OPTIMAL SOLUTIONS FOR THE LOT-SIZING PROBLEM IN MULTI-STAGE ASSEMBLY SYSTEMS

Alistair R. Clark

Vinicius A. Armentano

Departamento de Engenharia de Sistemas
FEE - UNICAMP
CP 6101 - 13081 - Campinas - SP

Resumo - O Problema de Dimensionamento de Lotes em Sistemas de Montagem Multi-Estágio com custos que variam no tempo é considerado. O problema é formulado como um Programa Linear Inteiro Misto que permite uma decomposição através de métodos de Relaxação Lagrangeana. O problema Lagrangeano, combinado com otimização por subgradiente, fornece limites inferiores que são utilizados em um algoritmo de Branch-and-Bound. Testes computacionais mostram que, embora a convergência do algoritmo possa ser lenta para o caso de custos gerais, uma solução ótima pode ser rapidamente determinada no caso de custos de preparação e custos unitários de produção que não aumentam ao longo do tempo.

Abstract - The Lot-Sizing Problem in Multi-Stage Manufacturing Assembly Systems with time-varying costs is considered. The problem is formulated as a Mixed Integer Linear Program that permits decomposition through Lagrangian Relaxation. The Lagrangian problem combined with subgradient optimization provides lower bounds that are used in a Branch-and-Bound algorithm. Computational tests show that, while an optimal solution is difficult to achieve within reasonable time in the general costs case, an optimal solution is quickly found in the case of setup and unit production costs that do not increase over time.

1. INTRODUCTION

Material Requirements Planning (MRP) is widely used in the batch manufacture of products in multi-stage assembly systems. MRP takes into account the component structure of each product to determine a plan for the production of all components over a given planning horizon. One of the limitations of MRP is that when production and inventory costs are taken into consideration it is through the use of single-stage lot-sizing methods such as the optimal Wagner-Within algorithm (Wagner & Within, 1958; Wagner, 1960), or heuristics such as (Silver & Meal, 1973), (De Matties, 1968) or (Groff, 1979). Such methods are applied to one component stage at a time and do not consider the impact of costs at predecessor stages. Since a component's demand depends on the lot-sizes of its successor component stages, the lot-sizing decisions at a given stage should ideally simultaneously consider all component stages. This is a Multi-Stage Lot-Sizing (MSLS) problem.

The MSLS problem has received considerable research attention in the last twenty years. A recent review can be found in (Bahl et alli, 1987). Several heuristic approaches (McLaren & Whybark, 1976; Graves, 1981; Blackburn & Millen, 1982; Afentakis, 1987) have been developed with some good results, particulary in (Afentakis, 1987), being obtained in reasonable computing time. In contrast optimal solutions have proved to be much more elusive. Optimal approaches include Dynamic Programming and/or Networks (Zangwill, 1969; Crowston et alli, 1973; Crowston & Wagner, 1973; Steinberg & Napier, 1980) and Branch-and-Bound (Crowston & Wagner, 1973; Schwarz & Schrage, 1975; Afentakis et alli, 1984), but these are often either restricted to special cases such as constant demand over an infinite horizon (Crowston et alli, 1973; Schwarz & Schrage, 1975) and near-serial systems (Zangwill, 1969; Crowston & Wagner, 1973), or are computationally feasible only for small problems (Steinberg & Napier, 1980). However Afentakis, Gavish and Karmarkar (1984) were able to find optimal solutions in reasonable computing time for non-serial assembly systems of medium size with time-constant costs, using Branch-and-Bound with lower bounds obtained via Lagrangean Relaxation. This paper attempts a similar approach for the more general case of time-varying costs.

2. THE MODEL

Our concern is to plan the production in discrete lots of a simple item and its components over a planning horizon of T periods. We must decide in which periods we will produce which components and what the corresponding production lot sizes will be. A decision to produce a lot in a period will incur a fixed setup cost and a unit production cost. Components may be held in inventory from one period to the next in which case a unit stock holding cost will be incurred. A component's costs are allowed to differ between periods. We assume that a component produced in a period is available for consumption in the same period, that there are no capacity constraints on the amount produced, and that there is no initial stock of the item and its components. Our objective is to find a plan that minimizes the total of setup, production and inventory costs over the planning horizon.

