the sum of the costs implied by each feasible replenishment sequence on each item will not be greater than the cost of the total problem. Thus the sum of the minimum costs incurred for each item after reallocating the joint cost will be a lower bound on the minimum cost for the total problem. This is true for any allocation which satisfies 2.1 above and is true in particular if we maximize the lower bound with respect to $\alpha = (\alpha_1, \ldots, \alpha_N)$.

In the sequel we show how to use this cost allocation method to derive lower bounds on the costs of any feasible replenishment policy for each of the following problems:

1. The (single facility) joint replenishment problem
2. The one-warehouse multi-retailer problem
3. The multi-facility joint replenishment problem.

For the one-warehouse multi-retailer problem the item reorder which is relevant in applying the cost allocation method is the reorder at the warehouse. Also for the multi-facility joint replenishment problem, the relevant item reorders are the reorders at facilities which distribute more than one item.

The one-warehouse multi-retailer problem is the production/inventory problem where a warehouse supplies an item to many retailers. The (single facility) joint replenishment problem is a special case of the one-warehouse multi-retailer problem. In the joint replenishment problem the warehouse does not hold any stock. The multi-facility joint replenishment problem is a problem with an arbitrary number of warehouses such that a warehouse has a unique immediate predecessor. A warehouse supplies an arbitrary number of products to warehouses or retailers in the set of its immediate successors and so on until the products get to the retailers.

One common feature of these problems is that a warehouse or retailer has a unique immediate predecessor but may have any number of successors. We shall sometimes refer to a warehouse or retailer simply as a facility. Another feature of these three groups of problems is that a group of items shares a common replenishment or production set-up cost, which must be paid whenever one or more items in the group is replenished or produced. This joint cost is referred to as a *major* set-up cost at the facility where the joint replenishment takes place. The group of items that shares this major set-up cost includes all the items supplied by the end facilities that are also successors of the 'major' facility. Each item has a 'minor' or 'line' set-up cost which is paid when a particular item is withdrawn by an *end facility*, that is, a retailer. There is also an inventory cost at each facility which depends on the stock level and the stock withdrawal pattern at the facility. The specific nature of this cost depends on the problem structure and will be described as each example is introduced.

The problem is generally to schedule replenishments of the various items at each facility so as to minimize the total cost or average cost per unit time of ordering and holding inventory at each facility over a given horizon, subject to a given set of constraints. The *effectiveness* of a feasible solution is the ratio of the best available lower bound on the cost of any feasible solution to the cost of that feasible solution. One use of lower bounds is in evaluating the effectiveness of heuristic solutions to difficult problems. 94% and 98% effective solution methods exist respectively for the joint replenishment problem [36,57] and the one-warehouse multi-retailer problem [57]; both with stationary, deterministic and continuous-time demands. Another major use of lower bounds is apparent from the methods of obtaining the 94% and 98% effective solutions for the above problems: that of deriving feasible solutions from the bound.

The bound introduced here is derived by using a general principle of allocating the set-up cost of a facility amongst all the items that are replenished at the facility. The total of all allocated fractions of cost is required to be unity for feasibility. With this allocation scheme, we can decompose the problem into N facilities-in-series problems, if there are N end facilities, as illustrated in figures 2.1 and 2.2 below.

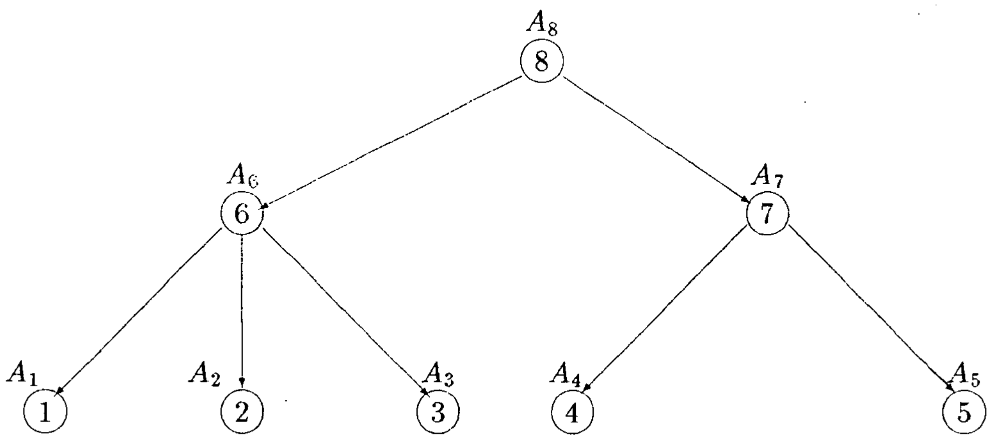

Figure 2.1: A Multi-Facility Joint Replenishment Problem

The decomposed problem corresponding to figure 2.1 is shown in figure 2.2 below. It is required that

$$\begin{aligned} \alpha_{18}+\alpha_{28}+\alpha_{38}+\alpha_{48}+\alpha_{58} &= 1 \\ \alpha_{16}+\alpha_{26}+\alpha_{36} &= 1 \\ \alpha_{47}+\alpha_{57} &= 1 \end{aligned}$$

The major result of this cost allocation method is that:

the sum of the minimum costs of the feasible solutions to the subproblems, for every allocation scheme such that the total fractional cost allocation at each facility is unity, is a lower bound on the cost of any feasible solution to the original problem.

Because it is usually easier to solve a facilities-in-series problem, the bound is easy to derive for every cost allocation scheme. The only problem is to determine a cost allocation scheme which gives a 'good' bound.

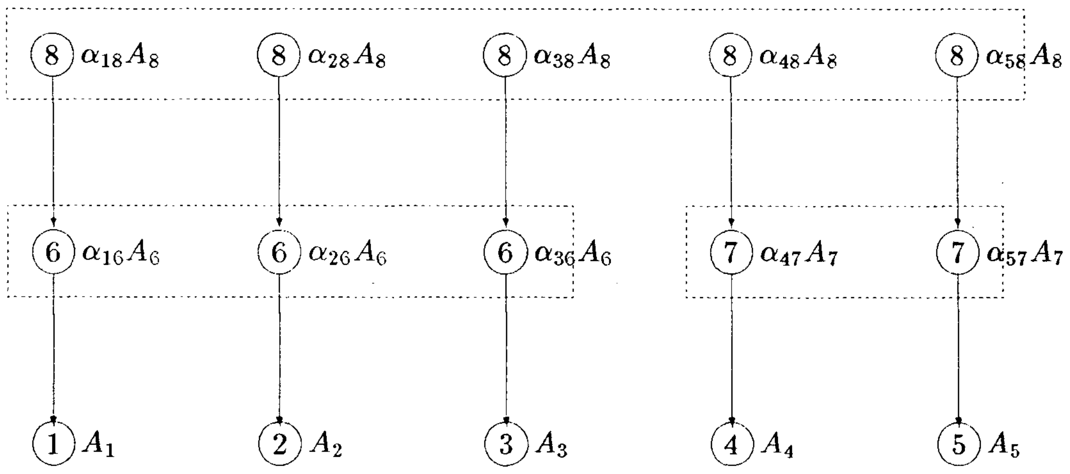

Figure 2.2: Cost Allocation for Fig. 2.1

The major part of this chapter will be devoted to determining good bounds for the joint replenishment and the one-warehouse multi-retailer problems, using the cost allocation method. The application of this general bound to the multi-facility joint replenishment problem will be deferred to the next chapter. These bounds will, henceforth, be referred to as the *cost allocation bounds*. The *cost allocation bound*, for each of the problems, coincides with the bounds in [36] for the joint relenishment problem, and in [57] for the one-warehouse N-retailer problem. One further advantage of the bound, apart from the simplicity of deriving it, is its very intuitive base, which renders it very general.

The *cost allocation bound* is determined and proved for the joint replenishment problem in the next section, while the same is done for the one-warehouse multi-retailer problem in section 3.

2.2 The Joint Replenishment Problem

In general there is a major set-up cost A_0 which is paid whenever a replenishment is made, and a minor set-up cost A_i also paid whenever item i is replenished. The joint replenishment problem can generally be categorized according to the demand patterns of the items involved, because the methods of solution are similar for similar demand patterns. Three cases will be considered.

2.2.1 Deterministic, Continuous-Time Stationary Demand

It is desired to schedule the replenishment or production of N items over the infinite horizon. Each item experiences a known constant and continuous demand per period. The inventory related cost consists of a linear echelon holding cost rate per unit of stock per unit time.

The concept of echelon holding cost reflects the value added to an item as it moves from a facility to the successor facility. In the joint replenishment problem, a major assumption is that no value is added in moving items from the warehouse to the retailer, so the echelon holding cost at the retailer is zero. The warehouse and the retailers are in fact the same physical facility.

The individual unit variable procurement cost is assumed constant throughout the horizon, and need not be considered since backlogging is not allowed. The total accumulated cost of procurement will clearly not depend on any specific feasible replenishment policy with no excess stock at the end of the horizon.

We consider the class of feasible policies consisting of all policies which replenish an item only when the inventory of the item is zero, that is an item is replenished only at the last minute. It is easy to check that this class of policies dominate any other class of policies in the sense that it contains the optimum replenishment schedule. Let a particular feasible policy in this class be specified by Ψ. Ψ will be uniquely defined by

- the replenishment epochs of item i denoted by $\mathbf{T}_i^0$.
- the replenishment quantities of item i, $Q_i(\mathbf{T}_i^0)$ at the epochs $\mathbf{T}_i^0$.

Let Ψ_i be the replenishment schedule induced on item i by Ψ. We see that Ψ_i is feasible for item i when viewed alone. We can then say that Ψ consists of Ψ_i for all i in the group, and hence write

$$\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_N)$$

and $Z(\Psi)$ as the average cost of policy Ψ.

2.2.2 The Cost Allocation Method

The cost allocation method requires that a set-up cost that is common to a group of items be shared amongst the items in the group. In the joint replenishment problem, the major set-up cost is shared amongst all the items that can be jointly replenished. Let α_i be the fractional allocation of the major set-up cost to item i. It is required that the total of all fractional allocations derived from this major set-up cost be unity.

By this allocation, the problem decomposes to N two stage facilities-in-series problems. Let $Z_i(\Psi_i, \alpha_i)$ be the cost for the i^{th} facilities-in-series problem if the allocation to item i is α_i and the policy is Ψ_i. The pattern of inventory levels for item i resulting from using Ψ for the general problem is exactly the same as that resulting from using Ψ_i for the i^{th} facilities-in-series system. Thus the inventory holding costs are equal in both cases. Let us now examine the set-up costs. Let τ be a member of some $\mathbf{T}_i^0$. This means that at time τ, there is a minor set-up and therefore also a major set-up. Let E_τ be the set of items that are replenished at time τ. The total minor set-up cost at time τ is the same using policy Ψ, or the individual facilities-in-series, that is using Ψ_i individually. The major set up cost using Ψ for the general problem is A_0. Using the individual policies Ψ_i, the total major set-up costs incurred at time τ is the quantity given below:

$$\text{Sum of the major set-up costs at time } \tau = A_0\Big(\sum_{i\in E_\tau} \alpha_i\Big) \leq A_0.$$

We have thus shown that the sum of the set-up costs for all the facilities-in-series problems is always bounded above by the total set-up cost for the original problem, and the patterns of inventories in the system are identical in both cases. Therefore

$$Z(\Psi) \geq \sum_{i=1}^{N} Z_i(\Psi_i, \alpha_i). \tag{2.2}$$

Thus the cost allocation method gives a lower bound on the cost of all feasible policies to the joint replenishment problem. The result applies to any demand pattern at the retailers and with *backorders*, including *random demands*, since it applies verbatim to *each sample path* if the items are distinct so that there is no risk pooling.

2.2.3 The Lower Bound Problem

Let

$$PZ_i(\alpha_i): \qquad Z_i(\alpha_i) = \min_{\Psi_i} Z_i(\Psi_i, \alpha_i).$$

The echelon holding cost at the retailer is zero in the joint replenishment problem. Because of this, in any feasible solution to the i^{th} facilities-in-series problem we must have that the replenishment epochs of the warehouse in the facilities-in-series coincide with that at the retailer. The i^{th} facilities-in-series problem is thus equivalent to a simple Economic Order Quantity (EOQ) problem with a set-up cost $\alpha_i A_0 + A_i$ and a holding cost rate of h_i. Without loss of generality, we assume for the rest of the analysis that delivery lead time is zero, and that demands occur at a constant rate of 2 per unit time. This latter assumption was first used by Roundy [57]. We also assume that there is no backlogging of demands. With these assumptions we can write $Z_i(\alpha_i)$ explicitly as

$$Z_i(\alpha_i) = 2\sqrt{(\alpha_i A_0 + A_i)h_i} \quad \text{and} \quad Z(\alpha) = \sum_{i=1}^{N} Z_i(\alpha_i).$$

Let

$$Z(\alpha^*) = \max_{\alpha}\{Z(\alpha) | \sum_{i=1}^{N} \alpha_i = 1; \quad \alpha_i \geq 0\}$$

We have pointed out that the lower bound in equation 2.2 is true for any given feasible vector α. In particular

$$Z(\Psi) \geq Z(\alpha^*)$$

The desired lower bound is given by $Z(\alpha^*)$ written below.

$$Z(\alpha^*) = 2\{\max_{\alpha} \sum_{i=1}^{N} \sqrt{(\alpha_i A_0 + A_i)h_i} | \sum_{i=1}^{N} \alpha_i = 1; \quad \alpha_i \geq 0\} \tag{2.3}$$

Because Z_i, is strictly concave and continuously differentiable in α_i, the Kuhn-Tucker necessary conditions are also sufficient for optimality. Thus, after introducing the Lagrange multiplier λ, and letting prime denote the first derivative, the relevant $K - T$ conditions can be written as:

$$\begin{aligned} Z_i'(\alpha_i) &= A_0(\frac{h_i}{\alpha_i A_0 + A_i})^{1/2} \\ &\leq \lambda && (2.4) \\ \alpha_i &> 0 \Longrightarrow Z_i'(\alpha_i) = \lambda && (2.5) \\ \sum_{i=1}^{N} \alpha_i &= 1 && (2.6) \end{aligned}$$

The first two conditions imply that N is partitioned into two sets of items $\aleph_1$ and $\aleph_2$, such that:

$$\begin{aligned} i \in \aleph_1 &\Longrightarrow \alpha_i > 0 \\ i \in \aleph_2 &\Longrightarrow \alpha_i = 0 \end{aligned}$$

Note that $\aleph_2$ can be empty. From condition 2.5, it can be deduced that:

$$i \in \aleph_1 \Longrightarrow \frac{\alpha_i A_0 + A_i}{h_i} = \text{constant} = (\frac{A_0}{\lambda})^2$$

Combinining this with condition 2.6 and solving for α_i, $i \in \aleph_1$, gives;

$$\alpha_i = \frac{(\frac{A_0 + \sum_{i \in \aleph_1} A_i}{\sum_{i \in \aleph_1} h_i}) h_i - A_i}{A_0} \tag{2.7}$$

The problem is to partition the items into the sets $\aleph_1$ and $\aleph_2$, such that α_i defined above is non-negative. The partitioning problem is solved in [36]. We give a similar algorithm which is consistent with the spirit of our lower bound.

The algorithm

step 0 Initialize

Let $c \leftarrow$ a big number, $t_i = A_i/h_i$ for all i. Rank all t_i in ascending order and assume for simplicity that $t_1 \leq t_2 \cdots \leq t_n$; set $k \leftarrow 1$, $\aleph_2 \leftarrow \{1, 2, \ldots, N\}$, $\aleph_1 \leftarrow \emptyset$ $a \leftarrow A_0$, $h \leftarrow 0$.

step 1 Allocation

If $c > t_k$, do the following: $a \leftarrow a + A_k$, $h \leftarrow h + h_k$, $\aleph_2 \leftarrow \aleph_2 \setminus \{k\}$, $\aleph_1 \leftarrow \aleph_1 \cup \{k\}$, $c \leftarrow \frac{a}{h}$;

$k \leftarrow k + 1$, and go to step 1;

otherwise the problem is solved the current partition is optimum, go to step 2.

step 2 Calculate α_i using equation 2.7 for all $i \in \aleph_1$ and let $\alpha_i \leftarrow 0$ for all $i \in \aleph_2$

The cost given by the algorithm is then

$$2\sqrt{(A_0 + \sum_{i \in \aleph_1} A_i)(\sum_{i \in \aleph_1} h_i)} + 2 \sum_{i \in \aleph_2} \sqrt{A_i h_i}$$

which coincides with the lower bound in [36]. A close look at the set $\aleph_1$ reveals that the 'run-out' times (the cycle times in this case) for each $i \in \aleph_1$ is λ/A_0, and these are the items with $\alpha_i > 0$, while the 'run-out' times for each $i \in \aleph_2$ is greater than

λ/A_0, and these are the items with $\alpha_i = 0$. Thus we can say that the allocation is done to equalize the 'run-out' times as much as possible, or equivalently to maximize the minimum 'run-out' time.

2.2.4 The Multiproduct Dynamic Lotsize Problem

Kao [41] and Silver [68] amongst others have studied a finite horizon deterministic production planning problem without backlogging or other constraints with a major (joint) set-up cost A_0 and minor (product) set-up cost A_i. In this case the problem $PZ_i(\alpha_i)$ becomes a version of the dynamic lot size model for which much computational experience has been accumulated and for which efficient routines exist (e.g., Wagner and Whitin [80]). However here the choice of optimal α_i^* is by no means obvious. Of course any feasible choice will still provide a bound. As is demonstrated in Chapter 4, a heuristic choice of α_i to make 'run-out' times as equal as possible without α_i becoming negative gives a bound which is close enough to the optimal to be useful. Further research is needed on optimal choices of α_i.

