THE DETERMINISTIC MULTI-ITEM DYNAMIC LOT SIZE PROBLEM WITH JOINT BUSINESS VOLUME DISCOUNT

Jiefeng Xu

Delta Technology Inc., 1001 International Blvd., Atlanta, GA 30354

Leonard L. Lu

AT&T Laboratories, 101 Crawfords Corner Road, Holmdel, NJ 07733

Fred Glover

Graduate School of Business, University of Colorado at Boulder, Boulder, CO 80309

Abstract

This paper considers the multi-item dynamic lot size model where joint business volume discount is applied for all items purchased whenever the total dollar value of an order reaches a certain level. Multi-item discounts are prevalent in practical applications, yet the literature has only considered limited instances of single-item models. We establish the mathematical formulation and design an effective dynamic programming based heuristic. Computational results disclose our approach obtains high quality solutions that dominate the best known heuristic for the simplified one-item case, and that proves vastly superior to the state-of-the-art CPLEX MIP code for the multi-item case (for which no alternative heuristics have been devised). We obtained significantly better solutions than CPLEX for the more complex problems, while running from 4,800 to over 100,000 times faster. Enhanced variants of our method improve these outcomes further.

Keywords: lot size model, business volume discount, heuristic, dynamic programming.

Published in *Annals of Operations Research*, **96** (2000) 317-337, J.C. Baltzer.

1. Introduction

Cost effectiveness has been a key factor underlying success in logistics management, and usually a vendor's price discount offer is an important factor in purchasing considerations. In traditional inventory models, to keep price-competitive, a vendor offers discounts based on the quantity ordered. Buyers attempt to order an item in large quantities in spite of incurring an associated storage cost, so as to dilute the high one time setup ordering cost and take advantage of the discount provided by the vendor. In the case of one item, the traditional quantity discount models have been studied extensively in the literature, for both EOQ type models and lot sizing models (see Hadley and Whitin 1963, Johnson and Montgomery 1974, Buffa and Miller 1979, Silver and Peterson 1985, Federgruen and Lee 1990, Bregman 1991, and Bregman and Silver 1993).

In the case of multiple items, the discount can be based either on the order quantity of each item individually, or on the total dollar value of all items purchased. The first type of discount is virtually the same as the quantity discount model. The second type of discount, referred to as the Joint Business Volume Discount, can be advantageous to both buyers and vendors. For the buyer, it can reduce the number of vendors utilized and strengthen the purchasing power and negotiation position. For the vendor, it can simplify the discount schedules and promote more balanced sales over multiple products. Sadrian and Yoon (1992) compared the practical impact of the joint business volume discount with those of traditional quantity discounts, and concluded that the joint business volume discount is more realistic for both vendors and buyers. They presented a successful implementation of a decision support system based on the joint business volume discount concept, and reported savings of millions of dollars in several local telephone companies (Sadrian and Yoon 1994). However, their model is static and fails to include important aspects such as inventory costs and time-varied parameters.

In real world applications, a vendor usually provides two different unit prices for each item: an annual commitment price, and an as-ordered price. Schedules for discounts are also provided separately on commitment and as-ordered bases. When an item is purchased on a commitment basis, the buyer makes a contract with a vendor in advance to purchase a certain number of units of the item from that vendor, and this quantity should be purchased during the year regardless of the possible changes in buyer's demand and/or budget. When an item is purchased on an as-ordered basis, the buyer is not obliged to buy any particular quantity.

The as-ordered unit prices are usually higher than those of commitment prices, and this provides an incentive for the buyer to purchase items on a commitment basis. On the other hand, the buyer would like to develop a flexible purchasing plan that can be changed if the forecasted demand changes during the year. Consequently, the buyer faces the conflicting objectives of reducing the purchasing costs and developing a flexible purchasing plan, and needs to test different commitment percentages for the forecasted demand and examine the cost differences. It can be seen that for a fixed commitment percentage, it is not difficult to evaluate the discount schedules of commitment basis. However, it is not easy to apply the discount schedules of as-ordered basis, and the outcome will depend on the order frequencies and order quantities.