Before specifying the model, let us label the item's components from 1 to N such that if component i uses component j then $i < j$. The end-item is labelled 1. The component structure of the item can be represented by an acyclic graph. In this paper we consider only assembly structure and so the acyclic graph is a tree with the end-item as its "root".

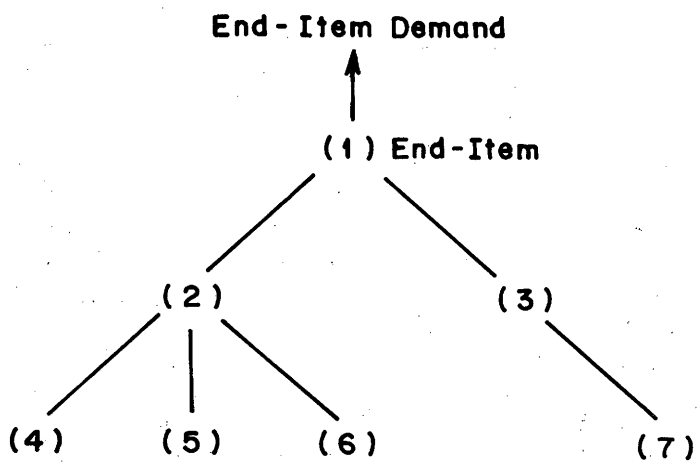

Figure 1

Denote $s(i)$ as the immediate successor component to component i, $R(i)$ as the set of all successors to i, $P(i)$ as the set of immediate predecessors of i, and $Q(i)$ as the set of all predecessors of i. Thius, in Figure 1, $s(5) = 2$, $R(5) = \{1,2\}$, $P(2) = \{4,5,6\}$, and $Q(1) = \{2,3,4,5,6,7\}$. Let $r_{i,s(i)}$ be the number of units (not necessarily integer) of component i needed by one unit of component $s(i)$. Then our model can be formulated as

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} [s_{it} y_{it} + c_{it} x_{it} + h_{it} I_{it}]$$

such that

$$I_{1,t-1} + x_{1t} - I_{1t} = d_{1t} \qquad t = 1 \cdots T$$

$$I_{i,t-1} + x_{it} - I_{it} - r_{i,s(i)} x_{s(i),t} = d_{it} \qquad \begin{matrix} i = 2 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \leq M_{it} y_{it} \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \geq 0; \quad I_{it} \geq 0; \quad y_{it} = 0 \text{ or } 1; \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

where

x_{it} = lot-size of component i in period t
I_{it} = inventory of component i at the end of period t
s_{it} = setup cost of component i in period t
c_{it} = unit production cost of component i in period t
h_{it} = unit inventory cost applied to I
d_t = end-item demand in period t
M_{it} = an upper bound on x_{it}

This is a Mixed Integer Linear Program (MILP) with NT 0/1 variables and we will attempt a Branch-and-Bound (B&B) approach to solving it to optimality. Sections 3 and 4 describe respectively the lower and upper bounding procedures used at each B&B node; section 5 explains the B&B separation strategy; section 6 considers the special case of costs that do not increase over time; while finally section 7 reports on our computational experience.

3. THE NODE LOWER BOUNDING PROCEDURE

Define E_{it}, the *echelon stock* of component i at the end of period t, as

$$E_{it} = I_{it} + r_{i,s(i)} E_{s(i),t} = I_{it} + \sum_{j \in R(i)} p_{ij} I_{jt}$$

where $p_{ij} = \prod_k r_{k,s(k)}$ and the product $\prod_k$ is over all components k on the path from component i to component j, excluding component j itself. i.e., p_{ij} is the quantity of component i in one unit of component j.