2.2.5 'Can Order' Policies

Take the joint replenishment problem but now with the demand process as stochastic. With $A_i = 0$ we know, given other assumptions, that a (σ, S) type policy is optimal, Johnson [38] and Kalin [39], where σ is a 'do-not-order' subset of the possible inventory states. However few properties are known about σ and with $A_i > 0$ interest has focused more on (s, c, S) heuristics, [68,23,65,69,75]. Here an item i is ordered up to S_i if its inventory position drops to s_i and at that time any other item j below its 'can order' point c_j is also ordered up to S_j. Ignall [35] has demonstrated that such policies are not optimal, however the suspicion remains that they are fairly good and of course are relatively easy to implement once the parameters (s, c, S) have been calculated. Again, many authors have studied this problem, in particular Silver [69], and Federgruen et al [23]. Different authors have taken different distributions, although all are independent between products, and alternative service level constraints have been tried. Concentrating on shortage costs to control service level rather than explicit constraints, the problem $PZ_i(\alpha_i)$ is now a single product with a fixed cost of $\alpha_i A_0 + A_i$. The lower bound also holds in this case. Again much is known about the optimality of (s, S) type policies for such problems and considerable computational experience has been accumulated [76,77,79]. An exploration of a choice of α_i to equalize expected 'run-out' times as much as possible using periodic policies rather than (s, c, S) policies has been made with very encouraging results, shown in Chapter 5 where it is also shown that even this heuristic choice of α_i within a periodic policy, outperforms the (s, c, S) policies on a wide range of examples.

2.3 The One-Warehouse Multi-Retailer Problem

Consider the production/inventory problem where a warehouse acts as the sole distributor to many retailers. The warehouse gets its supply from an external source. Because there is no constraint on capacity, the warehouse can order any positive amount of any item at any time. Each retailer gets its supply from the warehouse and demands are deterministic.

There is a fixed charge A_0 which must be paid whenever the warehouse replenishes any item, independent of the number or nature of the items replenished. A retailer also incurs a set-up cost A_i whenever retailer i replenishes. There is an echelon holding cost rate at the warehouse for each item and a strictly positive echelon holding cost rate at each retailer for the corresponding item. This distinguishes the one-warehouse multi-retailer problem from the joint replenishment problem where the echelon holding cost rate at the retailers are all zeros.

Since the demand on each retailer is deterministic, the amount ordered by the warehouse at any time is determined on the basis of the needs of the different retailers over a specific time interval. The part of the order meant for a particular retailer is predetermined at the time of order. This initial allocation does not change, because of the deterministic nature of the demand patterns. The warehouse can be imagined as consisting of N slots, if there are N retailers, one slot per retailer. When the warehouse's replenishment is received, the amount is shared amongst the slots, with each slot receiving the preassigned amount for the corresponding retailer. With this interpretation, each retailer can be thought of as representing a distinct item, and the replenishment problem at the warehouse is *a joint replenishment problem.* This joint replenishment problem is characterized by a fixed cost to be paid whenever one or more items are ordered. An item-dependent minor set-up cost is paid only when the item is withdrawn from the warehouse. Thus the one-warehouse multi-retailer problem can be thought of as a special case of the multi-facility joint replenishment problem to be discussed in Chapter 3.

The problem is to schedule replenishments of all items at the warehouse and each item at the corresponding retailer so as to minimize the average set-up plus inventory holding cost in the infinite horizon. As in the joint replenishment problem, we restrict attention to the class of feasible policies characterized by 'last minute' orders for each item. That is, an item is ordered (at the warehouse or retailer) only when the inventory level of the item at the facility is zero. Let a particular feasible policy in this class be Ψ. Such a policy is characterized by:

- the replenishment epochs of item i at the warehouse denoted by $\mathbf{T}_i^0$ and at the retailer i denoted by $\mathbf{T}_{ii}$.
- the corresponding order quantities $Q_i^0(\mathbf{T}_i^0)$ at the warehouse and $Q_i(\mathbf{T}_{ii})$ at the retailer.

Let Ψ_i denote the policy induced on item i by Ψ. We can then write

$$\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_N)$$

and $Z(\Psi)$ the cost of policy Ψ.

2.3.1 Cost Allocation and the Lower Bound

The procedure for deriving the lower bound is exactly the same as for the joint replenishment problem. However, the lower bound result is no longer true when demands are random because of the possibility of risk pooling at the warehouse. Let $Z(\Psi_i, \alpha_i)$ be the cost of the i^{th} facilities-in-series problem if the fractional allocation of the warehouse set-up cost to item i is α_i. For the rest of the analysis, we assume that demands occur at a constant and continuous rate of 2 per unit time for each retailer, and the delivery lead time is zero.

$$PZ_i(\alpha_i) : \qquad Z_i(\alpha_i) = \min_{\Psi_i}\{Z_i(\Psi_i, \alpha_i)$$

and

$$Z(\alpha) = \sum_{i=1}^{N} Z_i(\alpha_i)$$

Schwarz [60] has shown that the optimum policy to this serial problem is periodic with constant cycle times. Thus for any feasible fractional allocations,

$$PZ_i(\alpha_i) : \qquad Z_i(\alpha_i) = \min_{t_i^0, t_i}\{\frac{\alpha_i A_0}{t_i^0} + \frac{A_i}{t_i} + h_i^0 t_i^0 + h_i t_i\} \qquad (2.8)$$

subject to

$$t_i^0 = m_i t_i; \quad m_i \geq 1, \quad \text{integer; for every } i \qquad (2.9)$$

Relax the integrality constraint 2.9 to

$$t_i^0 \geq t_i; \quad \text{for every } i$$

and call the objective function of this later problem $S_i(\alpha_i)$. Let

$$S(\alpha) = \sum_i^N S_i(\alpha_i)$$

We note here that since $S_i(\alpha_i)$ is a lower bound on $Z_i(\alpha_i)$, $S(\alpha)$ is also a lower bound on $Z(\alpha)$ for any feasible α. The value of $S_i(\alpha_i)$ depends on α_i as follows:
When $\alpha_i \geq \frac{h_i^0 A_i}{h_i A_0}$

$$\begin{aligned} t_i^0 &= \sqrt{\frac{\alpha_i A_0}{h_i^0}} \\ t_i &= \sqrt{\frac{A_i}{h_i}} \\ S_i(\alpha_i) &= 2\sqrt{\alpha_i A_0 h_i^0} + 2\sqrt{A_i h_i} \end{aligned}$$

And when $\alpha_i \leq \frac{h_i^0 A_i}{h_i A_0}$

$$\begin{aligned} t_i^0 &= t_i = \sqrt{\frac{\alpha_i A_0 + A_i}{h_i^0 + h_i}} \\ S_i(\alpha_i) &= 2\sqrt{(\alpha_i A_0 + A_i)(h_i^0 + h_i)} \end{aligned}$$

For any feasible choice of α, we can then partition the items into the two sets, $\aleph_1$ and $\aleph_{23}$ below.

$$\begin{aligned} \aleph_1 &= \{i | \alpha_i > \frac{h_i^0 A_i}{h_i A_0}\} \\ \aleph_{23} &= \{i | \alpha_i \leq \frac{h_i^0 A_i}{h_i A_0}\} \end{aligned}$$

and write $S(\alpha)$ in terms of these two sets as follows:

$$S(\alpha) = 2\{\sum_{i \in \aleph_{23}} \sqrt{(\alpha_i A_0 + A_i)(h_i^0 + h_i)} + \sum_{i \in \aleph_1} \sqrt{(\alpha_i A_0 h_i^0} + \sqrt{A_i h_i})\} \qquad (2.10)$$

and the desired cost allocation bound is

$$S(\alpha^*) = \max_\alpha\{S(\alpha) | \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0, \forall i\}$$

which can be solved to determine α^*. The value of α^* which solves this maximization problem can be determined as follows. First note that $S(\alpha)$ is concave in α. After introducing the Lagrange multiplier λ, the $K-T$ conditions are:

$$S_i'(\alpha_i) \leq \lambda \tag{2.11}$$

$$\alpha_i > 0 \Longrightarrow S_i'(\alpha_i) = \lambda \tag{2.12}$$

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{2.13}$$

The first two conditions imply that there are two sets of items, $((\aleph_1 \cup \aleph_{23}) \setminus \aleph_3)$ and $\aleph_3$ such that:

$$i \in ((\aleph_1 \cup \aleph_{23}) \setminus \aleph_3) \Longrightarrow \alpha_i > 0 \quad \text{and} \quad i \in \aleph_3 \Longrightarrow \alpha_i = 0.$$

From the definition of the set $\aleph_1$, the intersection of $\aleph_1$ and $\aleph_3$ is empty, since $\alpha_i > 0$ for $i \in \aleph_1$ if all holding cost and set-up costs are strictly positive. Let $\aleph_2 = \aleph_{23} \setminus \aleph_3$. From 2.11 and 2.12, it can be deduced that;

$$i \in \aleph_1 \Longrightarrow (\frac{A_0 h_i^0}{\alpha_i}) = \lambda^2 \tag{2.14}$$

$$i \in \aleph_2 \Longrightarrow (A_0)^2(\frac{h_i^0 + h_i}{\alpha_i A_0 + A_i}) = \lambda^2 \tag{2.15}$$

$$i \in \aleph_3 \Longrightarrow (A_0)^2(\frac{h_i^0 + h_i}{A_i}) < \lambda^2 \tag{2.16}$$

From 2.14 and 2.15, it is found that:

$$\{\frac{\alpha_i A_0 + A_1}{h_i^0 + h_i} | i \in \aleph_2\} = \{\frac{A_0 \alpha_i}{h_i^0} | i \in \aleph_1\} \overset{def}{=} c \tag{2.17}$$

Observe that c is the square of the replenishment cycle time at the warehouse for those items with $\alpha_i > 0$. That is

$$t_i^0 = \sqrt{c} \qquad \text{for} \qquad i \in \aleph_1 \cup \aleph_2$$

which can be determined from 2.17 using the fact that the sum of the fractional allocations must be unity. The value of c is:

$$c = \frac{A_0 + \sum_{i \in \aleph_2} A_i}{\sum_{i \in \aleph_2}(h_i^0 + h_i) + \sum_{i \in \aleph_1} h_i^0} \tag{2.18}$$

Substituting for c in 2.17 the values of α^* are obtained and are given below:

$$\alpha_i^* = \frac{c(h_i^0 + h_i) - A_i}{A_i^0}; \qquad i \in \aleph_2 \tag{2.19}$$

$$\alpha_i^* = \frac{ch_i^0}{A_i^0} \qquad i \in \aleph_1 \tag{2.20}$$

$$\alpha_i^* = 0 \qquad i \in \aleph_3 \tag{2.21}$$

It can be verified that the set $\aleph_3$ satisfies:

$$\aleph_3 = \{i | \frac{A_i}{h_i^0 + h_i} > c; \ i \in \aleph_{23}\}$$

Three sets are now clearly identified in terms of c.

$$\begin{aligned} \aleph_1 &= \{i | c \geq \frac{A_i}{h_i}\} \\ \aleph_2 &= \{i | \frac{A_i}{h_i} > c > \frac{A_i}{h_i^0 + h_i}\} \\ \aleph_3 &= \{i | c \leq \frac{A_i}{h_i^0 + h_i}\} \end{aligned}$$

The problem of determining α^* is equivalent to the problem of partitioning the items into these three sets. Once the partitioning problem is solved c is obtained from 2.18, while the values of α^* are obtained from 2.19 - 2.21. Substituting α^* for α in 2.10, the cost allocation bound is obtained as given below.

$$S(\alpha^*) = 2\left\{\sqrt{(A_0 + \sum_{i\in\aleph_2} A_i)(H_2 + H_1)} + \sum_{i\in\aleph_3} \sqrt{A_i(h_i^0 + h_i)} + \sum_{i\in\aleph_1} \sqrt{A_i h_i}\right\}$$

where

$$H_2 = \sum_{i\in\aleph_2} (h_i + h_i^0) \qquad \text{and} \quad H_1 = \sum_{i\in\aleph_1} h_i^0.$$

The cost allocation bound above coincides with the lower bound in Roundy [57] where also an $O(N \log N)$ algorithm for partitioning the items into the three sets is given.

2.4 Concluding Remarks

In this chapter we have proposed an extremely simple lower bound for a fairly general class of cost minimization problems in production and inventory control. Two examples have been worked out in detail giving much easier derivations of bounds than in the original papers. We believe these papers [57,36] are pivotal in their respective fields in the sense that, they solve their respective problems for practical purposes. Our approach however has much wider generality. This fact is demonstrated in Chapter 3.

CHAPTER THREE

The Multi-Facility Joint Replenishment Problem

Introduction

The multi-facility joint replenishment problem is a multi-facility production/inventory problem where the facilities (warehouses or production facilities) are arranged as a directed tree network. Each facility is a *sole* distributor to one or more facilities, called the *successor* facilities. A facility without successor facilities is a *sole* distributor of only one item to meet external demand for that item. A characteristic of this problem which needs emphasis is that a facility, except for the *root facility*, gets all its supplies from only one facility, referred to as the *immediate predecessor*; that is, a facility has a unique immediate predecessor in the facilities configuration. The *root facility* is the unique facility without a predecessor. A facility may have any number of successors.

A particular system that fits this description is a distribution network which consists of a national warehouse, regional warehouses, branch warehouses and retail outlets. A retail outlet gets all its supplies from a branch warehouse, while a branch warehouse gets all its supplies from a regional warehouse which in turn gets all its supplies from the national warehouse. A retail outlet may stock any number of items, provided that these items are obtained from one source. We consider the problem with this configuration of facilities, where the facilities without successor facilities, referred to as *end facilities*, each faces a constant and continuous time demand over the infinite horizon. Each facility has a major set-up cost which must be paid at each replenishment regardless of the number of items replenished. With this cost structure, the generalization to the multi-item case does not complicate the problem because a retailer with j items is equivalent to a warehouse that is a *sole* distributor of an item each to j retailers. Each retailer in this case will have the demand and cost characteristics of the corresponding item. Also the conventional holding cost of an item at the retailer-turned warehouse will be equal to the conventional holding cost of the item at the corresponding retailer, that is, the echelon holding cost at the retailer is zero. This implies that it is irrelevant where the stock is kept, either at the warehouse or at the retailer. It can be assumed therefore that the warehouse, representing a retailer with many items, does not hold any item in stock. Because of this relationship, 'items' and 'retailers' will be used interchangeably.

A facility without successors distributes exactly one item. The number of items distributed by a facility with more than one successor facility is the number of its successors which are end facilities, since each such end facility distributes exactly one item. The problem is a joint replenishment problem because a distributor facility incurs a fixed set-up cost whenever it orders or produces at least one of the items it distributes. This replenishment cost is sometimes referred to as the *joint* or *major set-up cost.* This cost, as in the conventional joint replenishment problem, is independent of the nature or the number of items replenished. This implies that a distributor facility has some incentive to coordinate the replenishment of all the items it distributes. A *minor set-up cost* at the distributor facility is the cost incurred by a successor facility when the successor facility withdraws items from the distributor. This minor set-up cost at a distributor facility is a joint or a major set-up cost at the successor facility, when the successor facility has successors of its own. The simplest case of this problem is the one warehouse multi-retailer problem where, as pointed out in Chapter 2, there is only one joint replenishment facility, the warehouse.

A more general case of this problem where a facility may have any number of successors and predecessors has been considered in the literature, [57,86] The solution algorithms are generally complex. Recently Roundy [57] developed an algorithm which solves this general multi-facility production/ inventory problem by solving a series of min-cut max-flow problems. Although the solution therein is guaranteed to be 98% effective in the class of all feasible policies, the complexity of the algorithm is $O(\Phi^4)$, where Φ is the number of paths that can be drawn from an end facility through all its predecessors. In the special case of a directed network of facilities, each with a unique predecessor facility, Φ is of the order of the total number of end facilities. The algorithm for the general multi-facility problem, when the policy space is restricted to nested policies, has a complexity of $f \log f$ when there are f facilities in the network [57]. Included in this class is the one warehouse multi-retailer and multi-item problem [51].

A solution method is proposed here for the multi-facilities joint replenishment problem which is guaranteed to be 98% effective in the class of *all feasible* policies, in the worst case. The complexity of the solution method is $O(dr \log r)$ where r is the number of end facilities and d is the maximum number of predecessors of any one facility. This coincides with the $r \log r$ algorithm for the one-warehouse multi-retailer problem. The facility with the maximum number of predecessors is obviously an end facility. The method derives naturally from a lower bound on the cost of all feasible solutions to the problem. The lower bound is new and belongs to the general class of the *cost allocation bounds* introduced in chapter 2.

The remainder of this chapter is organized as follows: the next section introduces the notation. Section 3 gives the cost allocation method and the lower bound, while section 4 discusses the solution to, and gives an algorithm for solving, the lower bound problem to the multi-facility joint replenishment problem.

3.2 Problem Definition and Notation

Let the set F of facilities be numbered from 1 to f, where $f = |F|$, ($|F|$ denotes the cardinality of F), such that the number assigned to a facility is greater than the number assigned to any of its successors. In addition the facilities without successors will be numbered consecutively from 1 to $r = |R|$, where R is the set of the facilities without successors. Each facility $i \in R$ faces a constant deterministic demand for an item replenishes only this item, also numbered item i. A facility may have any number of successors but only one predecessor. Facility f is the only facility without a predecessor, and is called the *root facility.* Because a facility has a unique predecessor, the path from any facility to any of its predecessors is unique. Let p_j and s_j denote the immediate predecessor (a single facility) and the set of immediate successors of facility j respectively, while P_j and S_j are the set of *all* predecessors plus facility j, and the set of all successors plus facility j, respectively. It is important to note that *each of P_j and S_j includes facility j.* We shall use R_j to denote the intersection of R and S_j, that is, the set of retailers or items that are replenished from facility j. Let A_j be the set-up cost associated with facility j, and h_{ij} the echelon holding cost of item i at facility j, for each item $i \in R_j$. The concept of echelon holding cost was discussed in Chapter 2.

A facility with no predecessor either is a production facility, or replenishes from an outside supplier. In the multi-facility joint replenishment problem considered here, there is only one facility, f, without a predecessor. The set-up cost associated with a facility must be paid whenever a replenishment or production takes place at the facility. The echelon holding cost, h_{ij}, is the cost rate per unit time per unit of echelon stock of item i at the facility j. The individual unit variable cost is assumed to be constant throughout the horizon, and need not be considered since backlogging is not allowed. The average procurement cost should therefore not depend on any particular policy.

The problem is to schedule replenishments of all items at all the facilities so as to minimize the average set-up plus inventory holding cost over the infinite horizon.

Let Ψ be a feasible policy defined by

- the replenishment times of item i at facility j denoted by the sequence $\mathbf{T}_{ij}$, for every $i \in R_j$ and every j.
- the corresponding order quantities $Q_{ij}(\mathbf{T}_{ij})$ of item i at facility j, at times in T_{ij}, $i \in R_j$.