While there is some discussion in the literature of the dynamic lot sizing model with quantity discounts (see Federgruen and Lee 1990, Bregman 1991, and Bregman and Silver 1993), these results can only be used for one-item cases. To the authors' knowledge, there is no literature available for multiple-item cases, and this paper provides the first investigation of the multi-item dynamic lot sizing model that incorporates joint business volume discounts.

We describe the problem more precisely as follows. Consider an inventory model with finite numbers of time periods and items, with demand known for each item at each time period. The setup cost for each item occurs at the time the item is ordered, and inventory costs are applied only to the inventory carried at the end of each time period. Each item has a unit purchasing price. If the total dollar value of items purchased during a time period reaches or exceeds the critical value of discount level, an all-unit discount rate is realized for all items ordered during that period. Backorders, initial and trial inventories are not allowed. The problem is to determine when and how much to order for each item, with the objective of minimizing the total costs made up of purchasing costs, inventory costs and setup costs. For the sake of simplicity, we use the term "discount" to mean "joint business volume discount" in the rest of this paper. It should be pointed out that though our model assumes only one price break point, it can be easily extended to the case with multiple price break points.

The rest of this paper is organized as follows. Section 2 formulates the model as a mixed integer programming model. Section 3 describes a basic heuristic procedure based on dynamic programming, and discusses the extension to the case of multiple price break points. Section 4 derives some alternative decision rules that can be used in the basic heuristic procedure. Section 5 presents two fine tuned methods that further improve the quality of the heuristic solution, and a speedup method that significantly improves the speed of the algorithm. Section 6 reports computational results on two sets of randomly generated test problems which disclose the effectiveness of our procedure. Our algorithm performs very favorably compared with the optimal algorithm and the best available heuristic for the single-item cases, and produces far superior solutions for the multi-item instances than a standard branch and bound algorithm (which is the only alternative currently available). Section 7 summarizes with concluding remarks.

2. Mathematical Formulation

We formulate the problem as a mixed integer programming model as follows. First, we define:

Constants

d_{it} : demand of item i at period t;

S_i : one time setup cost of item i;

v_i : price of item i;

h_i : unit inventory holding cost of item i;

β : critical dollar value (breakpoint) of the discount level;

α : price discount rate.

Variables

x_{it} : the quantity of item i ordered at period t;

δ_{it} : a binary variable equal to 1 if and only if $x_{it} > 0$;

I_{it} : the inventory of item i in period t;

y_{it} : the quantity of item i ordered at period t that can be purchased with discount;

z_t : a binary variable equal to 1 if and only if period t is the discount period.

The model is then

Model 1

$$\text{minimize} \quad \sum_{i=1}^{m}\sum_{t=1}^{n}(S_i\delta_{it} + h_i I_{it} + v_i x_{it} - \alpha v_i y_{it}) \tag{1}$$

subject to:

$$x_{it} + I_{i,t-1} - I_{it} = d_{it}, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{2}$$

$$x_{it} \le M\delta_{it}, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{3}$$

$$\sum_{i=1}^{m} v_i y_{it} \ge \beta z_t, \quad t = 1,\ldots,n \tag{4}$$

$$y_{it} \le M z_t, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{5}$$

$$y_{it} \le x_{it}, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{6}$$

$$I_{i,0} = I_{i,n} = 0, \quad i = 1,\ldots,m \tag{7}$$

$$x_{it} \ge 0, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{8}$$

$$I_{it} \ge 0, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{9}$$

$$y_{it} \ge 0, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{10}$$

$$\delta_{it} \in \{0,1\}, \quad i = 1,\ldots,m, \quad t = 1,\ldots,n \tag{11}$$

$$z_t \in \{0,1\}, \quad t = 1,\ldots,n. \tag{12}$$

In this formulation, the objective function (1) seeks to minimize the total setup costs, inventory costs and purchasing costs. Constraint (2) embodies the demand balance requirements, while (3) expresses the incidence relationship of δ_{it} and x_{it}, where M is a "sufficiently big" number (i.e. $M > \max_i\{\sum_{t=1}^{n} d_{it}\}$). Clearly, $\delta_{it} = 1$ if and only if $x_{it} > 0$. Constraints (4) and (5) specify the discount condition. If period t is the discount period ($z_t = 1$), then the joint purchase volume $\sum_{i=1}^{m} v_i x_{it}$ must equal or exceed β; if period t is not a discount period ($z_t = 0$), then no items at period t can be purchased at the discount prices ($y_{it} = 0$).