The echelon stock E_{it} is the total system stock of component i at the end of period t, both as a stand-alone component (I_{it}) and as part of successor components

$$\sum_{j \in R(i)} p_{ij} I_{jt}$$

Echelon stock is a useful analytically simplifying concept because, whereas the amount of conventional stock I_{it} depends not only upon the lot-size of component i but also upon the lot-sizes of the components j in $R(i)$, the amount of echelon stock depends only upon the lot-size of component i (and, of course, the ever-present 'no-stockout' constraints). The echelon stock concept was introduced

by Clark and Scarf (1960) and has been used by a number of authors (Crowston et alli, 1973; Crowston & Wagner, 1973; Schwarz & Schrage, 1975; Afentakis et alli, 1984).

To E_{it} we can attach a unit echelon stock holding cost defined as

$$e_{it} = h_{it} - \sum_{j \in P(i)} r_{ji} h_{jt}$$

It can be shown that

$$\sum_{t=1}^{T}\sum_{i=1}^{N} h_{it} I_{it} = \sum_{t=1}^{T}\sum_{i=1}^{N} e_{it} E_{it}$$

i.e. that e_{it} is the unit holding cost of E_{it}. The cost e_{it} can be viewed as the unit holding cost per period of the value added at the processing of component i which had not been added in the processing of its predecessor components, i.e. e_{it} is the unit cost per period for the delay of the receipt of this added value.

The MILP can be reformulated in terms of echelon stock as follows:

$$\min \sum_{i=1}^{N}\sum_{t=1}^{T} [s_{it} y_{it} + c_{it} x_{it} + e_{it} E_{it}]$$

such that

$$E_{i,t-1} + x_{it} - E_{it} = p_{it} d_t \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$r_{i,s(i)} E_{s(i),t} - E_{it} \leq 0 \quad (*) \qquad \begin{matrix} i = 2 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \leq M_{it} y_{it} \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \geq 0; \quad E_{it} \geq 0; \quad y_{it} = 0 \text{ or } 1; \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

A proof that the two formulations are equivalent can be found in (Afentakis et alli, 1984). Note that the echelon stock formulation consists of N separate single-level lot-sizing problems (one per component) which would be independent were it not for the constraints:

$$r_{i,s(i)} E_{s(i),t} - E_{it} \leq 0 \qquad \begin{matrix} i = 2 \cdots N \\ t = 1 \cdots T \end{matrix} \qquad (*)$$

which are merely the constraints

$$I_{it} \geq 0 \qquad \begin{matrix} i = 2 \cdots N \\ t = 1 \cdots T \end{matrix}$$

expressed in terms of echelon stock. The echelon stock reformulation suggests the relaxation of the constraints $(*)$ into the objective function to provide a Lagrangian problem whose value is a lower bound on the MILP (Fisher, 1981). The Lagrangian problem is

$$Z_D(\lambda) = \min \sum_{i=1}^{N}\sum_{t=1}^{T} [s_{it} y_{it} + c_{it} x_{it} + e_{it} E_{it}] +$$

$$+ \sum_{i=2}^{N}\sum_{t=1}^{T} \lambda_{it} \left[r_{i,s(i)} E_{s(i),t} - E_{it} \right]$$

such that

$$E_{i,t-1} + x_{it} - E_{it} = p_{i1} d_t \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \leq M y_{it} \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

$$x_{it} \geq 0; \quad E_{it} \geq 0; \quad y_{it} = 0 \text{ or } 1; \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

where $\{\lambda_{it} \geq 0 \mid i = 2 \cdots N; t = 2 \cdots T\}$ is the vector of Lagrangian Multipliers. $Z_D(\lambda)$ decomposes into N separate single-level lot-sizing problems, one for each component i, each of which can be restated as

$$Z_{iD}(\lambda_i) = \min \sum_{t=1}^{T} [s_{it} y_{it} + c_{it} x_{it} +$$

$$\left(e_{it} + \sum_{j \in P(i)} r_{ji} \lambda_{jt} - \lambda_{it} \right) E_{it}]$$

such that

$$E_{i,t-1} + x_{it} - E_{it} = p_{i1} d_t \qquad t = 1 \cdots T$$

$$x_{it} \leq M_{it} y_{it} \qquad t = 1 \cdots T$$

$$x_{it} \geq 0; \quad E_{it} \geq 0; \quad y_{it} = 0 \text{ or } 1; \qquad t = 1 \cdots T$$

where $\lambda_{1t} = 0$ for $t = 1 \cdots T$ in $Z_{1D}(\lambda_1)$.