Let Ψ_i denote the replenishment schedule induced by Ψ on item i. We can then write

$$\Psi = (\Psi_1, \cdots, \Psi_r)$$

Let $Z(\Psi)$ be the average cost of policy Ψ.

3.3 Cost Allocation and the Lower Bound

The set-up cost of facility j is allocated amongst all the items it produces or replenishes, that is, all $i \in R_j$. Let α_{ij} be the fractional allocation from facility j to item i, $i \in R_j$. It is required that the sum of all fractional allocations, derived from a single facility be unity.

The tree network of facilities is now decomposed into r facilities-in-series such that the facilities-in-series corresponding to each $i \in R$, consists only of P_i. The holding cost rates for an item at the facilities-in-series are the same as the holding cost rates for the item at the corresponding facilities before the allocation scheme. The set-up cost at facility j in the facilities-in-series corresponding to item i is $\alpha_{ij}A_j$. Each facilities-in-series represents an inventory problem of its own; that is, to determine a feasible replenishment policy at each facility in the facilities-in-series which minimizes the average set-up plus inventory holding cost over the infinite horizon.

Let

$$\alpha = (\alpha_1, \cdots, \alpha_r)$$

where

$$\alpha_i = (\alpha_{ij} | j \in P_i).$$

The allocation vector α is admissible if

$$\sum_{i \in R_j} \alpha_{ij} = 1; \quad \alpha_{ij} \geq 0 \qquad \forall j \in F \text{ and } i \in R_j \tag{3.1}$$

Using the policy Ψ for the r facilities-in-series, we observe that the replenishment times associated with the i^{th} facilities-in-series problem, Ψ_i, are feasible for this problem. Let the cost of this facilities-in-series problem be denoted by $Z_i(\Psi_i, \alpha_i)$, if α_i is the fractional allocation vector to item i. First note that the sequence of orders implied by Ψ is the same as the sequences implied by Ψ_i, applied to the i^{th} facilities-in-series problem. So the inventory patterns are the same, and hence the inventory holding costs are also equal in both cases. Suppose τ is a replenishment time of a particular facility, say j. The set-up cost of facility j, A_j is paid at time τ when Ψ is applied to the general problem. Let $E(\tau)$ be the set of items in R_j that are replenished at time τ by facility j. When Ψ_i is applied individually to each facilities-in-series problem i, $i \in R_j$, the total set-up cost incurred at facility j is given by:

$$A_j \sum_{i \in E(\tau)} \alpha_{ij} \leq A_j \sum_{i \in R_j} \alpha_{ij} = A_j$$

It follows that, for all admissible cost allocation vectors α_i,

$$Z(\Psi) \geq \sum_{i \in R} Z_i(\Psi_i, \alpha_i) \tag{3.2}$$

Thus the cost allocation method gives a lower bound on the average cost of all feasible policies to the multi-facility joint replenishment problem. Let

$$Z_i^*(\alpha_i) = \min_{\Psi_i} Z_i(\Psi_i, \alpha_i) \tag{3.3}$$

This is the minimum average cost over all feasible policies to the i^{th} facilities-in-series problem.

We remark that a lower bound on the general problem would be obtained if only the set-up cost of any one facility is allocated. However, allocating the set-up costs of all facilities as is done here decomposes the problem to units that can be solved.

We can rewrite this section by replacing the word 'average' wherever it occurs by the word 'total', and 'infinite' by 'finite', and the result is still true. That is, the lower bound holds true for both finite and infinite horizon problems with any deterministic demand pattern. It also applies to non-deterministic demands if items are all distinct.

3.3.1 The Lower Bound Problem

A policy is *product-nested* if each item must be replenished at a facility whenever the immediate predecessor of the facility replenishes the same item. Product-nested policies should be distinguished carefully from the nested policies in the literature, which we might call *facility-nested* and require that a facility replenishes whenever its predecessor replenishes. Facility-nestedness implies product-nestedness but not vice versa. Consider periodic policies where the intervals between replenishments at a facility, and for each item are constant. The replenishment times for item i at facility j will be, for example, of the form

$$\mathbf{T}_{ij} = \{0, c_{ij}, 2c_{ij}, \ldots\}$$

The replenishment cycle time of item i at facility j in this case is c_{ij}.

We consider periodic and product-nested policies. Power-of-two policies [57,36] which require that the ratio of the replenishment cycle time of an item at a facility to the replenishment cycle time of the same item at a successor facility be an integer power of two, are special cases of the periodic and product-nested policies. The lower bound problem is developed by first restricting attention to only periodic and product-nested policies. In this class of policies, each facility j has a fixed cycle time of T_j, and a fixed replenishment cycle time T_{ij} for each item $i \in R_j$. The minimum cost equation is developed for this class of policies. It is then shown that when the integrality requirement of product-nestedness is relaxed, a lower bound on the cost of *all feasible policies* is obtained.

Let $s(i,j) = P_i \cap s_j$, that is, the immediate successor of facility j which is also a predecessor of i. Let P be the following problem:

$$P: \qquad Z_P = \min\{\sum_{j \in F}(\frac{A_j}{T_j} + \sum_{i \in R_j} h_{ij}T_{ij})\} \tag{3.4}$$

subject to

$$T_{ij} = m_{ij}T_{is(i,j)};\ m_{ij} \geq 1, \text{ integer}; \forall j \in F, \text{ and } i \in R_j \tag{3.5}$$

$$T_{ij} \geq T_j > 0; \forall j, \text{ and } i \in R_j \tag{3.6}$$

This problem P is the multi-facility joint replenishment problem restricted to product-nested policies. Constraint 3.5 is the product-nested constraint. Replace 3.5 by the following relaxation:

$$T_{ij} \geq T_{is(i,j)} > 0\ \forall j, \text{ and } i \in R_j \tag{3.7}$$

This relaxes the integrality requirement of product-nestedness. Denote this relaxed problem by RP and its objective function by Z_{RP}. Because constraint 3.7 is a relaxation of 3.5, we have $Z_P \geq Z_{RP}$. The right hand side of equation 3.4, that is the objective function of this problem, can be re-written as

$$Z_P(T) = \sum_{j\in F}[\sum_{i\in R_j}\frac{\alpha_{ij}A_j}{T_j} + \sum_{i\in R_j} h_{ij}T_{ij}] = \sum_{i\in R}\sum_{j\in P_i}[\frac{\alpha_{ij}A_j}{T_j} + h_{ij}T_{ij}] \tag{3.8}$$

Note that this function is independent of α whenever α is admissible, that is α satisfies equation 3.1. Because $T_{ij} \geq T_j$, and T_j is only in the denominator of the objective function of RP, we can get a lower bound on RP by replacing T_j in 3.8 by T_{ij} and deleting the constraint 3.6. Let the resulting problem be denoted by $RRP(\alpha)$, which is stated below.

$$\begin{aligned} RRP(\alpha): \quad Z_{RRP(\alpha)} &= \min \sum_{i\in R}\sum_{j\in P_i}[\frac{\alpha_{ij}A_j}{T_{ij}} + h_{ij}T_{ij}] \\ \text{subject to } &3.7 \end{aligned}$$

This problem is clearly dependent on α. We can now write down the following relationships.

$$Z_P \geq Z_{RP} \geq Z_{RRP(\alpha)} \tag{3.9}$$

Let $RS_i(\alpha_i)$ be defined as:

$$RS_i(\alpha_i): \quad Z_{RS_i(\alpha)} = \min \sum_{j\in P_i}[\frac{\alpha_{ij}A_j}{T_{ij}} + h_{ij}T_{ij}] \tag{3.10}$$

subject to

$$T_{ij} \geq T_{is(i,j)} > 0; \quad j \in P_i. \tag{3.11}$$

With this definition,

$$Z_{RRP(\alpha)} = \sum_{i\in R} Z_{RS_i(\alpha_i)}.$$

The notation $RS_i(\alpha_i)$ is meant to remind us that $RS_i(\alpha_i)$ is a relaxation of the following periodic facilities-in-series problem:

$$S_i(\alpha_i): \quad Z_{S_i(\alpha)} = \min \sum_{j\in P_i}[\frac{\alpha_{ij}A_j}{T_{ij}} + h_{ij}T_{ij}] \tag{3.12}$$

subject to

$$T_{ij} = m_{ij}T_{is(i,j)} > 0; \quad m_{ij} \geq 1, \text{integer}, \ j \in P_i. \tag{3.13}$$

The relaxation stems from the fact that 3.11 is a relaxation of 3.13. The desired result is stated after the following proposition.

Proposition 3.1

$$Z_{RS_i(\alpha_i)} \leq Z_i^*(\alpha_i)$$

where $Z_i^*(\alpha_i)$ is defined in 3.3.

Proof:

In [56] Roundy showed that when the integrality requirements on the set of constraints 3.13 are relaxed for the pure assembly system, the resulting problem provides a lower bound on the cost of all feasible solutions to the assembly problem. The facilities-in-series problem is a special case of the pure assembly system. So, that result holds for facilities-in-series problem. That is

$$Z_i^*(\alpha_i) \geq Z_{RS_i}(\alpha_i).$$

■

The following relationship is an obvious consequence of the definitions of $P, RP, RRP(\alpha)$ and proposition 3.1.

$$\sum_{i \in R} Z_i^*(\alpha_i) \geq Z_{RRP(\alpha)} = \sum_{i \in R} Z_{RS_i(\alpha_i)}. \tag{3.14}$$

The above relationship is true for any admissible vector α, that is $Z_{RRP(\alpha)}$ is a lower bound on the cost of any feasible solution to the original problem, and also on problem RP, for any admissible α. In particular the maximum of $Z_{RRP(\alpha)}$ over all admissible α satisfying 3.1 is our desired *lower bound.*

$$RRP^* : \quad Z_{RRP(\alpha^*)} = \max_{\alpha} Z_{RRP(\alpha)} \tag{3.15}$$

subject to

$$\sum_{i \in R_j} \alpha_{ij} = 1; \quad \alpha_{ij} \geq 0 \qquad \forall j \text{ and } i \in R_j$$

In section 4 we provide an algorithm for this problem which gives T and α such that the following holds.

$$\sum_{\{i|i\in R_j, T_{ij}=\min_k T_{kj}\}} \alpha_{ij} = 1; \quad \alpha_{ij} \geq 0. \tag{3.16}$$

Denote by α', the particular α which satisfies equation 3.16.

First we establish that $Z_{RRP(\alpha')} = Z_{RP} = Z_{RRP(\alpha^*)}$ and then discusss how α^*, or equivalently α' can be determined.

Theorem 3.2 *Suppose α' satisfies equation 3.16, then any solution to problem $RRP(\alpha')$ is feasible for problem RP and $Z_{RRP(\alpha')} = Z_{RP} = Z_{RRP(\alpha^*)}$ where α^* is defined in 3.15. Thus Z_{RP} is a lower bound on the cost of any feasible solution to the original problem.*

Proof

Let T' be a solution of $RRP(\alpha')$. Let $T'_j = \min_k\{T'_{kj}|k \in R_j\}$. The cycle times T' satisfy equations 3.6 and 3.7 and are therefore feasible for problem RP. So we have $Z_{RRP(\alpha')} = Z_{RP}$. But as shown in equation 3.9, $Z_{RRP(\alpha)} \leq Z_{RP}$ for any admissible α. The maximum of $Z_{RRP(\alpha)}$ over all admissible α cannot be greater than Z_{RP}. Thus $Z_{RRP(\alpha')} = Z_{RRP(\alpha^*)}$. The last statement of the theorem follows since $Z_{RRP(\alpha^*)}$ is a lower bound on the cost of any feasible solution to the original problem.

■

The implication is that if we solve the lower bound problem and hence can determine the optimal values of α, we will have solved problem RP. We now consider the implication of this theorem in the context of the solution procedure we shall propose for providing α' and by implication solving the lower bound problem and RP. Purely as a motivational device and anticipating somewhat the results below, we shall consider $RRP(\alpha)$ further in the context of 3.16. We know from the above that the solution to $RS_i(\alpha_i)$ for any particular $i \in R$ of the r facilities-in-series problems gives a lower bound on the cost of an optimum policy for the facilities-in-series problem.

The policy however, may not be feasible. In [57] Roundy has shown that when the constant cycle times T_{ij} are rounded to a multiple of 2 of some common base period, then the cost penalty incurred is less than 6%, or less than 2% if the base period is optimized, of the cost of the optimum policy. In addition the resulting power-of-two policy is feasible. If we take the problem of figure 2.1 and create the facilities-in-series problems as in figure 2.2 for the values of α shown, we could get the solution shown in figure 3.1 below.

The numbers to the right of each facility-in-series are the values of the fractional cost allocations from the corresponding facility, while the the numbers to the left are the unrounded cycle times or solution to each $RS_i(\alpha_i)$, $i \in R$. If these cycle times are rounded to a common base period, say 2, then we could have the picture shown in figure 3.2.

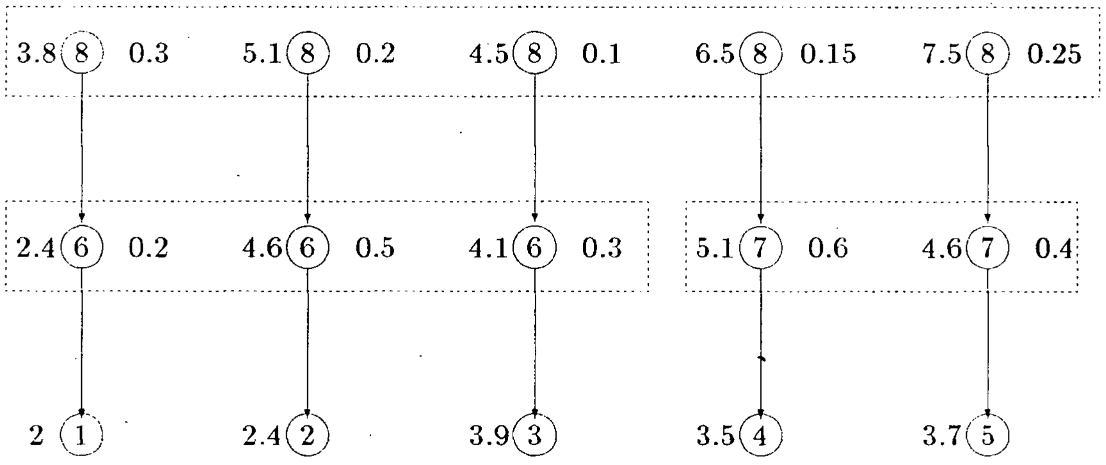

Figure 3.1: A Typical Solution for a Given Cost Allocation

Although this is a feasible solution, the accounting of set-up costs is not done

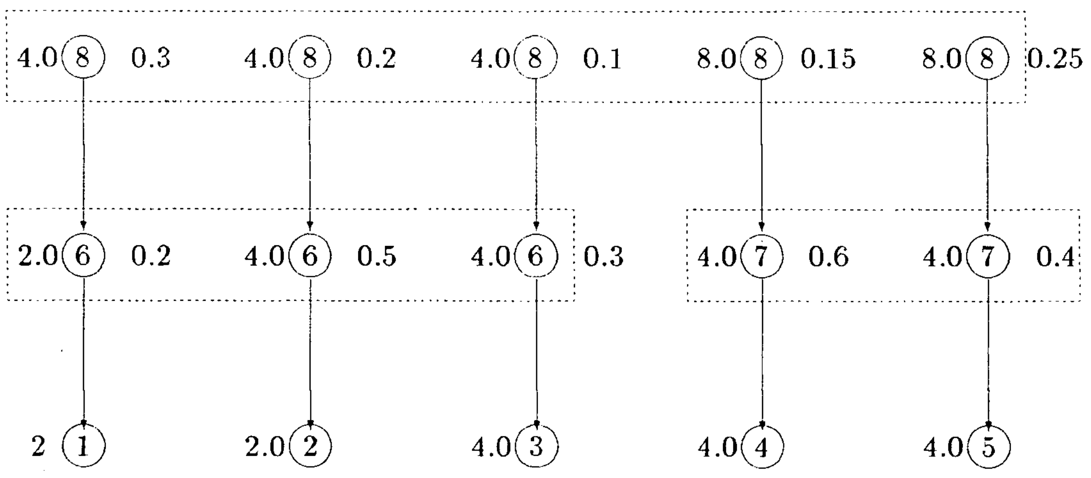

Figure 3.2: 'Rounded' Power-of-two Solution for Figure 3.1

correctly. For example, consider the second replenishment that takes place at time 4.0, after the initial replenishment at time 0 at facility 8, the total set-up cost incurred at the facility is $0.6A_8$, which does not correctly account for the actual set-up cost of A_8. What we need for proper accounting is that those facilities-in-series that drive the joint facility cycle times have their values of α that sum to 1, and hence account correctly for the facility set-up cost. Accounting correctly for the set-up cost is exactly the statement of equation 3.16. For example suppose $\alpha_{18} = 0.5$, $\alpha_{28} = 0.3$ and $\alpha_{38} = 0.2$, the full set-up cost of facility 8 will always be paid whenever a set-up cost takes place at the facility. This correctly accounts for the set-up cost at facility 8. In the sequel we shall provide an algorithm for choosing α which gives correct accounting, and simultaneously solves $RS_i(\alpha_i)$, for all $i \in R$. Such a solution ensures that all facilities-in-series that share a common facility j and have a non-zero allocation $\alpha_{ij} > 0$, $i \in R_j$, will have an identical cycle time. Written formally, we shall have that:

$$T_{ij} > T_{kj} \quad \text{implies } \alpha_{ij} = 0. \tag{3.17}$$

This is obviously equivalent to 3.16 and implies the following:

$$T_{ij} = T_{kj} \quad \text{whenever } \alpha_{ij} > 0 \text{ and } \alpha_{kj} > 0 \tag{3.18}$$

Any such choice of α will be termed *correctly accountable*. The main objective of this chapter is to provide an algorithm which solves $RS_i(\alpha_i)$, $i \in R$ and *simultaneously* provides *correctly accountable* α. Before describing this algorithm, we shall give some useful properties of any optimum solution to a facilities-in-series problem.

3.3.2 The Facilities-in-Series Problem

One algorithm which gives the optimum solution to a facilities-in-series problem is the minimum violators algorithm [56], described in Appendix A. Let j be the topmost facility, that is, the facility without a predecessor and let $n \in s_j \cap P_i$. Facility n is the immediate successor of j in this series system. Suppose the set-up cost $\alpha_{in}A_n$ of facility n in the i^{th} facilities-in-series is strictly positive, that is $\alpha_{in} > 0$ and $A_n > 0$. Let $\alpha_{ij} = 0$. In the optimum solution provided by the minimum violators algorithm, facility n and j have the same cycle time. Suppose we wish to solve this i^{th} facilities-in-series problem such that the maximum index with a positive set-up cost is n. The set-up cost for a facility k $k \in S_j \cap P_i$ is $\alpha_{ik}A_k$. Let this problem be called $PS_i(n,j)$.

The algorithm results in a partition of the set of facilities in the series system into series *clusters*. The topmost *cluster* for problem $PS_i(n,n)$ is $CL(i,n,n)$ while it is $CL(i,n,j)$ for $PS_i(n,j)$. The clusters for $PS_i(n,j)$ consists of $CL(i,m,m)$ where $m \in P_i \cap S_n$ and m is the root of the cluster $CL(i,m,m)$, and a topmost cluster $CL(i,n,j)$. Note that $CL(i,n,j)$ is different from $CL(i,n,n)$; the former being the topmost cluster for problem $PS_i(n,j)$ and the latter being the topmost cluster for problem $PS_i(n,n)$. The optimality conditions for these clusters are given in Roundy [56], and repeated here. For any subset B, define

$$T[B] = \frac{\sum_{j\in B}\alpha_{ij}A_j}{\sum_{j\in B}h_{ij}}$$

The necessary and sufficient conditions for optimality are:

1. $T[CL(i,m,m)] > T[CL(i,k,k)]$, implies $m > k$

Also $T[CL(i,n,j)] > T[CL(i,m,m)]$ for $m \in S_n \setminus CL(i,n,j)$.

2. $T[CL(i,m,m) \cap S_k] \geq T[CL(i,m,m)]$, for any $k \in CL(i,m,m)$.

These two conditions will be referred to as the *optimum series conditions.* The following is a direct consequence of the second optimum series condition.

3. If $l \in S_n \cap P_i$, $T[CL(i,l,l)] < T[CL(i,n,j)]$ implies that $l \notin CL(i,n,j)$.

Another condition that can be deduced from the first two conditions above is:

4. If $l \in CL(i,n,j)$, then $CL(i,l,l) \subseteq CL(i,n,j)$

We now show the implications of these conditions in the context of the solution procedure for the lower bound problem. Suppose we wish to solve problem $PS_i(j,j)$ parametrically as follows. We first assume that α_{ij} is set equal to zero, that is the set-up cost at facility j is zero. Thus we have problem $PS_i(n,j)$. (Recall $n \in s_j \cap P_i$). The algorithm, for problem $PS_i(n,j)$, gives us the topmost cluster $CL(i,n,j)$, with

$$T[CL(i,n,j)] = \frac{\sum_{k\in CL(i,n,j)\setminus\{n,j\}}\alpha_{ik}A_k + \alpha_{in}A_n + 0}{\sum_{k\in CL(i,n,j)\setminus\{n,j\}}h_{ik} + h_{in} + h_{ij}} \overset{def}{=} L_n \tag{3.19}$$

For $l \in S_n \cap CL(i,n,j)$ let

$$U_l = \frac{\sum_{k\in CL(i,l,l)}\alpha_{ik}A_k}{\sum_{k\in CL((i,l,l)}h_{ik}}$$

Note that $U_l = T[CL(i,l,l)]$. Suppose we decide to raise the set-up cost of facility j from its zero level by increasing α_{ij} to a positive value. This process is referred to as *allocation* in the sequel, and it is the general step in both the algorithms for the facilities-in-series described in this section and for the general problem in the next section. Because facility j is now assigned a set-up cost $\alpha_{ij}A_j$, the resulting optimal topmost cluster is no longer $CL(i,n,j)$ but $CL(i,j,j)$. Let this new cluster be referred to as the *current cluster.* To determine membership of this current cluster, replace the zero in the numerator of 3.19 by $\alpha_{ij}A_j$ and examine the relationship of $T[CL(i,j,j)]$ and U_l, $l \in S_n \cap CL(i,n,j)$. So long as the new value of $T[CL(i,j,j)]$ is less than or equal to the minimum of U_l, $l \in S_n \cap CL(i,n,j)$, all the facilities in $CL(i,n,j)$ will remain in the new cluster $CL(i,j,j)$ and the optimum series conditions are satisfied. Suppose, on the other hand, the value of $T[CL(i,j,j)]$ is greater than some U_l. This violates the third condition above and by implication the second optimum series condition. To rectify this situation, we detach $CL(i,l,l)$ from $CL(i,j,j)$. Because of condition 4 above, if $CL(i,l,l)$ is the first such cluster to be deleted, then $CL(i,l,l) = S_l \cap CL(i,n,j)$. By this detachment, the feasible clusters become the clusters obtained when solving $PS_i(l,l)$, plus the new cluster $CL(i,j,j)$. The detachment also preserves the optimum series conditions because if $T[CL(i,n,j)]$ was greater than $T[CL(i,k,k)]$ for some $k \in S_n$, the new cluster $CL(i,j,j)$ has a cycle time at least as large as $T[CL(i,n,j)]$, and then also larger than $T[CL(i,k,k)]$. The detached cluster $CL(i,l,l)$ and the newly created cluster $CL(i,j,j)$ clearly satisfy condition 1 of the optimum series conditions. The second condition is also satisfied because as the set-up cost A_j is being allocated, any violation of the second condition is immediately rectified by detaching the portion of the cluster causing the violation. We can now call U_l, $l \in S_n \cap CL(i,n,j)$, the *upper point* and $CL(i,l,l)$ the *upper cluster* of facility l, which is also the topmost cluster for problem $P_i(l,l)$, when allocating the set-up cost of facility j. The upper point is the point at which facility l with the upper cluster $CL(i,l,l)$ drops out of the current cluster and the optimum clusters of $PS_i(l,l)$ become members of the optimum clusters for $PS_i(j,j)$. We also refer to L_n as the *lower point* and $CL(i,n,j)$ as the *lower cluster* of facility n. The lower point is thus the point at which we can begin to consider facility n with the lower cluster of n as a member of the current cluster; in this case cluster $CL(i,j,j)$, when the amount of set-up cost allocated from facility j is raised just above zero. The main point to note here is that when allocating the set-up cost A_j the immediate successor of j, that is, n in this case has two sets of clusters associated with it, the lower cluster $CL(i,n,j)$ and the upper cluster $CL(i,n,n)$ while each facility $l \in S_n \cap CL(i,n,j)$ has one cluster associated with it, that is, the upper cluster $CL(i,l,l)$. The lower cluster must always contain j. This also implies that whenever the set-up cost of a facility is being allocated, the cluster that can be added to the current cluster *must* contain the *immediate successor* of the facility whose set-up cost is being allocated, and the cluster that can be deleted *must* have been a member of the current cluster, that is, the cluster which contains the current facility (the one whose cost is being allocated). An algorithm based on this allocation method can be used to solve the general facilities-in-series problem if we choose the sequence of subproblems to solve in the correct order. In particular if we have a facilities-in-series problem from i to f, once the problem from i to f is solved with the set-up cost of f temporarily set to zero, the final solution is obtained by allocating the set-up cost $\alpha_{if}A_f$.

An Algorithm for the Facilities-in-Series Problem

```
Statement of the Algorithm for $PS_i(f,f)$

Initialize: $j \leftarrow p_i$; $l \leftarrow i$

Step 1: Solve $PS_i(l,j)$ with set-up cost of all $P_l$
        temporarily set to zero.
        Allocate $\alpha_{il}A_l$ to topmost cluster $CL(i,l,j)$.
        to determine the new set of optimum clusters for $PS_i(l,j)$.
        $l \leftarrow p_l$. Repeat step 1 until $l = j$.

Step 2: $j \leftarrow p_j$; $l \leftarrow i$.
        Go to step 1 if $j \leq f$.
```

The algorithm for the general problem determines how much of each set-up cost to allocate to each facilities-in-series, so that series by series, the optimum series conditions are satisfied and also the allocated fractions are correctly accountable. That is, all those receiving an allocation must have the minimum cycle time before allocation and equal cycle time after allocation. This would suggest that if we only allocate enough to the cluster that has the minimum cycle time to bring its cycle time up to that of the next minimum, then allocate again to all those at this minimum until their cycle time becomes equal to the next, and repeat until no more allocation can be made, then we would have satisfied the criterion of correct accontability. Also using the allocation method described earlier in this section, the allocation satisfies the optimum series conditions for each series. By theorem 3.2, the problem is then solved.

We now consider the problem of simultaneously allocating set-up cost to more than one serial cluster, so that the optimum series conditions are not violated and the allocation is correctly accountable. As an example, suppose after allocating the set-up cost A_n to four facilities-in-series, we have the lower clusters $CL(1,n,j)$, $CL(2,n,j)$, $CL(3,k,j)$, $CL(4,l,j)$, with k and l in S_n. Suppose we have

$$T[CL(4,l,j)] > T[CL(1,n,j)] = T[CL(2,n,j)] = T[CL(3,k,j)].$$

This situation implies that $\alpha_{1n} > 0$, $\alpha_{2n} > 0$, $\alpha_{3n} = \alpha_{4n} = 0$. For a cluster $CL(3,k,j)$ to be a lower or topmost cluster, the set-up cost of the predecessors of k including n are zeroes. We say that $CL(3,k,j)$ has no allocation from n. Notice that although $CL(3,k,j)$ has no allocation from facility n, it has the same cycle time as the facilities that do receive an allocation. This is a special case when, after allocating the set-up cost of facility n, the resulting cycle time coincides with the cycle time of a series without any allocation. We now wish to allocate A_j to these four lower series clusters so that the optimum series conditions are satisfied and the resulting allocation is correctly accountable. We start by allocating to a lower cluster with minimum cycle time. In this case there is a tie; $CL(1,n,j)$, $CL(2,n,j)$ and $CL(3,k,j)$. Because of this, whenever an allocation is made to any one of these clusters, we can *always* share this allocation amongst the relevant clusters with the same cycle time so that their resulting cycle times after the allocation are all equal. This condition will be referred to as keeping the clusters with the same cycle time *in balance*. Suppose we allocate enough of the set-up cost A_j to the three clusters to raise their cycle time to that of $CL(4,l,j)$. At this point all the four clusters are in balance and will remain so in any further allocation. Further the clusters are now $CL(1,j,j)$, $CL(2,j,j)$, $CL(3,j,j)$ and $CL(4,j,j)$. We will at any point take note of any upper cluster to delete as our allocation gets past the cycle time of the cluster. Because all clusters with an allocation are kept in balance, the fractional allocations to the clusters are correctly accountable, since the clusters receiving allocation have all the minimum cycle time. This is equivalent to correct accountability as in 3.16.

Suppose the maximum index of a facility whose set-up cost has been allocated is l. Let us say that those serial clusters that have received an allocation from A_l are similar because they have equal cycle time by our allocation regime. Thus cluster $CL(i,l,j)$ and cluster $CL(m,l,j)$ are *similar* if $T[CL(i,l,j)] = T[CL(m,l,j)] = T_l$. Similar clusters are those clusters whose cycle times are equal and must, by necessity, be kept in balance during any later allocation. If this is the case we loose no information if we combine all similar serial clusters before making any allocation. We can now formally define a combination of serial clusters: Let

$$CL(l,j) = \cup\{CL(i,l,j)\,|\, i \in R_l, T[CL(i,l,j)] = T_l\}.$$

Any reference to a cluster, without qualification, in the subsequent discussion is a reference to a combination of similar serial clusters.

3.4 Solution to the General Problem

We shall solve the general problem by solving subproblems defined on subtrees of the original problem. Let the path from facility i to facility j consist of facilities in $(P_i \cap S_j)$. Let Π_{nj} be the subtree with root node n plus a path from n to j, $j \in P_n$. For the subtree Π_{nj}, the set-up cost of all facilities on the path from p_n to j are temporarily set to zero, while the set-up costs at all facilities $k, k \in S_n$ are unchanged. The echelon holding costs of items $i, i \in R_n$ at facility n are also temporarily replaced by h'_{in}, defined below.

$$h'_{in} = \sum_{k \in P_n \cap S_j} h_{ik}. \tag{3.20}$$

The reason for replacing the echelon holding cost at facility n by h'_{in} when solving the problem on Π_{nj} will become clear in what follows.

The cost allocation method applied to Π_{nj} defines $|R_n|$ facilities-in-series problems. The problem on the subtree Π_{nj} is said to be 'solved' when we have partitioned each of its facilities-in-series problems into serial clusters which, series by series, satisfy the *optimum series conditions* and we have allocated the set-up cost of all facilities $k \in S_n$

between series such that the α's are *correctly accountable*. Ultimately, we wish to 'solve' the tree Π_{ff}. Thus we consider the equivalent $|R_n|$ series problem from i to n with the echelon holding cost at facility n in the i^{th} facilities-in-series replaced by h'_{in}.

A formal definition of problem $RP_{\Pi_{nj}}$ on Π_{nj} is as follows:

$$RP_{\Pi_{nj}}: \qquad \min \sum_{i \in R_n} \sum_{m \in P_i \cap \Pi_{nj}} \left[\frac{\alpha_{im} A_m}{T_m} + h_{im} T_{im}\right] \tag{3.21}$$

subject to

$$T_{im} \geq T_{is(i,m)}; \quad \forall m \in \Pi_{nj} \quad \& \quad i \in R_m; \tag{3.22}$$

$$T_{im} \geq T_m \geq 0; \quad \forall m \in \Pi_{nj} \quad \& \quad i \in R_m. \tag{3.23}$$

And a formal statement of the desired solution is as follows: find clusters which partition the $|R_n|$ facilities-in-series and a feasible allocation

$$\alpha_{ik} \geq 0; \quad \sum_{i \in R_k} \alpha_{ik} = 1; \quad i \in R_n, \quad k \in S_n$$

such that the *optimal series conditions* hold, and the allocation is is *correctly accountable*, condition 3.16 is satisfied. Theorem 3.2 tells us that an optimum solution to this problem does in fact possess these properties. Recall that correct accountability is defined by:

$$T_{km} > T_{im} \quad \text{implies } \alpha_{km} = 0$$

We can deduce from these that whenever correct accountability is achieved, the following must also be satisfied:

1: $\sum_{i \in \{k | T_{km} = \min_j T_{jm}\}} \alpha_{im} = 1$

2: $\alpha_{im} > 0$ and $\alpha_{km} > 0$ imply $T_{im} = T_{km}$.

We know by Theorem 3.2 that the above conditions, that is, correct accountability and the optimum series conditions, are sufficient for the optimality of any α. The algorithm below is designed such that at each allocation step, the optimum series conditions are satisfied and moreover the α's resulting from the allocation are correctly accountable. The algorithm ensures that the relevant information is available when allocating A_f.

3.4.1 The Algorithm

This algorithm provides correctly accountable α's which also satisfy the optimum series conditions for each subtree Π_{nj} in $O(d_n|R_n|\log|R_n|)$ time, where d_n is the number of facilities on the longest path from an item to root facility, which in this case is facility n. The algorithm is constructive. It consists of solving a sequence of problems on subtrees Π_{nj}, $j \in P_n$. The set-up cost A_n is allocated sequentially to the lower clusters to raise the minimum cycle time as much as possible. If at any time during the allocation, the cycle time of the current cluster becomes greater than that of an upper cluster which is a subset of the current cluster, that upper cluster is deleted from the current cluster. This allocation rule is justified on the ground that series-by-series, it preserves the optimum series conditions and the resulting fractional allocations are correctly accountable since all clusters with the same cycle time before allocation are kept in balance during allocation. By Theorem 3.2 this allocation rule solves $RP_{\Pi_{nj}}$.

A simple way to carry out this allocation procedure is to rank and merge all the existing lower clusters $CL(l,j)$ where $l \in S_n$ and the upper clusters, that is those clusters whose roots are in $CL(l,j)$, and to perform the allocation step to be described later in this section.

Initially the current cluster $CL(n,j)$ only consists of facilities $(P_n \setminus P_j) \cup \{j\}$. A list of the the relevant lower and upper clusters, ranked in ascending order of their cycle times, is kept in the list of *unchecked clusters* denoted by $UP(n,j)$. The set-up cost A_n is allocated first to the unchecked cluster with the smallest cycle time. Members of this cluster are added to cluster $CL(n,j)$. The set-up cost associated with this cluster is added to the set-up cost A_n to determine the new set-up cost of $CL(n,j)$. The holding cost of the new cluster is the holding cost of the old cluster. Let τ_{nj} denote $T[CL(n,j)]$. The resulting τ_{nj} is then compared with the next point on the list $UP(n,j)$. If τ_{nj} is greater than or equal to the next point and the point is a lower point, the cluster associated with it is added to the current cluster, if it is strictly greater and the point is an upper point, the cluster associated with the point is deleted from the current cluster. In either case τ_{nj} is updated. Deletion of a cluster consists of simply removing the set-up cost and the holding cost of the cluster to be deleted from the set-up cost and holding cost respectively of the current cluster, while addition of a cluster consists of adding the set-up cost and the holding cost of the cluster to be added to the set-up cost and holding cost respectively of the current cluster.

When allocating, a record of all the upper clusters deleted from the current cluster, that is the clusters in the *immediate* successor set of the current cluster, is kept in a list known as the *active points* $AP(n,j)$. It will be seen that $AP(l,j)$ $l \in S_n$ is a record of the feasible clusters for Π_{nj}.

The algorithm consists of two major steps.

Step 1: Determination of the unchecked points.

Step 2: Allocation of the set-up cost to determine the active points, while simultaneously updating the unchecked points.

Let $UP(k,j)$ be the ordered list of unchecked points, representing the clusters when solving the problem on Π_{kj} and when allocating A_k. The points are quadruples. The first element is the cycle time of the cluster, the second element is the first index of the cluster, that is, the root of the cluster, and the third element is the second index of the cluster. The fourth element is an indicator variable indicating whether the point (cluster) is a lower or an upper point (cluster).