The coefficient of E_{it} may sometimes be negative (because of the $-\lambda_{it}$ term), but the holding cost function is linear. Production costs are concave and so, as Denardo (1982) shows, for all $i = 1 \cdots N$ there exists at least one optimal solution to $Z_{iD}(\lambda_i)$ for which $E_{i,t-1} x_{it} = 0$ for all $t = 1 \cdots T$. Thus we may use the time-varying costs version of the Wagner-Whitin (W-W) algorithm (Wagner & Whitin, 1958; Wagner, 1960) to solve $Z_{iD}(\lambda_i)$.

At a given node of the B&B tree some of the y_{it} are fixed at either 0 or 1 and the rest are free. In periods t for which y_{it} is fixed at 0, production is forbidden and this can be forced by setting s_{it} to infinity inside the W-W algorithm. In periods t for which y_{it} is fixed at 1, the setup cost s_{it} is always incurred in which case s_{it} can be set to zero inside the W-W algorithm and the original value of s_{it} added to the W-W solution. An example of such cost modifications is given at the end of section 4.

A lower bound at the node is given by the dual

$$Z_D(\lambda) = \sum_{i=1}^{N} Z_{iD}(\lambda_i)$$

We attempt to increase the value of $Z_D(\lambda)$ at the node by iterating a fixed number of times on λ, using subgradient optimization with the modifications proposed by Camerini, Fratta and Maffioli (1975).

4. THE NODE UPPER BOUNDING AND INCUMBENT UPDATING PROCEDURE

The upper bound at a node is used in the determination of the step size in the lower bound subgradient optimization. In addition, if the node upper bound is less than the incumbent value, then the upper bound becomes the new incumbent. The upper bound is chosen to be the better of the following two methods:

1. Use the Wagner & Within (W-W) algorithm at the end-item level and a lot-for-lot policy at component level. Thus at the B&B root node the modified end-item costs used in period t in the W-W algorithm are

$$\hat{s}_{1t} = \sum_{i=1}^{N} s_{it}; \quad \hat{c}_{1t} = \sum_{i=1}^{N} p_{i1} c_{it}; \quad \hat{h}_{1t} = h_{1t}$$

At B&B nodes $2, 3, 4, \ldots$ the modified costs used in period t in the application of the W-W algorithm to the end-item depend on the value of y_{1t} as follows:

- if y_{1t} is fixed at 0 then the modified end-item costs in period t are

$$\hat{s}_{1t} = \text{infinity}; \quad \hat{c}_{1t} = \text{irrelevant}; \quad \hat{h}_{1t} = h_{1t}$$

- if y_{1t} is fixed at 1 or is free then the modified end-item costs in period t are calculated taking into account that, for any t, if y_{1t} is fixed at 0 then the production of component i in period t is prohibited and any desired "lot-for-lot" production of component i and its predecessors must be anticipated. For example, with $N = T = 3$, $s(2) = 1$, $s(3) = 2$, then when y_{22} and y_{33} are fixed at 0 we use the following modified end-item costs:

$$\hat{s}_{11} = s_{11} + s_{21} + s_{31} \qquad \hat{s}_{12} = s_{12} + s_{21} + s_{31}$$

$$\hat{s}_{13} = s_{13} + s_{23} + s_{32}$$

$$\hat{c}_{11} = c_{11} + p_{21} c_{21} + p_{31} c_{31} \qquad \hat{h}_{11} = h_{11}$$

$$\hat{c}_{12} = c_{12} + p_{21} (c_{21} + h_{21}) + p_{31} c_{31} \qquad \hat{h}_{21} = h_{12}$$

$$\hat{c}_{13} = c_{13} + p_{23} c_{23} + p_{32} (c_{32} + h_{32}) \quad \hat{h}_{13} = h_{13}$$

2. Use the W-W algorithm on the end-item and components $2, 3, \ldots, N$ in that order with a component's demand being dependent on the lot-sizes of its immediate successor component. Setup costs are adjusted in the manner described at the end of section 4. Thus in periods t for which y_{it} is fixed at 0 s_{it} is set to infinity inside the W-W algorithm. In periods t for which y_{it} is fixed at 1, s_{it} is set to zero inside the W-W algorithm and the original value of s_{it} added to the W-W solution. For example, with $N = T = 3$, $s(2) = 1$, $s(3) = 2$, then when y_{22} and y_{33} are fixed at 0 and y_{11}, y_{12} and y_{32} are fixed at 1 then we use the following modified costs:

$$\hat{s}_{11} = 0 \qquad \hat{s}_{12} = 0 \qquad \hat{s}_{13} = s_{13}$$

$$\hat{s}_{21} = s_{21} \qquad \hat{s}_{22} = \text{infinity} \qquad \hat{s}_{23} = s_{23}$$

$$\hat{s}_{31} = s_{31} \qquad \hat{s}_{32} = 0 \qquad \hat{s}_{33} = \text{infinity}$$

$$\hat{c}_{it} = c_{it} \text{ and } \hat{h}_{it} = h_{it} \text{ for } i = 1, 2, 3 \text{ and } t = 1, 2, 3$$

and add the quantity $s_{11} + s_{12} + s_{32}$ to the sum of the costs of the W-W solutions.

5. THE NODE SEPARATION STRATEGY

Our separation strategy is to always separate that node with the least lower bound. We favoured this strategy over "depth first" because initial evidence showed that it probably leads more rapidly to an optimal solution and because it provides tighter guarantees of the distance from optimality of the incumbent if the B&B search has to be prematurely terminated (for example, when the procedure is used heuristically on large problems).

When a node is separated on a given y_{it} variable, then one of its two child nodes fixes y_{it} at the 0/1 value present in the best lower bound W-W solution of the parent node. This first child node will have the same lower bound as the parent node if the same Lagrangian multipliers λ are used. The other child node will fix y_{it} at the 0/1 value contrary to that in the lower bound W-W solution of the parent node. This second child node will have a lower bound at least as large as that of the parent node if the same Lagrangian Multipliers are used. When we iterate on the Lagrangian Multiplier values at both child nodes, then both lower bounds may increase, although in our computational experiments the lower bound of the first child node very often did not increase after the fixed number of subgradient iterations. our variable separation strategy assumes that, on the whole, the second child node will not only still have the larger lower bound, but will also tend to retain its rank position (among all y_{it}) in the size of the gap between the lower bounds of the two child nodes. The reason for this assumption is that, given the parent node to be separeted, our strategy chooses to separate on that y_{it} variable which gives the highest second-child-node lower bound using the parent node Lagrangian Multipliers. This strategy provides maximum fathoming possibilities (both now and if a better incumbent is found) at as high a level as possible. To implement the strategy we could identify that y_{it} which gives the highest second-child-node lower bound by calculating this lower bound for all free y_{it}. This would be very laborious and very little effort we can guess at such a y_{it} by using the following rule: Separate on y^*_{it} where

$$it^* = arg \max_{(i,t)} \left\{ s_{it} \;\middle|\; y_{it} \text{ is a free variable and equal to zero in the lower bound W-W solution at the present node} \right\}$$

If all free variables are set to 1 in the lower bound W-W solution then we use the following alternative rule: Separate on y^*_{it} where

$$it^* = arg \max_{(i,t)} \left\{ s_{it} \;\middle|\; y_{it} \text{ is a free variable and equal to one in the lower bound W-W solution at the present node} \right\}$$

If y^*_{it} is fixed at 0, then this low setup cost cannot be taken advantage of, hence increasing the lower bound.