Determination of unchecked points:

Let $\oplus$ be a notation for 'rank and merge'. The following are the formulae for determining the unchecked points relevant to problem Π_{kj}.

$$
\begin{aligned}
UP(i,j) &= ((\frac{A_i}{h'_{ij}}, i, j, lower); (\frac{A_i}{h_{ii}}, i, i, upper)); \quad i \in R; \quad j = p_i; \\
UP(i,j) &= (\frac{A_i}{h'_{ij}}, i, j, lower); \quad i \in R \quad j > p_i \\
UP(n,j) &= \oplus\{UP(k,j);\, k \in s_n\} \oplus \{AP(n,l);\, n \notin R, \\
&\qquad l \in s_j \cap P_n\}
\end{aligned}
$$

since $s_j \cap P_n = \emptyset$ for $n = j$

$$UP(n,n) = \oplus UP(k,n); \quad k \in s_n.$$

Allocation of set-up cost A_n to $UP(n,j)$, $n \notin R$:

Initially set τ_{nj} to a very large number. Compare it with the first element of the next point in the ordered list $UP(n,j)$. Let a point whose second and third elements are l and k repectively be denoted by U_{lk}. Let the next point in the ordered list of points of $UP(n,j)$ be U_{lk}. B and E are initially zero. If $\tau_{nj} \geq \tau_{lk}$, and it is a lower point, do the following: $B_{nj} \leftarrow A_n + B_{lk}$, $E_{nj} \leftarrow E_{nj} + E_{lk}$, $CL(n,j) \leftarrow CL(n,j) \cup CL(l,k)$ otherwise if it is an upper point, and the inequality is strict do the following: $B_{nj} \leftarrow B_{nj} - B_{lk}$, $E_{nj} \leftarrow E_{nj} - E_{lk}$, $CL(n,j) \leftarrow CL(n,j) \setminus CL(l,k)$, $AP(n,j) \leftarrow AP(n,j) \cup \{U_{lk}\}$. In either case $UP(n,j) \leftarrow (UP(n,j) \setminus U_{lk})$. Let $\tau_{nj} \leftarrow \frac{B_{nj}}{E_{nj}}$ and $T'_l \leftarrow \tau_{nj}$ for all $l \in CL(n,j)$. Compare τ_{nj} with the next unchecked point of $UP(n,j)$ and repeat the above update until $\tau_{nj} < \tau_{lk}$, where U_{lk} is the next unchecked point of $UP(n,j)$. The current point is denoted by U_{nj}. Include the current point in $UP(n,j)$ as the first unchecked *lower* point.

If $n = j$, after the above update, check all *remaining* unchecked points in $UP(n,n)$. If the next unchecked point is a lower point, include it in $AP(n,n)$ as an *upper* point and delete it from $UP(n,n)$. If the point is an upper point, simply delete it from $UP(n,n)$. Repeat until $UP(n,n)$ is empty.

This later step is necessary to ensure that all serial clusters of the $CL(i,l,n)$ $l \in s_n$ become $CL(i,n,n)$ after allocating the set-up cost A_n even for those without an allocation.

Statement of the Algorithm

```
Initialize: $E_{ij} \leftarrow h'_{ij}$ and $B_{ij} \leftarrow A_i$; $\forall j$ and $i \in R_j$
    $UP(n,j) \leftarrow \emptyset$,   $AP(n,j) \leftarrow \emptyset$,   $CL(n,j) \leftarrow \{n\}$,   $n \notin R; j \in P_n$
    $AP(i,i) \leftarrow \emptyset$,   $i \in R$
    Determine $UP(i,j)$, and set $CL(i,j) \leftarrow \{i\}$, $i \in R$, $j \in P_i$.

Main Step: For $n = |R| + 1$ step 1 until $n = f$
    Determine $UP(n,j)$; $j \in P_n$. Allocate $A_n$ to determine $AP(n,j)$ and update
    $UP(n,j)$.
```

The cycle times from the algorithm are given by

$$T_k^* = \sqrt{T_k'}; \qquad k \in F$$

To get a policy which is 94% effective, we simply choose a base period and round the cycle times to the nearest power-of-two of the base period in such a way that the order of the cycle times are preserved. That is if $T_k^* > T_l^*$, then the cycle times should be rounded such that if T_k denotes the rounded cycle time of facility k then $T_k \geq T_l$. To obtain a solution which is 98% effective, we optimize over the base period. Details of this are in Roundy [57].

Complexity of the Algorithm

For illustrative purpose we assume a symmetric tree. If the number of facilities in P_i is d for $i \in R$, we say there are d stages in the facilities network. Let F_k be the set of facilities in stage k and let the number of successors for each facility in F_k be n_k. With this notation $F_1 \equiv R$ and $n_d n_{d-1} \dots n_2 = r$. For each $j \in F_2$, the ranking done to determine unchecked points is $n_2 \log n_2$. This is done for the $n_d n_{d-1} \dots n_3$ facilities in F_2. The required amount of work for this initial ranking is thus

$$n_d n_{d-1} \dots n_3 n_2 \log n_2 = r \log n_2$$

The amount of computational work involved is r which is no more than in ranking as long as $n_2 > 1$. The ranking is done $d-1$ times. So the total amount of work in ranking involving $j \in F_2$ is

$$(d-1) r \log n_2$$

For $j \in F_3$ we need to merge already ranked points of F_2. The amount of work for the merging operation for each $j \in F_3$ is

$$n_3 n_2 \log n_3$$

Again the amount of computational work is no more than the ranking as long as $n_3 > 1$. The ranking is done for the $n_d n_{d-1} \dots n_4$ facilities in F_3, $(d-2)$ times in all. The total amount of work involved is thus

$$(d-2) n_d n_{d-1} .. n_2 \log n_3 = (d-2) r \log n_3$$

In general the amount of ranking for facilities in stage k is

$$(d-k+1) r \log n_k$$

The computational work is again, no more than the ranking as long as $n_k > 1$. The total amount of work involved in ranking for the algorithm is thus

$$\sum_{k=2}^{d} (d-k+1) r \log n_k \leq d r \log r$$

The complexity of the algorithm is therefore

$$O(dr \log r).$$

The case where some or all $n_k = 1$, $k = 2, \ldots, d$, is rather too special to be of general interest here.

3.4.2 Example 3.1

Let

$A_1 = 4$, $A_2 = 5$, $A_3 = 7$, $A_4 = 2$, $A_5 = 20$, $A_6 = 8$, $A_7 = 6$ and $A_8 = 10$.

$$\begin{array}{lll} h_{11} = .14 & h_{16} = .16 & h_{18} = .05 \\ h_{22} = .25 & h_{26} = .20 & h_{28} = .10 \\ h_{33} = .15 & h_{36} = .10 & h_{38} = .05 \\ h_{44} = .30 & h_{47} = .20 & h_{48} = .15 \\ h_{55} = .20 & h_{57} = .15 & h_{58} = .10 \end{array}$$

Initialize

A * will be used to denote a lower point.

$$\begin{aligned} UP(1,6) &= \{4/.14\,(1,1);\ 4/.3 *\ (1,6)\} \\ UP(2,6) &= \{5/.25\,(2,2);\ 5/.45 *\ (2,6)\} \\ UP(3,6) &= \{7/.15\,(3,3);\ 7/.25 *\ (3,6)\} \\ UP(4,7) &= \{2/.30\,(4,4);\ 2/.50 *\ (4,7)\} \\ UP(5,7) &= \{20/.20\,(5,5);\ 20/.35 *\ (5,7)\} \\ UP(1,8) &= \{4/.35 *\ (1,8)\} \\ UP(2,8) &= \{5/.55 *\ (2,8)\} \\ UP(3,8) &= \{7/.30 *\ (3,8)\} \\ UP(4,8) &= \{2/.65 *\ (4,8)\} \\ UP(5,8) &= \{20/.45 *\ (5,8)\} \end{aligned}$$

Set $CL(n,j) \leftarrow n$ for all relevant n and j.

$$\begin{aligned} UP(6,6) &= UP(1,6) \oplus UP(2,6) \oplus UP(3,6) \\ &= 5/.45 * (2,6);\ 4/.3 * (1,6);\ 5/.25\,(2,2); \\ &\quad 7/.25 * (3,6);\ 4/.14\,(1,1);\ 7/.15\,(3,3). \end{aligned}$$

Allocate $A_6 = 8$ to obtain

$$\begin{aligned} AP(6,6) &= \{5/.25\,(2,2);\ 12/.5\,(6,6);\ 7/.25\,(3,6)\} \\ CL(6,6) &= \{6,1\} \end{aligned}$$

$$\begin{aligned} UP(6,8) &= UP(1,8) \oplus UP(2,8) \oplus UP(3,8) \oplus AP(6,6) \\ &= \{5/.55 * (2,8);\ 4/.35 * (1,8);\ 5/.25\,(2,2); \\ &\quad 7/.3 * (3,8);\ 12/.5\,(6,6);\ 7/.25\,(3,6)\,\} \end{aligned}$$

Allocate $A_6 = 8$ to obtain

$$\begin{aligned} AP(6,8) &= \emptyset \\ UP(6,8) &= \{17/.9 * (6,8);\ 5/.25\,(2,2);\ 7/.3 * (3,8); \\ &\quad 12/.5\,(6,6);\ 7/.25\,(3,6)\,\} \\ CL(6,8) &= \{6,1,2\} \end{aligned}$$

$$\begin{aligned} [UP(7,7) &= UP(4,7) \oplus UP(5,7) \\ &\quad \{2/.5 * (4,7);\ 2/.3\,(4,4);\ 20/.35 * (5,7);\ 20/.2\,(5,5)\} \end{aligned}$$

Allocate $A_7 = 6$ to obtain

$$\begin{aligned} AP(7,7) &= \{2/.3\,(4,4);\ 6/.2\,(7,7);\ 20/.35\,(5,7)\} \\ CL(7,7) &= \{\,7\,\} \end{aligned}$$

$$\begin{aligned} UP(7,8) &= UP(4,8) \oplus UP(5,8) \oplus AP(7,7) \\ &= \{2/.65 * (4,8);\ 2/.3\,(4,4);\ 6/.2\,(7,7);\ 20/.35\,(5,7); \\ &\quad 20/.45 * (5,8)\,\} \end{aligned}$$

Allocate $A_7 = 6$ to obtain

$$\begin{aligned} AP(7,8) &= \{\,2/.3\,(4,4)\,\} \\ UP(7,8) &= \{6/.35 * (7,8);\ 6/.2\,(7,7);\ 20/.45 * (5,8); \\ &\quad 20/.35\,(5,7)\,\} \\ CL(7,8) &= \{\,7\,\} \end{aligned}$$

$$\begin{aligned} UP(8,8) &= UP(6,8) \oplus UP(7,8) \\ &= \{6/.35 * (7,8);\ 17/.90 * (6,8);\ 5/.25\,(2,2); \\ &\quad 7/.3 * (3,8);\ 12/.50\,(6,6);\ 7/.25\,(3,6); \\ &\quad 6/.20\,(7,7);\ 20/.45 * (5,8);\ 20/.35\,(5,7);\ \} \end{aligned}$$

Allocate $A_8 = 10$ to obtain

$$\begin{aligned} AP(8,8) &= \{\,5/.25\,(2,2);\ 12/.50\,(6,6);\ 7/.25\,(3,6); \\ &\quad 16/.55\,(8,8);\ 20/.45\,(5,8)\,\}. \\ CL(8,8) &= \{8,7\} \end{aligned}$$

The resulting clusters and their cycle times are recorded in $AP(6,8)$, $AP(7,8)$ and $AP(8,8)$. The clusters are obtained starting from the 'topmost' that is facility 8, and working down through the facility network, such that a facility is examined only after its predecessors have been examined. Thus

CLUSTER	facilities in Cluster	Cycle Time
	$AP(8,8)$	
$CL(2,2)$	2	$\sqrt{5/0.25} = 4.47$
$CL(6,6)$	1,6	$\sqrt{12/0.50} = 4.90$
$CL(3,6)$	3	$\sqrt{7/0.25} = 5.29$
$CL(8,8)$	7,8	$\sqrt{16/0.55} = 5.39$
$CL(5,8)$	5	$\sqrt{20/0.45} = 6.67$
	$AP(7,8)$	
$CL(4,4)$	4	$\sqrt{2/0.30} = 2.58$
	$AP(6,8)$	
— — —	— — —	— — —

The cycle times at each facility and for each item at a facility are obtained directly from the table above.

$$
\begin{aligned}
T_{11} &= T_{16} = T_{26} = T_1 = T_6 = 4.90 \\
T_{22} &= T_2 = 4.47 \\
T_{33} &= T_{36} = T_3 = 5.29 \\
T_{44} &= T_4 = 2.58 \\
T_{55} &= T_{57} = T_{58} = 6.67 \\
T_{18} &= T_{28} = T_{38} = T_{48} = T_{47} = T_7 = T_8 = 5.39
\end{aligned}
$$

The determination of the fractional allocation is not a necessary part of the algorithm but it can be done as an illustration to check that the solution obtained corresponds to correctly accountable cost allocation and also satisfies the optimum series conditions.

We can determine the cost allocations as follows.

Allocation at facility 6

First note that $\alpha_{36} = 0$ since $T_{36} > T_6$. The following gives α_{16} and α_{26}

$$\frac{A_1 + \alpha_{16}A_6}{h'_{16}} = \frac{\alpha_{26}A_6}{h_{26}}; \quad \text{and} \quad \alpha_{16} + \alpha_{26} = 1.$$

This gives us $\alpha_{16} = 0.4$ and $\alpha_{26} = 0.6$.

Allocation at facility 7

Since $T_{57} > T_7$ $\alpha_{57} = 0$ and hence $\alpha_{47} = 1$.

Allocation at facility 8

Since $T_{58} > T_8$, $\alpha_{58} = 0$. The other allocations can be obtained using the equations below.

$$\frac{\alpha_{18}A_8}{h_{18}} = \frac{\alpha_{28}A_8}{h_{28}} = \frac{\alpha_{38}A_8}{h_{38}} = \frac{A_7 + \alpha_{48}A_8}{h_{47} + h_{48}}$$

and

$$\alpha_{18} + \alpha_{28} + \alpha_{38} + \alpha_{48} = 1$$

With these two equations we have $\alpha_{18} = 0.145$, $\alpha_{28} = 0.290$, $\alpha_{38} = 0.145$, and $\alpha_{48} = 0.420$.

It can be checked from figure 3.3 that these allocations (on the right of circles) are correctly accountable and that the cycle times (on the left of circles) satisfy the optimum series conditions.

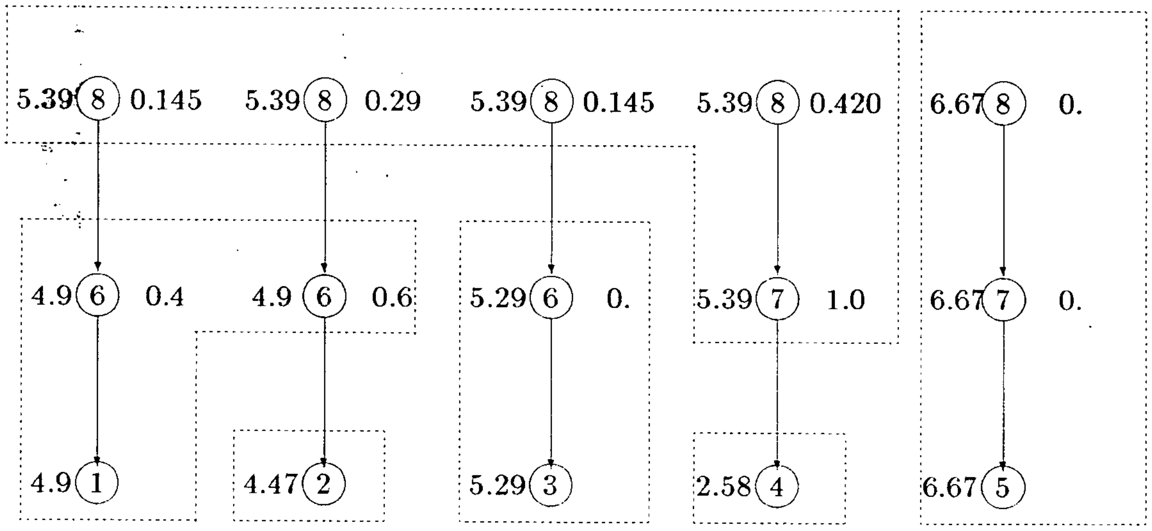

Figure 3.3: Optimum Clusters for Example 3.1

CHAPTER FOUR

The Multi-Product Dynamic Lot-Size Problem

Introduction

This chapter considers the determination of lot sizes in a multiple product environment. A fixed set-up or order cost A_0 is incurred whenever any product is ordered or produced, independently of the number or type of products; and an extra cost A_i is added if product i is included in the joint order. The demand for each item is discrete in time and known over a given time horizon H. Linear holding costs are charged on the end of period inventories and backlogging is not permitted. It is assumed that the variable purchase cost for each product is constant through out the horizon, so that the total purchase cost of any item for total demand in the horizon is a constant independent of the replenishment policy since backlogging is not permitted at the end of the horizon. Quantity discounts are not considered here. The problem is to schedule the replenishment of each item so that the total set-up and inventory holding cost is minimized over the horizon.

This problem has been studied very widely in the literature with several optimal solutions proposed. The dynamic programming solutions for this problem are complex. The solutions by Zangwill and Veinott described in Chapter 1 are generalizations of this problem. When specialized to this problem Zangwill's method [86] is exponential in the number of products, while Veinott's [78] and Kalymon's [40] solutions are exponential in the number of time periods. Solutions which are exponential in the number of products have also been proposed by Kao [41] and Silver [66]. These solutions are not useful for practical problems which usually involve many items and many time periods. Efforts have therefore shifted to the development of heuristic solutions. Unfortunately, though these heuristics are relatively simple when compared to the optimum solutions they still have two major disadvantages. First, they generally depend on the Wagner-Whitin dynamic programming solution to the single item dynamic lot size problem. Second, it is not known how good these heuristics are. Some of these heuristics are described in Chapter 1. Silver [67] reports on a simple one-pass heuristic with a three product example. Because a typical practical problem involves many items, and managers find it difficult to understand dynamic programmin solutions, these heuristics are not desirable from a practical standpoint. The aim of this chapter is to present heuristics which overcome the two disadvantages of earlier heuristics. We introduce a new lower bound (the cost allocation bound) whose computational complexity varies as the square of the number of time periods and linearly as the number of products. In addition we present generalizations of some of the very simple heuristics for the one product dynamic lot size problem. Thus for any data set we can establish easily a bound on how far the optimum is from the heuristic solution.

The heuristics that we generalize are; the Silver-Meal [54], the part-period balancing (PPB) and a variant of it proposed by Bitran, Magnanti and Yanasse [8] which we denote as BMY heuristic. It is known that the Silver-Meal heuristic is arbitrarily bad in the worst case but performs very well in practical problems. We show by a series of examples that the generalization we propose also performs very well. It is, of course, not questionable that this generalization can also be arbitrarily bad in the worst case. The Silver-Meal heuristic is known to be widely used in practice [54] because of its simplicity and the very highly intuitive base. So, we hope that this generalization will also capture the attention of practitioners who have joint replenishment problems to solve.

The part-period balancing heuristic is known to perform well in practice and it is simple, but it has an effectiveness ratio of 1/3. That is the ratio of the cost of an optimum solution to the cost of the heuristic solution cannot be less than 1/3 in the worst case. We shall show that when this heuristic is generalized to the multi-product dynamic lot size problem, the effectiveness of the generalized heuristic cannot be less than 1/3 either. The other heuristic (BMY) reported in [8] which is a simple variant of the part-period balancing heuristic is shown to have an effectiveness ratio of 1/2. We show that when this heuristic is generalized, the generalization does not affect its effectiveness.

These generalized heuristics are simple one pass heuristics, which are linear in the number of products and the length of the horizon. The lower bound run time is linear in the number of products and the same as the Wagner-Whitin algorithm in the number of time periods. We thus have heuristics and also a posteriori guarantee of the heuristics' performances on all data set. In two cases, the genaralized part period balancing and a variant of it, the BMY heuristic, we in fact have the effectiveness ratios. For the generalized Silver-Meal heuristic we report results over a wide variety of cost and demand scenarios for a thirty product, twenty-four period problem, which is a much larger problem than any reported result in the literature. We emphasize that the heuristics can be used very easily for *any* number of products. The heuristic is usually between 95% and 100% of the lower bound and hence maybe even closer to the optimum.

This chapter is divided into four sections. In section 2, we formally define the problem and describe the lower bound. In section 3, we describe the Silver-Meal heuristic and the generalization. The result on a series of examples are also presented.

In section 4, we present the two other heuristics, the generalized PPB (GPPB) and the generalized BMY (GBMY), and establish their effectiveness ratios.

4.2 The Lower Bound

Let N be the number of items and H the time horizon. There is a linear inventory holding cost h_i per unit per period for item i. Backlogging is not allowed. The major reorder cost is A_0 while the item-specific reorder cost is A_i for item i. Let d_{it}, X_{it} and I_{it} be the demand, order quantity and end of period inventory respectively for item i in period t. The holding cost rate can be allowed to vary from period to period but it is assumed constant for expository simplicity. Let $\delta(x) = 0$ if $x = 0$ and $\delta(x) = 1$ if $x > 0$. The problem of minimizing the total order cost plus inventory holding cost over the horizon can be stated as follows:

$$P: \qquad Z_P = \min \sum_{t=1}^{H} \left\{ A_0 \delta[\sum_{i=1}^{N} X_{it}] + \sum_{i=1}^{N} [A_i \delta(X_{it}) + h_i I_{it}] \right\}$$

subject to

$$X_{it} + I_{it-1} - d_{it} - I_{it} = 0; \quad \forall i \quad \text{and} \quad t = 1, 2, \ldots, H \tag{4.1}$$

$$X_{it} \geq 0, \quad I_{it} \geq 0, \quad d_{it} \geq 0. \tag{4.2}$$

Without loss of generality, we can assume that $I_{i0} = 0$. Let us say that the fractional allocation α is admissible if

$$\sum_{i=1}^{t} \alpha_{it} = 1; \quad \alpha_{it} \geq 0; \quad \forall t$$

Let $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iH})$ For any admissible cost allocation we can write the following single item dynamic lot-size problem:

$$P_i(\alpha_i): \qquad Z_{P_i(\alpha_i)} = \min\{(\alpha_{it} A_0 + A_i)\delta(X_{it}) + h_i I_{it}\}$$

subject to 4.1 and 4.2

Since

$$\delta(\sum_{i=1}^{N} X_{it}) \geq \sum_{i=1}^{N} \delta(X_{it})$$

it must be true that

$$Z_P \geq \sum_{i=1}^{N} Z_{P_i(\alpha_i)}$$

Note that $\sum_{i=1}^{N} Z_{P_i(\alpha_i)}$ is the objective function value of a decomposition problem, $P(\alpha)$, of problem P. The lower bound expressed in equation 4.2 is true for any admissible α. It is reasonable therefore to choose the α which maximizes $Z_{P(\alpha)}$. The problem of making such a choice is still rather complex. The problem $P_i(\alpha_i)$ is a single item dynamic lot size problem. Many heuristics exist for solving this problem. In Chapter 2, it was observed in the case of continuous review, infinite horizon stationary demands and cost parameters, and minimization of long run avearge cost problem, that the choice of α was made so that the minimum cycle time was as large as possible. We use a similar strategy for choosing α in this case. We select α such that when each problem $P_i(\alpha_i)$ is solved then in any interval, the minimum 'run-out' time for any item is as large as possible. Having chosen α we can then determine $Z_{P_i(\alpha_i)}$ using the Wagner-Whitin's algorithm for the dynamic lot size problem.

In the following sections, we describe heuristics solutions to the multi-item dynamic lot-size problem and the corresponding methods of choosing the fractional allocations.

4.3 Generalized Silver-Meal Heuristic

First we review the Silver-Meal heuristic for the single item dynamic lot size problem. Let

$$SM_{it}(A_i) = \frac{A_i + h_i \sum_{j=L_i}^{t} (j - L_i) d_{ij}}{t - L_i + 1}$$

where L_i is the time of the last set-up for item i. The heuristic recommends a set-up at time t if t is the first time after L_i such that $SM_{it}(A_i) > SM_{it-1}(A_i)$

Multiple Items

The following is the Silver-Meal heuristic generalized to the multiple item environment. Let T be a set-up epoch. At any time $t \geq T$, products are in one of two sets, set R contains all items for which it has already been decided to include in the joint replenishment at time T, that is we have $L_i = T$. Set C, the 'continuation' set contains all other items, that is those for which $L_i < T$.

For the products not in R, we have the choice of reordering them at time T, or continuing to satisfy demands from the previous set-up. The total cost of continuing to satisfy demand between T and $t > T$ from the previous set-up L_i is given by

$$h_i \sum_{j=T}^{t} (j - L_i) d_{ij}$$

while the cost of ordering at time T and continuing to $t > T$ is

$$A_i + h_i \sum_{j=T}^{t} (j - T) d_{ij}.$$

So these products would rather not join the reorder set until

$$h_i(T - L_i) \sum_{j=T}^{t} d_{ij} > A_i \tag{4.3}$$

The Heuristic

Initialize: Put all items in R and let C be the empty set. Let $T = t = 1$.

Step 1 Calculate $SM_{it}(A_i)$ for each $i \in C$. If $SM_{it}(A_i) > SM_{it-1}(A_i)$ and the condition of equation 4.3 holds, move i from C to R and set $L_i \leftarrow T$.

Step 2: Calculate $SM_{it}(A_i)$ for each $i \in R$. If $SM_{it}(A_i) > SM_{it-1}(A_i)$, we allocate as much of the major set-up cost A_0 to item i to equalize the two quantities. The allocation, Δ_i, necessary to do this is given by

$$SM_{it}(A_i + \Delta_i) = SM_{it-1}(A_i + \Delta_i) \tag{4.4}$$

Set $t \leftarrow t+1$ and repeat steps 1 and 2 until $\sum_i \Delta_i \geq A_0$. In this case go to step 3.

Step 3 Set $T \leftarrow t$ and reorder at time T. Let $R = \{i | \Delta_i > 0\}$ and $L_i = T$ if $i \in R$. Return to step 1 until $t = H$.

This algorithm is easy to program and clearly runs in time proportional to N and H.

An Example

Three items and five periods problem with the following parameter values.
$A_0 = 39$, $A_1 = A_2 = A_3 = 20$, $h_1 = h_2 = h_3 = 1$. The demands are shown in the table below.

	Time				
	1	1	3	4	5
item 1	10	6	20	10	10
item 2	5	4	12	16	10
item 3	10	10	6	2	7

The Algorithm

Initially $R = \{1,2,3\}$; $C = \emptyset$; $L_1 = L_2 = L_3 = 1$.

Time $t=1$	$SM_{11} = 20$	$SM_{21} = 20$	$SM_{31} = 20$
	$\Delta_1 = 0$	$\Delta_2 = 0$	$\Delta_3 = 0$
Time $t=2$	$SM_{12} = 13$	$SM_{22} = 12$	$SM_{32} = 15$
	$\Delta_1 = 0$	$\Delta_2 = 0$	$\Delta_3 = 0$
Time $t=3$	$SM_{13} = 22$	$SM_{23} = 16$	$SM_{33} = 14$
	$\Delta_1 = 54$	$\Delta_2 = 24$	$\Delta_3 = 0$

This is because $SM_{12}(20+54) = SM_{13}(20+54) = 40$ and $SM_{22}(20+24) = SM_{23}(20+24) = 24$

As $\Delta_1 + \Delta_2 > 39$ a reorder is scheduled at $T = 3$ and we reset $L_1 = L_2 = 3$ and $R = \{1,2\}$, $C = \{3\}$.

Time 4 :

$$SM_{14} = 15 \quad SM_{24} = 18 \quad SM_{34} = 12$$

$$\Delta_1 = 0 \qquad \Delta_2 = 0 \qquad \Delta_3 = 0$$

Time 5: $SM_{35} > SM_{34}$ and $(3-1)(6+2+7) > 20$ so item 3 joins R, is reordered at $T = 3$, and set $L_3 = 3$.

$$SM_{15} = 50/3 \quad SM_{25} = 56/3 \quad SM_{35} = 12$$

$$\Delta_1 = 10 \qquad \Delta_2 = 4 \qquad \Delta_3 = 6$$

No reorder is scheduled at time 5 because the major set-up cost is not exhausted, that is

$$\Delta_1 + \Delta_2 + \Delta_3 = 20 < 39 = A_0$$

So the heuristic gives a reorder schedule of

	Time				
	1	2	3	4	5
item 1	16	0	40	0	0
item 2	9	0	38	0	0
item 3	20	0	15	0	0

and a total cost of $2(39) + 6(20) + 1(102) = 300$.

The lower bound

We left the lower bound without specifying a choice of α_{it}. From the heuristic, at a period when we decide to reorder we have allocated $\Delta_i > 0$ to $i \in R$ and $\Delta_i = 0$ to $i \in C$. Let

$$\alpha_{it} = \frac{\Delta_i}{\sum_{i \in R} \Delta_i}$$

for each $i \in R$ and for each period t, from the period immediately following the last joint reorder period to period T. If the last joint reorder is in period 1, then α_{it} is as above from $t = 1$ to $t = T$.

The logic behind this rule is that this allocation lengthens the joint reorder interval as much as possible. When $\sum_i \Delta_i = 0$, which may be the case when $t = H$, we simply allocate A_0 equally amongst all the items. Returning to the example we have the values of α_{it}.

	Time				
	1	2	3	4	5
item 1	54/78	54/78	54/78	10/20	10/20
item 2	54/78	24/78	54/78	4/20	4/20
item 3	0	0	0	6/20	6/20

The Wagner-Whitin solution to each in turn is

	Time					Cost
	1	2	3	4	5	
item 1	16	0	40	0	0	130
item 2	9	0	38	0	0	104
item 3	20	0	15	0	0	66

which gives a total cost of 300. The heuristic gives the optimum solution in this case.

We now present the results for realistically sized problems.

4.4 Computational Experience

We take the following problem. Horizon=24 periods and the number of items equals 30. Without loss of generality we take $h_i = 1$ for all items. The item-specific set-up costs A_i are uniformly distributed on $[30, 50]$. The demand is uniformly distributed on $[a, b]$ as shown in Table 4.1. Each cell of the table gives the average performance for

	A_0				
demand	0	100	200	300	400
$[20,20]$	0	0	0	0	0.2
$[20,30]$	0.3	0.2	0.1	0.2	0.4
$[15,25]$	0.5	0.6	0.9	1.8	2.5
$[15,30]$	1.3	0.6	1.0	1.4	3.0
$[10,20]$	3.8	3.5	3.5	2.4	2.4
$[5,10]$	1.5	2.7	2.8	2.8	3.3
$[5,15]$	4.0	2.7	3.3	5.2	6.4
$[10,30]$	4.4	2.7	3.0	4.3	5.3

Table 4.1: Heuristic Performance

five different runs. The performance measure used is

$$\frac{\text{heuristic cost} - \text{lower bound cost}}{\text{lower bound cost}} \times 100$$

The first row is for a constant demand. The first column is essentially the Silver-Meal heuristic, although the possibility of the rule in equation 4.3 changes things slightly. We see that over a wide data set, seldom is the heuristic more than 5% away from the lower bound and hence closer to the unknown optimum. We see however that when the demand variability is proportionally high, that is $(b-a)/a = 2$, as in the last two rows and the joint cost A_0 is high, the performance of either the heuristic or the bound or both deteriorates.

We have presented a simple heuristic, easy to code and calculate, which is essentially an extension of the Silver-Meal heuristic for single products. The key importance of this heuristic compared with other heuristics is that a posteriori performance *guarantee* is given for the particular data set being considered. A variety of experiments on varied data sets indicate performance of usually better than a 5% error. In view of the inaccuracies of demand and cost data typically available to the production planner, a 5% gap from the lower bound but not necessarily from the optimum is a relatively small price to pay for the simplicity of the algorithm.

4.5 Other Heuristics

The heuristics to be generalized in this section are the part period balancing heuristic and a variant of it introduced in Bitran et al. [8] to be referred to as BMY heuristic. The main purpose of this section is to show that the effectiveness ratios of these generalized heuristics are the same as the effectiveness ratios of the corresponding single item heuristics.

4.5.1 PPB and GPPB Heuristics

The part period balancing heuristic minimizes the absolute difference between the set-up cost and the inventory holding cost until the next set-up. Let $H_i(L_i, t)$ be the cummulative inventory holding cost from period L_i to $t-1$ given that a set-up occurs at period L_i. Let t' be the smallest time greater than L_i such that $H_i(L_i, t') - A_i > 0$. The part period balancing heuristic sets up in period t' if

$$H_i(t_i, t') - A < A - H_i(t_i, t' - 1) \tag{4.5}$$

and in period $t' - 1$ otherwise.

Proposition 4.1 *The effectiveness of the PPB heuristic is at least 1/3.*

Proof

The proof of this proposition is given in [8] and is repeated here. The maximum inventory holding cost between any two set-ups is $2A_i$. So the maximum total cost incurred from one set-up to the period before the next set-up is $3A_i$. If the two consecutive set-ups of the heuristic are t_i and t_i' the cost incurred in an optimum policy between t_i and t_i' cannot be less than A_i, being either a set-up cost or inventory holding cost. Thus if C_H is the cost of the heuristic solution and C_0 is the cost of the optimum solution, the effectiveness of the PPB heuristic is given by

$$\frac{C_O}{C_H} \geq \frac{A_i}{3A_i} = \frac{1}{3}$$

■

For the GPPB heuristic, let L_i be the last time item i was ordered. Let G_t be defined as the set of items each of whose accumulated holding cost at time t exceeds its item-dependent set-up cost, that is

$$G_t = \{i | H_i(L_i, t) > A_i\}$$

and let t' be the first time after L_i such that the following holds

$$\sum_{i \in G_{t'}} (H_i(L_i, t') - A_i) - A_0 > 0 \tag{4.6}$$

A set-up is made at t' if

$$\sum_{i \in G_{t'}} (H_i(L_i, t') - A_i) - A_0 < \sum_{i \in G_{t'}} (A_i - H_i(L_i, t' - 1)) + A_0$$

and at time $t' - 1$ otherwise. The items reordered are those in $G_{t'}$.

Proposition 4.2 *The effectiveness of the GPPB heuristic is at least 1/3.*

Proof

Suppose two consecutive set-ups are in periods t' and t''. The maximum cost incurred for those items ordered at t'' between periods $t' + 1$ and t'' inclusive is the set-up cost at period t'', which consists of A_0 plus the the set-up cost of the items ordered at time t''; and the inventory holding cost in these periods. But the inventory cost cannot be greater than twice the set-up cost at period t''. Thus if we denote the set-up cost by A'' and the cost given by the heuristic in the interval by C''_H

$$C''_H \leq 3A'' + \text{Inventory cost of items } \textit{not} \text{ ordered at } t''$$

If the optimum cost in this interval is C''_O then

$$C''_O \geq A'' + \text{Inventory cost of items } \textit{not} \text{ ordered at } t''$$

Note that the minimum cost for all those items ordered by the heuristic at t'' is A''. If an item not ordered by the heuristic is ordered in the optimum solution, the set-up cost incurred for that item is greater than the holding cost for the item in this interval. Also if an optimum set-up occurs between the last heuristic set-up and the end of the horizon, the cost incurred due to this set-up cannot be less than the holding cost incurred by the heuristic. Thus the effectiveness of the heuristic is given by

$$\frac{C''_O}{C''_H} \geq \frac{A'' + \text{Inventory cost of items } \textit{not} \text{ ordered at} t''}{3A'' + \text{Inventory cost of items } \textit{not} \text{ ordered at } t''} \geq \frac{A''}{3A''} = \frac{1}{3}$$

■

4.5.2 BMY and GBMY heuristics

BMY requires a set-up to be made at time $t' - 1$ if

$$H_i(t_i, t') > A_i \quad \text{and} \quad H_i(t_i, t'-1) \leq A_i$$

where t' is as defined in 4.5.

Proposition 4.3 *The effectiveness of the BMY heuristic is at least 1/2.*

Proof

The proof is similar to the proof of proposition 4.1. Here the inventory holding cost between two consecutive replenishments is bounded by the set-up cost.

■

For the GBMY heuristic, a set-up is made in period $t' - 1$ where t' is as defined in 4.6 above.