If we reach an end-node at the bottom of the B&B tree, then all the y_{it} are fixed and we have a large Linear Program (LP) which is time consuming to solve by primal methods. Since no duality gap exists (in contrast to non-end-nodes) we can continue to approach the LP solution value from below through the Lagrangian Relaxation lower bounding procedure, stopping after a certain number of subgradient iterations if the node has not been fathomed. We found from experience that the end-node upper bound in many cases gives the LP solution value. Thus we can often closely approach the LP solution value from above and below with considerably less additional effort than directly solving the LP using the primal simplex method. The disadvantage of this method is that it is possible that at some end-nodes we end up with upper and lower bounds that are some distance apart. If an end-node solution value is substancially more than the MILP's solution value, then the end-node will eventually be fathomed through its lower bound. However when an end-node solution value is optimal or near-optimal, then the end-node might never be fathomed, depending on the tightness of its lower bound. Our B&B strategy could in some cases never decisively conclude its search for the optimal solution, but instead would terminate by presenting the incumbent solution and a list of end-nodes,all candidates for the optimal solution and all with very similar lower bounds. We have not yet investigated the tradeoff between the advantage of not having to solve end-node LPs exactly and the disadvantage of terminating with perhaps more than one candidate optimal solution. The nature of the tradeoff depends of course on the efficiency of the method used to solve the end-node LPs.

6. THE CASE OF NON-INCREASING PRODUCTION COSTS

The above general separation strategy provides optimal solutions within feasible computing time for problems with up to about five components over 12 time periods. However there exists a special case where we are able to arrive at optimal solutions within feasible computing time for larger problems than in the general case. The special case is that of component setup and unit production costs that are non-increasing over time (for example, costs incurred in the future that are discounted to present values). It has been shown (Crowston & Wagner, 1973) that in this case there exists an optimal solution such that

$$x_{it} > 0 \quad \text{implies} \quad x_{s(i),t} > 0 \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

or equivalently

$$y_{it} = 1 \quad \text{implies} \quad y_{s(i),t} = 1 \qquad \begin{matrix} i = 1 \cdots N \\ t = 1 \cdots T \end{matrix}$$

Thus component i produces in period t only if its successor component is also produced. We can adjust our separation strategy to search for such an optimal solution. If we choose to separate on y_{it} then at the node which fixes y_{it} at 1 we can also fix y_{jt} at 1 for all successors to component i. Hence we choose to separate on y^*_{it} where

$$it^* = arg \max_{(i,t)} \sum_j s_{jt} \text{where the sum } \textstyle\sum_j \text{ is over the set}$$

$$\left\{ j \in R(i) \mid y_{jt} \quad \begin{matrix} \text{is free and equal to zero in the W-W} \\ \text{lower bound solution at the parent node} \end{matrix} \right\}$$

In words, we try to force up as much as possible the lower bound of the node at which y_{it} is fixed at 1, as in our strategy for the general costs case. If all free variables are set to 1 in the lower bound W-W solution then we search for a free y_{it} starting at as high a level as possible in the product structure tree and then with as large a value of t as possible. Fixing this y_{it} to 0 we can also fix y_{jt} to 0 for all free y_{jt} such that $j \in Q(i)$, the set of all predecessor components of i, thus depriving the W-W lower bound solution of the opportunity of using the setup costs s_{jt} for all $j \in Q(i)$. The variable y_{it} is chosen at as high a level as possible in the product structure tree so as to maximize the number of additional components that can be fixed at 0, and then with as large a value of t as possible so as to minimize the s_{it} values.

7. COMPUTATIONAL TESTS

The computational tests were programmed in C and executed on an 80286-based microcomputer with a mathematical coprocessor.

For the general time-varying costs case, we experimented with items with $N = 5$ and $N = 10$ components over a 12-period planning horizon (we were unable to experiment with more complex items due to a scarcity of computing time). For each item 5 runs were carried out using data randomly sampled from the following probability distributions: s_{it} from $U(0, 1000)$, vc_{it} from $U(0.5, 2.0)$, e_{it} from $U(0.1, 0.4)$, and d_t from $U(0, 2000)$. Without loss of generality, we set $r_{i,s(i)} = 1$ for $i = 2, \ldots, N$.