Proposition 4.4 *The effectiveness of the GBMY heuristic is at least 1/2.*

Proof

The proof is similar to that of proposition 4.2. The inventory holding cost between two consecutive replenishments is bounded by the cost of the second set-up in the interval.

■

4.6 Conclusion

We have presented very simple heuristics which are easy to implement and easy to code. The first is a generalization of Silver–Meal heuristic. This heuristic is shown to perform very well on a series of examples. The second is a generalization of the part period balancing while the third is also a generalization of a variant of the part period balancing first presented in [8]. The effectiveness ratios of these latter heuristics are shown to be the same as for their corresponding single item heuristics. It is also simple to derive lower bounds for these heuristics. The major contribution of this chapter, therefore, is that it gives simple heuristics with posteriori guarantees of performance.

CHAPTER FIVE

Periodic Versus 'Can-Order' Policies

Introduction

In this chapter we consider the problem of the coordination of replenishment orders for groups of items in a multi-item inventory system. A single fixed cost is incurred whenever any item is replenished and an individual fixed cost is incurred for each item included in the joint replenishment. The latter is often referred to as a 'line item' cost on a joint reorder. Demands are generated by independent Poisson Processes. Excess demands are backlogged and there is a constant delivery lead time for each item. The full model details are described in section 2. Here we discuss the motivation and the main results of this chapter.

Although in practice periodic replenishment policies may perhaps be more prevalent, the literature has tended to concentrate an (s, c, S) or *can-order* systems. Such a policy operates as follows. When any item's inventory drops to its reorder point s_i a reorder is scheduled. Other items j are inspected and any at or below their *can-order* point c_j are also included in the reorder. Items i (resp., j) are reordered up to S_i (resp., S_j). Such policies were proposed by Balintfy [6] and have been investigated in a series of papers by Silver [65,68,69], Thompstone and Silver [75] and recently by Federgruen et al. [23], and have been shown to be considerably better than uncoordinated policies, often with savings around 20%.

We know 'can-order' policies are not optimal, e.g., Ignall [35]. However, their performance on small problems (two products) has been good, Silver and Peterson [54, page 450]. The question arises as to how good they are for realistically sized problems.

The cost allocation bound described in Chapter 2 also applies to this problem but, used as a lower bound, it is 'a long way' below the 'can-order' policy. By a 'long way' we mean varying from nothing when the joint cost is zero, that is no coordination opportunities, to up to 50% for subtantial joint costs. This raises the question whether the bound is weak or whether can-order policies can be considerably improved or both. These results were obtained on a 12-item example. In order to answer this question, an alternative heuristic is proposed which is very easy to calculate and implement. An obvious choice would be a heuristic with opportunity for coordination. One such heuristic is an (R, T) periodic review policy where every T_i periods the inventory of item i is raised to R_i. (R, T) policies are common in practice as they tend to lower set-up cost because the process of setting up for replenishment becomes a routine periodic operation. To achieve coordination, all periods T_i are integral multiples of a base period. This policy was tested on a range of problems and performed consistently better than 'can-order' policies except for small values of the joint cost. We would expect poor performance for small values of the joint cost because, for example, when the joint cost is zero, we are simply comparing periodic review with continuous review models. What was surprising was the much better performance for significant values of the joint cost, where the gap between the lower bound and the can-order policy was typically halved, a saving of often 20%. Encouraged by this result we also replaced the (R, T) policy by a simple heuristic. In this heuristic the T_i periods were calculated from essentially an Economic Order Quantity and then rounded to a multiple of the base period. This heuristic which is very easy to calculate performed only slightly worse than the (R, T) policy, and still much better than the 'can-order' policy.

Thus the main result of this chapter is that periodic review policies can outperform 'can-order' policies for coordinated replenishment inventory systems, and can do it with considerably simpler calculations. The remaining part of this chapter is organized as follows: section 2 describes the model assumptions, section 3 presents s, c, S and independent policies tested, section 4 gives a brief description of the lower bound, section 5 describes the periodic heuristics, section 6 gives details of the numerical results, while section 7 concludes with a discussion of topics for further research.

5.2 Model Assumptions

The N-item inventory system is subject to a continuous review and demands are generated by independent Poisson processes with rate λ_i for item i. The possibility of continuous review is required by 'can-order' policies even though only periodic review will be required for the periodic policies. Excess demands are backlogged and each replenishment needs a constant lead time l_i. The joint cost is A_0 and the 'line' cost is A_i. Holding costs are charged at a rate h_i on every item of inventory. Two types of shortage costs are used, ϕ_i for every item i unable to be filled immediately on request, and a rate p_i for every unit of backlogging outstanding. Thus p_i is a 'time-weighted' shortage charge. These shortage and holding costs are linear. The criterion is to minimize the long run average cost per unit time. Which of these assumptions can be readily generalized will be discussed in section 7.

5.3 'Can-Order' Policies

The method of calculation used for the parameters (s, c, S) is that given by Federgruen et al. [23]. We used only simple, not compound, Poisson demands and did not include service level constraints. The two types of shortage costs were used alternatively, not together. Further details on the (s, c, S) policies and the method of calculation are in Federgruen et al [23]. The calculations for the (s, c, S) policy are not quite exact because of the assumption of independence needed for item reorders. Again, details on the extent of this exactness are given in the above paper. In addition, we calculated the 'independent' policy, where item i incurs the full cost $A_0 + A_i$ whenever it is reordered. The optimal (s, S) policies for this were calculated using the methodology, used in Federgruen et al.[23, page 352].

5.4 The Lower Bound

The cost allocation method decomposes the problem into N single item problems, if there are N items, as in the joint replenishment problem in Chapter 2. The key here is that the 'optimum' policy for the single item problem is known. The set-up cost for item i after the cost allocation is $\alpha_i A_0 + A_i$. The choice of the best α's to use is difficult in this case. Recall that any feasible values of α will give a lower bound. However, we use a heuristic method to determine the allocations for the lower bound. For the joint replenishment problem with constant and continuous demands discussed in Chapter 2, the choice of α_i was made as follows: first calculate the run-out times for each item initially with $\alpha_i = 0$ for all i. We then allocate the set-up cost to raise the minimum cycle time to equal the next higher cycle time and repeat the allocation until there is no more set-up cost to allocate. We use the same procedure here with mean demand used for calculating the run-out times. The minimum run-out time achieved after the allocation is termed the *base period* and all items i with $\alpha_i > 0$ the *base set*, B. If

$C_i(\alpha_i)$ is the minimum cost for item i with allocation α_i, the lower bound is

$$\text{LB} = \sum_{i \in B} C_i(\alpha_i) + \sum_{i \notin B} C_i(0)$$

We note here that 'better' lower bounds are clearly available with more calculations.

5.5 The Periodic Policies

The (R, T) policy is one that orders the inventory position up to R every T periods. We use the Poisson demand distribution and the equations for calculating the average cost of an (R, T) model for each item below are taken from Hadley and Whitin [33, page 260], equations 5.65–67 inclusine.

Let $P[q; n_i\lambda_i T]$ be the probability that the demand in a review interval of length $n_i T$ is at least q. Let

$$\begin{aligned}
E_i(R_i, n_i T) &= \text{Average backorders per period for} \\
& \text{item } i \text{ under } (R_i, n_i T) \text{ policy} \\
B_i(R_i, n_i T) &= \text{Average unit years of shortage for} \\
& \text{item } i \text{ under } (R_i, n_i T) \text{ policy} \\
D_i(R_i, n_i T) &= \text{Average on-hand inventory per period for} \\
& \text{item } i \text{ under } (R_i, n_i T) \text{ policy}
\end{aligned}$$

The average cost is given by

$$\frac{A_i + \alpha_i A_0}{n_i T}(P[1; n_i\lambda_i T]) + h_i D_i(R_i, n_i T) + \phi_i E_i(R_i, n_i T) + p_i B(R_i, n_i T)$$

The expressions for D_i, E_i and B_i are given below.

$$\begin{aligned}
B(R_i, n_i T) &= \frac{1}{n_i T}[\frac{\lambda_i}{2}\{(l_i + n_i T)^2 P[R_i - 1; \lambda_i(l_i + n_i T)] - l_i^2 P[R_i - 1; \lambda_i l_i]\} \\
&+ \frac{R_i(R_i + 1)}{2\lambda_i}\{P[R_i + 1; \lambda_i(l_i + n_i T)] - P[R_i + 1; \lambda_i l_i]\} \\
&- R_i\{(l_i + n_i T)P[R_i; \lambda_i(l_i + n_i T)] - l_i P[R_i; \lambda_i l_i]\}]
\end{aligned}$$

$$\begin{aligned}
E(R_i, n_iT) &= \frac{1}{n_iT}\{\lambda_i(l_i + n_iT)P[R_i - 1; \lambda_i(l_i + n_iT)] - R_iP[R_i; \lambda_i(l_i + n_iT)] \\
&- \lambda_i l_i P[R_i - 1; \lambda_i l_i] + R_iP[R_i; \lambda_i l_i]\} \\
D(R_i, n_iT) &= R_i - \mu_i - \frac{\lambda_i n_i T}{2} + B(R_i, n_iT)
\end{aligned}$$

Three versions of this policy were tried. The first we shall call simply (R, T). For this, all items used the same period T,that is, $n_i = 1$ for all i. A single variable search was used to determine T. For the second, which we shall term (R, nT), a single search over T was also performed but any item could have a period of any n_iT where n_i is a positive integer. The values of n_i were determined in the following way. For each value of T and for each i, n_i was increased from 1 until the resulting cost no longer decreased.

Finally a simple alternative was tried, motivated by the lower bound in section 4. For items in the base set, $i \in B$, T was taken as the common base period. All other items $i \notin B$ (or rather their deterministic equivalents) have run-out times greater than T. This is because of the way the α_i were defined. Each of these items i is given a period n_iT nearest to the runout time, where again n_i is a positive integer. This heuristic will be termed PH, for periodic heuristic. For all three options R_i is chosen to minimize expected carrying and shortage costs during $l_i + T_i$. It should be recalled that for periodic policies, if an order is placed now, then another chance to place an order will not occur until T_i later and that will not result in products being available until a further time l_i. Hence current decisions must face the uncertainty of demand over a time $l_i + T_i$.

5.6 Computational Results

Twelve products are listed in Table 5.1, along with values for λ_i, A_i and l_i. The products have been ranked by the value of λ_i/A_i which means that the base set B occurs first and is actually products 1 to 6 inclusive.

DATA				Results										
Prod.	A	λ	l	LB			PH		(R,T)	(s,c,S)			Indep.	
				(s,S,T)			(R,T)		T=0.8				s,S	
1	10	40	0.2	11	39	0.65	41	0.65	46	8	34	46	10	58
2	10	35	0.5	22	47	0.65	47	0.65	52	17	43	54	20	66
3	20	40	0.2	11	39	0.65	41	0.65	46	8	32	49	10	59
4	20	40	0.1	6	34	0.65	36	0.65	42	4	27	44	5	54
5	40	40	0.2	11	39	0.65	41	0.65	46	8	29	53	10	62
6	20	20	1.5	35	52	0.65	50	0.65	53	23	46	58	32	70
7	40	20	1	24	43	0.82	40	0.65	42	14	33	50	21	60
8	40	20	1	24	43	0.82	40	0.65	42	14	33	50	21	60
9	60	28	1	33	60	0.85	54	0.65	58	25	44	69	30	78
10	60	208	1	23	46	1	50	1.3	42	13	32	53	21	62
11	80	20	1	23	49	1.15	50	1.3	42	12	31	55	21	64
12	80	20	1	23	49	1.15	50	1.3	42	12	31	55	21	64
COST				2047			2291		2322	2620			3357	
COST/LB				100			112		113	128			164	

Table 5.1: Data and Typical Results ($A_0 = 150$, $\phi = 30$, $p = 0$, $h = 6$)

NOTES :	LB	=	Lower Bound
	PH	=	Periodic heuristic, using T from the lower bound to the nearest multiple.
	(R, T)	=	Periodic policy with common cycle length T.
	(s, c, S)	=	'can-order' policy.

The (R, nT) results are identical to the PH results up to the last digit or two and are omitted for clarity.

For different experiments A_0 varies from 20 to 250. All the p_i, (ϕ_i) are identical and equal to 30 or zero alternately. All the h_i are identical and equal to either 2 or 6. Experiments were run with different and nonidentical values of h_i, p_i, (ϕ_i) but no essentially different results were obtained. Neither did varying the range of λ_i, A_i and h_i affect the qualitative impact of the results, although of course the actual numbers varied considerably. In terms of relative performance the most sensitive factor is A_0. Examples of some of these results for varying λ_i and l_i are shown in Table 5.5

Table 5.1 gives details of a run with $A_0 = 150$, $h = 6$ and using the non-time weighted shortage cost $\phi = 30$. Under LB are given the values of s, S and the 'runout' times as described in section 4. Notice that the base set have identical times, the base period of 0.65. Notice under PH that products 6 through 9 have been 'rounded down' to the base period and 10 through 12 'rounded up' to twice the base period. The minimizing value of T found with (R, T) was $T = 0.8$. Only a coarse grid search with interval 0.05 was used. Smaller grids were not worthwhile, achieving only very small improvements.

The savings of the (s, c, S) policy over the independent policy are consistent with previous work, see [23]. Note that the cost/LB ratio is to be interpreted in the light that the bound LB might still be quite weak. The simple PH policy is seen to perform well in comparison with others.

A_0	LB	PH	(R,T)	(s,c,S)	Indep.
50	999	1135	1183	1172	1349
		(114)	(118)	(117)	(135)
100	1051	1195	1223	1327	1643
		(114)	(116)	(126)	(156)
150	1096	1244	1262	1459	1886
		(114)	(115)	(133)	(172)
200	1137	1289	1299	1571	2052
		(113)	(114)	(138)	(180)
250	1175	1329	1334	1676	2239
		(113)	(114)	(143)	(191)

Table 5.2: $\phi = 30$, $p = 0$, $h = 2$

The first main experiment is given in Table 5.2. Everything is identical to the previous experiment except $h = 2$. The ratio of cost/LB is given underneath each figure in parentheses, in percentage points

Again the improvement of (s, c, S) over independent is consistent with previous findings. The performance of the periodic compared with the (s, c, S) policy improves as A_0 increases and is quite pronounced, reaching a 20% improvement for high values of A_0. Interestingly the ratio of PH/LB seems independent of A_0. We know however that for small values of A_0 the periodic policy will begin to behave poorly compared with (s, c, S) policies. We can see this in Table 5.3. Here the data is identical to Table 5.2 except that we use the time-weighted shortage cost and A_0 has been extended to the value of 20.

We see that with $A_0 = 20$, the (s, c, S) policies are better, but $A_0 = 20$ is a rather small cost for this data set and the whole value of coordination is becoming marginal.

A_0	LB	PH	(R,T)	(s,c,S)	Indep.
20	815	907	961	922	975
		(111)	(118)	(110)	(120)
50	851	944	984	1114	1189
		(111)	(116)	(131)	(140)
100	901	1005	1023	1243	1476
		(112)	(114)	(138)	(164)
150	945	1046	1059	1350	1713
		(111)	(112)	(143)	(183)
200	981	1088	1095	1443	1908
		(111)	(112)	(147)	(194)

Table 5.3: $\phi = 0$, $p = 30$, $h = 2$

An extra run with $A_0 = 0$ confirmed what we already know. For this LB, (s, c, S) and Indep. were all identical and the periodic solutions more costly.

A final run is shown in Table 5.4, identical to Table 5.3 except that $h = 6$, and with a smaller range of A_0.

To demonstrate the robust nature of these results some rather different experiments were performed. The base case for these experiments, example 1 in Table 5.5, had the following parameters.

Eight identical products,

$$A_i = 20 \quad A_0 = 150$$

$$\lambda_i = 40 \quad l = 0.2 \quad h = 6 \quad \phi = 30 \quad p = 0$$

Only the values for the lower bound, the independent policy, the 'can-order' and periodic heuristic are shown.

In experiments 1–3 the lead times are lengthened and the biggest change is in the

A_0	LB	PH	(R,T)	(s,c,S)	Indep.
100	1493	1676	1673	1898	2362
		(112)	(112)	(127)	(158)
150	1562	1733	1743	2054	2738
		(111)	(112)	(131)	(175)
200	1626	1803	1798	2193	3069
		(111)	(111)	(135)	(189)

Table 5.4: $\phi = 0$, $p = 30$, $h = 6$

lower bound thereby improving the relative measure of performance of both heuristics. It needs to be emphasized that the actual performance need not be improving; only our estimate using the lower bound did. A caution needs also to be made with respect to the periodic heuristic.

The heuristic solution was calculated in the traditional way, assuming a very low chance of more than one outstanding order per item. As lead times lengthen this assumption becomes suspect. See [33] for more on this. Experiments 4 and 5 changed the demand and the number of products without changing the aggregate demand. Again the relative performance was little affected.

5.7 Concluding Remarks

For the model studied here simple periodic policies seem to show considerable promise over more complex 'can-order' system. Periodic policies were studied by Naddor [52] but without a bound or computational comparison with 'can-order' systems. Natural extensions to this work are to compound Poisson demands, systems under service level constraints and lost sales case. Alternatively the discrete nature of demand can be

Example	Description	LB	PH	(s, c, S)	Indep.
1	Base case	1357	1559	1929	2490
2	Base case except lead times have increased to 0.4	1451	1615	1991	2563
3	Base case except lead times have increased to 0.6	1523	1664	2043	2618
4	Base case except demand has changed to 4 products with $\lambda_i = 20$, and 4 products with $\lambda_i = 60$	1348	1543	1869	2405
5	Base case except only 4 products with $\lambda_i = 80$	1348	1543	1869	2405

Table 5.5: Summary of Results with Different Lead Times and Demands

dropped in favour e.g., of normal distribution of demands.

More profound extensions would raise the following questions. Can this bound be sharpened? We noted in Section 4 that we are not using the values of α_i that would maximize the lower bound as was the case with deterministic demands in previous chapters. A second question is, can the heuristic be improved? A third and most interesting question is, can a guaranteed performance be given for a heuristic compared to the bound?

One last comment which could have implications for extensions is the observation that although (s, c, S) policies require independence for their computation, nowhere is independence needed for either the lower bound or the periodic heuristic.