For the case of costs that do not increase over time, 15 runs each were carried out on items with $N = 5$, 10, 15, and 20 components. Of each set of 15 runs, 5 runs were carried out on a totally serial product structure with $s(i) = i - 1$ for $i = 2, \ldots, N$ (i.e. with N levels), 5 runs on an almost flat structure with $s(i) = 1$ for $i = 2, \ldots, N$ (i.e. with just two levels), and 5 runs on a typical intermediate structure. The e_{it} and d_t data were randomly sampled from $U(0.1, 0.4)$ and $U(0, 2000)$ respectively as in the general costs case. However only the period 1 production costs s_{i1} and vc_{i1} were randomly sampled, with the subsequent costs for periods $t = 2, \ldots, N$ being calculated as $s_{it} = \beta^{t-1} s_{i1}$ and $vc_{it} = \beta^{t-1} vc_{i1}$ respectively, where $\beta < 1$ is a per-period discount factor. In order to retain comparability with the data of the general costs case, s_{i1} and vc_{i1} were sampled from $U(500, 1000)$ and $U(1.0, 2.0)$ respectively and $\beta = 0.95$ was used.

In all tests, 200 subgradient iterations were carried out during the lower bounding procedure at the root B&B node (where all y_{it} are free). At subsequent nodes 20 subgradient iterations were performed, beginning with the best Lagrangian multipliers λ_{it} inherited from the parent node. At the B&B end-nodes, 100 iterations were performed in order to arrive close to the end-nodes's primal solution.

Figure 2 shows the test results for the general costs case. Note that only once was the optimal solution obtained in reasonable computing time. In the other 9 tests, the B&B search process had to be prematurely terminated after several hours leaving us with an incumbent solution whose guaranteed precision was still a mean two percentage points away from optimality having increased by only one percentage point on average.

In contrast Figure 3 shows the remarkable improvement in the case of non-increasing discounted production costs. The initial incumbent solution was on average within 0.17% of optimality and always within 1% of optimality. In addition about 3 times out of 4 the root node lower bound was the value of the optimal solution. In other words there seems to exist no duality gap in the discounted costs case. The result is that an optimal solution is found very quickly. These observations make it difficult to judge the utility of our separation strategy that was derived from the characterization of an optimal solution. Certainly the strategy caused the immediate fathoming of many brother nodes, leading the B&B seach rapidly to an optimal solution. However a strategy that is simpler to implement, such as that in the general costs case, might also lead to an optimal solution after the same number of nodes and perhaps in less CPU time.

Cmpts	Levels	% Precision 1st Incumbent	% Precision Last Incumbent	B & B nodes	Time
5	3	0.64*	zero	57	340
5	3	1.88	1.36	857	5110
5	3	4.41	3.41	1209	7216
5	3	3.00	2.34	1187	7053
5	3	2.47	0.94	1256	7381
10	4	2.65	1.30	967	10821
10	4	3.25	3.20	949	10800
10	4	3.25	0.20	1145	10810
10	4	4.63	3.97	931	10810
10	4	3.91	3.11	893	10801

* The optimal solution had the same value as the lower bound of the root node, i.e. no duality gap existed

Figure 2. Test Results with General Production Costs

Cmpts	Levels	% Mean Precision of 1st Incumbent	Mean no. of B & B nodes	Mean Time
5	2	0.20	46	217
5	3	0.33	74	335
5	5	0.05	50	219
10	2	0.43	146	1668
10	4	0.14	83	809
10	10	0.18	158	1008
15	2	0.22	275	6005
15	5	0.21	124	2179
15	15	0.03	91	1359
20	2	0.27	208	5691
20	6	0.07	150	3061
20	20	0.01	102	1913

Figure 3. Test Results with Discounted Production Costs

8. CONCLUSION

Taking into account that the tests were carried out on a microcomputer, it is difficult to compare our computing times in the discounted costs case with the best results in the literature, namely those of Afentakis et al (1984) who considered only the constant costs case and programmed in Fortran on an IBM 3032 machine. Given the recognized complexity of MSLS problems (Bahl et alli, 1987) and the fact that each 20-component problem above has 240 zero-one variables, our discounted costs results are encouraging and deserve more extensive testing on a faster computer. The results of the general costs case are not so encouraging and constitute a continuing challenge.

Several issues need further investigation. First, under what circumstances does no duality gap exist? A Branch-and-Bound search is unnecessary if we know previously that there is no duality gap to close. Secondly, why did our upper bounding procedure produce such good bounds in the discounted case compared to the relatively loose bounds it gave in the general costs case. Tighter upper bounds could significantly improve the performance of our general costs strategy by more rapidly pushing down the value of the incumbent solution. Finally, as discussed in section 5, an efficient method to solve the end-node LPs should be developed in order to avoid terminating the B&B search with candidate optimal solutions only.