CHAPTER SIX

Conclusion

Lower bounds for some production/inventory problems were derived using a cost allocation method. The generality of this method is demonstrated in its use for the joint replenishment problem with constant and time-varying demands, the one-warehouse multi-retailer problem, the multi-facility joint replenishment problem, the multi-item dynamic lot-size problem and the (s, c, S) or 'can-order' model. This work can be extended in various directions. For example, we could relax the assumption on backlogging or use more complex cost structures for the joint replenishment problem with constant and continuous demands and the multi-item dynamic lot-size problem. Backlogging of demand could also be allowed in the one-warehouse multi-retailer problem. For the 'can-order' system some possible extensions include: using alternative demand distributions, like compound Poisson distribution; allowing lost sales and no backlogging; incorporating service level constraints and using alternative heuristics. For the lower bound itself, research into better ways of choosing the cost allocations in the multi-product dynamic lot-size and the 'can-order' systems is needed. Better choices of the cost allocations will help in extablishing better performances for the heuristics introduced here and possibly give ideas about other heuristics.

Bibliography

[1] Afentakis, P., B. Gavish and U. Karmarkar. "Computationally Effecient Optimal Solutions to the Lot-Sizing Problem in Multi-Stage Assembly Systems." *Management Sci.*, 30 (1984), 222–239.

[2] Agnihotri, S., U. Karmarkar and P. Kubat. "Stochastic Allocation Rules." *Oper. Res.*, 30 (1982), 545–555.

[3] Allen, G. S. "Redistribution of Total Stock Over Several User Locations." *Nav. Res. Log. Quart.*, 5 (1958), 337–345.

[4] Andres, F. M. and H. Emmons. "A Multi-product Inventory System with Interactive Setup Costs." *Management Sci.*, 21 (1975), 1055–1063.

[5] Andres, F. M. and H. Emmons. "On the Optimal Packaging Frequency of Products Jointly Replenished." *Management Sci.*, 22 (1976), 1165–1166.

[6] Balintfy, J. L. "On a Basic Class of Multi-Item Inventory Problem." *Management Sci.*, 10 (1964), 287–297.

[7] Bensoussan, A. and J. M. Proth. "Economical Ordering Quantities for the Two Products Problem with Joint Production Costs." APII, Systemes de production. 19 (1985) 509–521.

[8] Bitran G. R. and T. L. Magnanti and H. H. Yanasse. "Approximation Methods for the Uncapacitated Dynamic Lot Size Problem." *Management Sci.*, 30(1984), 1121–1141.

[9] Blackburn, J, D. and R. A. Millen. "Improved Heuristics for Multi-Stage Requirement Planning Systems." *Management Sci.*, 28 (1982), 44–56.

[10] Bomberger, E. "A Dynamic Programming Approach to a Lot Size Scheduling Problem." *Management Sci.*, 12 (1966), 778–784.

[11] Brown, R. G. *Decision Rules for Inventory Management.* Holt, Rinehart, and Winston, NY, 1967.

[12] Cahen, J. F. "Stock Policy in Case of Simultaneous Ordering." *Int. J. Prod. Res.*, 10 (1972), 301–312.

[13] Chakravarty, A. K. "Multi-Item Inventory Aggregation into Groups." *J. Opl Res.*, 22 (1981), 19-26.

[14] Chakravarty, A. K., J. B. Orlin and U. G. Rothblum. "A Partitioning Problem with Additive Objective with an Application to Optimal Inventory Grouping for Joint Replenishment." *Oper. Res.*, 30 (1982), 1018–1022.

[15] Clark, A. "An informal Survey of Multi-Echelon Inventory Theory." *Nav. Res. Log. Quart.*, 19 (1972), 621–650.

[16] Clark, A. J. and H. Scarf. "Optimal Policies for a Multi-Echelon Inventory Problem", *Management Sci.*, 6 (1960), 475–490.

[17] Crowston, W. B., M. H. Wagner and A. Henshaw. "A Comparison of Exact and Heuristic Routines for Lot-Size Determination in Multi-Stage Assembly Systems." *AIIE Transactions*, 4 (1972), 313–317.

[18] Crowston, W. B., M. H. Wagner and J. F. Williams. "Economic Lot Size Determination in Multi-Stage Assembly Systems." *Management Sci.*, 19 (1973), 517–527.

[19] Das C. "Supply and Redistribution Rules for Two-Location Inventory Systems: One-Period Analysis." *Management Sci.*, 21 (1975), 765–776.

[20] Doll, C. L. and D. C. Whybark "An Iterative Procedure for the Single-Machine Multi-Product Lot Scheduling Problem." *Management Sci.*, 20 (1973), 50–55.

[21] Eppen, G. D. "Effects of Centralization on Expected Costs in a Multi-Location Newsboy Problem." *Management Sci.*, 25 (1979), 498–501.

[22] Eppen, G. and L. Schrage. "Centalized Ordering Policies in a Multiwarehouse System with Leadtimes and Random Demand." in *Multi-Level Production/Inventory Control Systems Theory and Practice.* Schwarz, L., Ed., North Holland, Amsterdam. (1981), 51–69.

[23] Federgruen, A., H. Groenevelt and H. C. Tijms. "Coordinated replenishment in a multi-item inventory system with compound Poisson Demands." *Management Sci.*, 30(1984), 344–357.

[24] Federgruen, A. and P. Zipkin. "Approximations of Dynamic, Multilocation Production and Inventory Problems." *Management Sci.*, 30 (1984), 69–84.

[25] Federgruen, A. and P. Zipkin. "Allocation Policies and Cost Approximations for Multilocation Inventory Systems." *Nav. Res. Log. Quart.*, 31(1984), 97–129.

[26] Goyal, S. K. "Determination of Economic Packaging Frequency for Items Jointly Replenished." *Management Sci.*, 20 (1973), 232–235.

[27] Goyal, S. K. "Determination of Optimum Packaging Frequency of Items Jointly Replenished." *Management Sci.*, 21 (1974), 436–443.

[28] Goyal, S. K. "Optimum Ordering Policy for a Multi Item, Single Supplier System." *Oper. Res. Quart.*, 25 (1974), 293–298.

[29] Goyal, S. K. and A. S. Belton. "On 'A Simple Method of Determining Order Quantities in Joint Replenishments Under Deterministic Demand'." *Management Sci.*, 26 (1979), 604.

[30] Graves, S. C. "On the Deterministic-Demand Multi-Product Single-Machine Lot Scheduling Problem." *Oper. Res.*, 25 (1979), 276–280.

[31] Graves, S. C. "Multi-stage Lot-Sizing: An Iterative Procedure." in *Multi-Level Production/Inventory Control Systems: Theory and Practice.* Schwarz, L. B. (Ed.), North Holland.

[32] Graves, S. C. and L. B. Schwarz. "Single Cycle Continuous Review Policies for Arborescent/Inventory Systems." *Management Sci.*, 23 (1977), 529–540.

[33] Hadley, G. and T. M. Whitin. *Analysis of Inventory Systems.* Prentice Hall, 1963.

[34] Hu, T. C. *Combinatorial Algorithms.* Addison-Wesley, 1982.

[35] Ignall, E. "Optimal Continuous Review Policies for the Two Product Inventory Systems with Joint Setup Costs." *Management Sci.*, 15(1969), 277-279.

[36] Jackson, P., W. Maxwell and J. Muckstadt. "The Joint Replenishment Problem with a Powers-of-Two Restriction." *IIE Transactions*, 17 (1985), 25–32.

[37] Jensen, P. A. and H. A. Khan. "Scheduling in a Multistage Production System with Set-up and Inventory Costs." *AIIE Transactions*, 4 (1972), 126–133.

[38] Johnson, E. L. "Optimality and Computation of (σ, S) Policies in the Multi-item Infinite Horizon Inventory Problem." *Management Sci.*, 13 (1967), 475–491.

[39] Kalin D. "On the Optimality of (σ, S) Policies." *Management Sci.*, 5 (1980), 293–307.

[40] Kalymon, A. B. "A Decomposition Algorithm for Arborescence Inventory Systems." *Oper. Res.*, 20 (1970), 860–873.

[41] Kao, E. P. C. "A Multi-Product Dynamic Lot-Size Model with Individual and Joint Set-Up Costs." *Oper. Res.*, 27 (1979), 279–289.

[42] Karmarkar, U. "Equalization of Runout Times." *Oper. Res.*, 29 (1981), 757–752.

[43] Lambrecht, M. and J. Eecken Vander. "A Facilities in Series Capacity Constrained Dynamic Lot-Sized Model." *European J. Oper. Res.*, 2 (1978), 42–49.

[44] Leopoulos, V. I. and J. M. Proth. "The General Multi-Products Dynamic Lot Size Model with Individual Inventory Costs and Joint Production Costs." *RAIRO APII*, 19 (1985), 117–130.

[45] Love, S. "A Facilities in Series Inventory Model with Nested Schedules." *Management Sci.*, 18 (1972), 327–338.

[46] Madigan, J. G. "Scheduling a Multi-Product Single Machine System for an Infinite Planning Period." *Management Sci.*, 14 (1968), 713–719.

[47] Mendelson,H., J. Pliskin and U. Yechiali. "A Stochastic Allocation Problem." *Oper. Res.*, 28 (1980), 687–693.

[48] Miltenburg, G. J. and E. A. Silver. "Accounting for Residual Stock in Continuous Coordinated Control of a Family of Items." *Int. J. Prod. Res.*, 22 (1984), 607–628.

[49] Miltenburg, G. J. and E. A. Silver. "The Diffusion Process and Residual Stock in Periodic Review Coordinated Control of Families of Items." *Int. J. Prod. Res.*, 22 (1984), 629–646.

[50] Moily, J. A. "Optimal and Heuristic Procedures for Component Lot-Splitting in Multi-Stage Manufacturing Systems." *Management Sci.*, 32 (1986), 113–125.

[51] Muckstadt, J. A. and R. O. Roundy. "Multi-Item, One-Warehouse, Multi-Retailer Distribution Systems. Technical Report No. 646, 1985. Dept. of Industr. Eng., Cornell University, Ithaca.

[52] Naddor, E. "Optimal and Heuristic Decisions in Single and Multi-Item Inventory Systems." *Management Sci.*, 21(1975), 1234-1249.

[53] Nocturne, D. J. "Economic Ordering Frequency for Several Items Jointly Replenished." *Management Sci.*, 19 (1973), 1093–1096.

[54] Peterson R. and E. A. Silver. *Decision Systems for Inventory Management and Production Planning.* Wiley, New York, 1985.

[55] Queyranne, M. "A Polynomial-Time, Submodular Extension to Roundy's 98%-Effective Heuristic for Production/Inventory Systems." Tecnichnical Report No. 1136, 1985. Faculty of Commerce and Bus. Admin., University of British Columbia, Vancouver.

[56] Roundy, R. "94%-Effective Lot-Sizing in Multi-Stage Assembly Systems." Technical Report No. 647. (1983), Dept of Indust. Eng., Cornell University, Ithaca.

[57] Roundy, R. O. "98%-Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems." *Management Sci.*, 31 (1985), 1416–1430.

[58] Roundy, R. "Efficient, Effective Lot-Sizing For Multi-Product Multi-Stage Production/Distribution Systems with Correlated Demands." Tecnhnical Report No. 671. (1985), Dept of Industr. Eng., Cornell University, Ithaca.

[59] Roundy R. "A 98%-Effective Lot-Sizing Rule for a Multi-Product, Multi-Stage Production/Inventory System." *Maths. of OR.*, 11 (1986), 699–727.

[60] Schwarz, L. B. "A Simple Continuous Review Deterministic One-Warehouse N-Retailer Inventory Problem." *Management Sci.*, 19 (1973), 555–566.

[61] Schwarz, L. B. "Physical Distribution: The Analysis of Inventory and Location." *AIIE Transactions*, 13 (1981), 138–150.

[62] Schwarz, L. B. and L. Schrage. "Optimal and System-Myopic Policies for Multi-Echelon Production/Inventory Assembly Systems." *Management Sci.*, 21 (1975), 1285–1294.

[63] Schweitzer, P. J. and E. A. Silver. "Mathematical Pitfalls in the One Machine Multiproduct Economic Lot Scheduling Problem." *Oper. Res.*, 31 (1983), 401–405.

[64] Shu, F. T. "Economic Ordering Frequency for Two Items Jointly Replenished." *Management Sci.*, 17 (1971), B-406–410.

[65] Silver, E. A. "A Control System for Coordinated Inventory Replenishment." *Int. J. Prod. Res.*, 12 (1974), 647–671.

[66] Silver, E. A. "A Simple Method of Determining Order Quantities in Joint Replenishment Under Deterministic Demand." *Management Sci.*, 22 (1976), 1351–1361.

[67] Silver, E. A. "Some thoughts on Extension of Lot-Sizing Procedures Under Deterministic Time-Varying Demand." Working Paper No. 104, 1976. Dept of Management Science, U. of Waterloo.

[68] Silver, E. A. "Coordinated Replenishments of Items Under Time Varying Demand: Dynamic Programming Formulation." *Nav. Res. Log. Quart.*, 26 (1979), 141–151.

[69] Silver, E. A. "Establishing Reorder Points in the (S, c, s) Coordinated Control System under Compound Poisson Demand." *Internat. J. of Prod Res.*, 9 (1981), 743–750.

[70] Simpson, F. K. Jr. "A Theory of Allocation of Stocks to Warehouses." *Oper. Res.*, 7 (1959), 797–805.

[71] Steinberg, E. and H. A. Napier. "Optimal Multi-Level Lot-Sizing for Requirements Planning Systems." *Management Sci.*, 26 (1980), 1258–1271.

[72] Szendrovits, A. Z. "Manufacturing Cycle Time Determination for a Multi-Stage Economic Production Quantity Model." *Management Sci.*, 22 (1975), 298–308.

[73] Taha, H. A. and R. W. Skieth. "The Economic Lot Sizes in Multi-Stage Production Systems." *AIIE Transactions*, 2 (1970), 157–162.

[74] Tan, K. F. "Optimal Policies for a Multi-Echelon Inventory Problem with Periodic Ordering." 20 (1974), 1104–1111.

[75] Thompstone, R. and E. A. Silver. "A Coordinated Inventory Control System for Compound Poisson Demand and Zero Lead Time." *Int. J. Prod. Res.*, 13 (1975), 581–602.

[76] Veinott, A. F. "Optimal Policy for a Multi-product, Dynamic, Non-stationar Inventory Problem." *Management Sci.*, 12 (1965), 206–222.

[77] Veinott, A. F. "On the Optimality of (s, S) Inventory Policies. New Conditions and a New Proof." *SIAM J. Appl Math.*, 14 (1966), 1067–1083.

[78] Veinott, A. F. "Minimum Concave-Cost Solution of Leontief Substitution Models of Multifacility Inventory Systems." *Oper. Res.*, 17 (1969), 262–291.

[79] Veinott, A. F., Jr. and H. M. Wagner. "Computing Optimal (s, S) Inventory Policies." *Management Sci.*, 11 (1965), 525–552.

[80] Wagner, H. M. and T. Whitin. "Dynamic Version of the Economic Lot Size Model", *Management Sci.*, 5 (1958), 89–96.

[81] Washburn, A. "A Note on Integer Maximization of Unimodular Functions." *Oper. Res.*, 23 (1975), 358–360.

[82] Wheeler, A. "Multiproduct Inventory Models with Set-up." *Technical Report* No. 106, 1968. Oper. Res. Dept. Stanford University, Stanford.

[83] Williams, J. F. "Heuristic Techniques for Simultaneous Scheduling of Production and Distribution in Multi-Echelon Structures: Theory and Empirical Comparisons." *Management Sci.*, 27 (1981), 336–352.

[84] Williams, J. F. "A Hybrid Algorithm for Simultaneous Scheduling of Production and Distribution in Multi-Echelon Structures. *Management Sci.*, 29 (1983), 77–105.

[85] Zabel, E. "Some Generalization of an Inventory Planning Horizon Theorem." *Management Sci.*, 10 (1964), 465–471.

[86] Zangwill, W. I. "A Deterministic Multiproduct Multifacility Production and Inventory Model." *Oper. Res.*, 14 (1966) 486–507.

[87] Zangwill, W. I. "A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot Size Production System—A Network Approach." *Management Sci.*, 19 (1969), 506–527.

[88] Zipkin, P. "Simple Ranking Methods for Allocation of One Resourse." *Management Sci.*, 26 (1980), 34–43.

[89] Zipkin, P. "On the Imbalance of Inventories in Multi-Echelon Systems." *Maths. of Oper. Res.*, 9 (1984), 402–423.

APPENDIX A

The Minimum Violators Algorithm for a Facilities-in-Series Problem

Initialize

Let the root facility be j and let $A_k \leftarrow 0$ temporarily, $k \in (P_n \cup S_j) \setminus \{n\}$ Let $a_m \leftarrow \alpha_{im} A_m$, $e_m \leftarrow h_{im}$, $m \leq n$, $e_n \leftarrow h'_{in}$; where

$$h'_{in} = \sum_{k \in P_n \cap S_j} h_{ik}$$

$FAC \leftarrow (P_i \setminus (P_n \setminus \{n\})$, $CL(i,m,m) \leftarrow \{m\}$, $m < n$; $CL(i,n) \leftarrow P_n$, $s_m \leftarrow \max\{k < m | m \in FAC|\}$, $c_m \leftarrow \frac{a_m}{e_m}$, $s_i \leftarrow \emptyset$, $FAC1 \leftarrow FAC$.

Step 1:

While $FAC1 \neq \emptyset$ repeat $l \leftarrow \arg\min\{c_k | k \in FAC1\}$ If $s_l = \emptyset$ or $c_l > c_{s_l}$, $FAC1 \leftarrow FAC1 \setminus \{l\}$.

If $FAC1 = \emptyset$, STOP, the current clusters are optimum; otherwise go to step 2.

Step 2:

$CL(i,l,l) \leftarrow CL(i,l,l) \cup CL(i,s_l)$; $a_l \leftarrow a_l + a_{s_l}$, $e_l \leftarrow e_l + e_{s_l}$, $c_l \leftarrow \frac{a_l}{e_l}$, $FAC \leftarrow FAC \setminus \{s_l\}$, $s_l \leftarrow s_{s_l}$,

Set $T_{ik} = c_l$ for all $k \in CL(i,l,l)$.

$FAC1 \leftarrow FAC$. Repeat step 1.