9. REFERENCES

AFENTAKIS P. (1987), A Parallel Heuristic Algorithm for Lot-Sizing in Multi-Stage Production Systems, Institute of Industrial Engineers Transactions, March 1987, pp. 34-42

AFENTAKIS P., GAVISH B. & KARMARKAR U. (1984), Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multi-Stage Assembly Systems, Management Sci., Vol. 30, No. 2, pp. 222-239

BLACKBURN J.D. & MILLEN R.A. (1982), Improved Heuristics for Multi-Stage Requirements Planning Systems, Management Sci., Vol. 28, No. 1, pp. 44-56

BAHL H.C., RITZMAN L.P. & Gupta J.N.D. (1987), Determining Lot-Sizes and Resource Requirements: A Review, Operations Research, Vol. 35, No. 3, pp. 237-249

CAMERINI P.M., FRATTA L. & MAFFIOLI F. (1975), On Improving Relaxation Methods by Modified Gradient Techniques, Mathematical Programming Study 3, pp. 26-34

CLARK A.J. & SCARF H. (1960), Optimal Policies for a Multi-Echelon Inventory Problem, Management Sci., Vol. 6, pp. 475-490

CROWSTON W.B. & WAGNER M.H. (1973), Dynamic Lot-Size Models for Multi-Stage Assembly Systems, Management Sci., Vol. 20, No. 1, pp. 517-527

CROWSTON W.B., WAGNER M.H. & WILLIAMS J.F. (1973), Economic Lot-Size Determination in Multi-Stage Assembly Systems, Management Sci., Vol. 19, No. 5, pp. 517-527

DE MATTIES J.J. (1968), An Economic Lot-Sizing Technique, IBM Systems Journal, Vol. 7, pp. 30-48

DENARDO E.V. (1982), Dynamic Programming: Models and Applications, Prentice-Hall, Eaglewood Cliffs, NJ, USA

FISHER M.L. (1981), The Lagrangian Relaxation Method for Solving Integer Programming Problems, Management Sci., Vol. 27, No. 1, pp. 1-18

GRAVES S.C. (1981), Multi-Stage Lot-Sizing: An Iterative Procedure, pp. 95-109 in TIMS Studies in the Manegement Sciences, Vol. 16, Multi-Stage Production/Inventory Control Systems: Theory and Practice, L.B. Schwarz (ed), North-Holland, Amsterdam

GROFF G.K. (1979), A Lot-Sizing Rule for Time-Phased Components Demand, Production and Inventory Management, Vol. 20, 2nd Quarter, pp. 47-53

MCIAREN B.J. & WHYBARK D.C. (1976), Multi-Level Lot-Sizing Procedures in a Material Requirements Planning Environment, Discussion Paper No. 64, Division of Research, School of Business, Indiana University

SCHWARZ L.B. & SCHRAGE L. (1975), Optimal and System Myopic Policies for Multi-Echelon Production/Inventory Assembly Systems, Management Sci., Vol. 21, No. 11, pp. 1285-1294

SILVER E.A. & MEAL H.C. (1973), A Heuristic for Selecting Lot-Size Quantities for the Case of a Deterministic Time-Varying Rate and Discrete Opportunities for Replenishment, Production and Inventory Management, Vol. 14, 2nd Quarter 2, pp. 64-74

STEINBERG E. & NAPIER H.A. (1980), Optimal Multi-Level Lot-Sizing for Requirements Planning Systems, Management Sci., Vol. 26, No. 12, pp. 1258-1271

ZANGWILL W.I. (1969), A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot-Size Production System - A Network Approach, Management Sci., Vol. 15, No. 9, pp.506-527

WAGNER H.M. (1960), Dynamic Problems in the Theory of the Firm, Naval Research Logistics Quarterly, Vol. 7, No. 7, pp. 7-12

WAGNER H.M. & WHITHIN T.M. (1958), Dynamic Version of the Economic Lot-Sizing Model, Management sci., Vol. 5, No. 1, pp. 89-96