Constraint (6) connects the discount quantity with the quantity ordered and ensures that the discount is applied to all units. That is, at each discount period t where $z_t = 1$ and $y_{it} > 0$, the quantity with discount, y_{it}, will equal the total quantity ordered at this period, x_{it}, because of the minimization of the objective function. Thus, $y_{it} = x_{it}$ when $z_t = 1$, and otherwise $y_{it} = 0$. Finally, constraint (7) indicates that there are no initial and final inventories, and constraints (8) – (12) stipulate the non-negativity and discrete requirements for the variables.

This model is very difficult to solve using latest state-of-the-art integer programming techniques, as demonstrated by computational testing subsequently reported. Even when all z_t's are fixed, the relaxed problem is still NP-hard. The existence of constraint (4) creates an interlinking among the different items and complicates the lot size model. Thus, the design of an efficient heuristic is of paramount importance and constitutes the key focus of our research.

3. A Dynamic Programming Based Heuristic

We develop a dynamic programming based heuristic in this section. First, we relax the problem by introducing lower bounds, and separate the difficult Model 1 into m single-item lot size models; then, we identify important properties of solutions to the single-item lot size model, and develop basic heuristics based on these properties.

3.1. Solution Properties. Recall that the fundamental difficult constraint in the foregoing *Model 1* is the existence of the constraint (4), which establishes the discount condition. It is well known that without discount, the whole model can be easily solved by solving m separate Wagner-Whitin (1958) lot size models. We attempt to replace the difficult constraint (4) by simpler conditions and disregard the discount benefit in the objective function so that the revised model can be easily solved by taking advantage of the simple solution approach for the variant of the Wagner-Whitin model.

To create such a strategy, we introduce a non-mandatory lower bound L_{it} for x_{it} to replace the constraint (4):

$$x_{it} \geq L_{it} \quad \text{if} \quad x_{it} > 0, \qquad i = 1, \ldots, m, \qquad t = 1, \ldots, n. \tag{13}$$

The value of L_{it} is determined by a specific procedure (which will be described later) and must satisfy the following conditions:

(C1) $L_{it} \le \sum_{k=t}^{n} d_{ik}$.

(C2) For each period t, either $L_{it} = 0$ for $i = 1, \dots, m$ (in which case we call t a nondiscount period) or $\sum_{i=1}^{m} v_i L_{it} \ge \beta$ (in which case we call t a discount period).

Now, for given lower bounds L_{it}, we determine the values of z_t as follows:

$$z_t = \begin{cases} 0, & \text{if } L_{it} = 0 \text{ for all } i \\ 1, & \text{otherwise.} \end{cases} \qquad t = 1, \dots, n.$$

Since $y_{it} = z_t x_{it}$, we can now remove the variables y_{it}. Furthermore, note that $\sum_{i=1}^{m} \sum_{t=1}^{n} v_i x_{it}$ is constant due to constraints (2) and (7), it can thereby be dropped from (1). Consequently, using the constaint (13) we separate the difficult Model 1 into m single-item lot size models as follwos:

Model 2 (Single-Item Lot Size Model for item i)

$$\text{minimize} \quad \sum_{t=1}^{n} (S_i \delta_{it} + h_i I_{it} - \alpha z_t v_i x_{it})$$

subject to:

$$\begin{aligned}
x_{it} + I_{i,t-1} - I_{it} &= d_{it}, & t &= 1, \dots, n \\
x_{it} &\le M \delta_{it}, & t &= 1, \dots, n \\
x_{it} &\ge L_{it} \quad \text{if} \quad x_{it} > 0, & t &= 1, \dots, n \\
I_{i,0} = I_{i,n} &= 0, & & \\
x_{it} &\ge 0, & t &= 1, \dots, n \\
I_{it} &\ge 0, & t &= 1, \dots, n \\
\delta_{it} &\in \{0, 1\}, & t &= 1, \dots, n.
\end{aligned}$$

We now present some important relationships satisfied by the optimal solution of Model 2. These relationships will allow us to develop an efficient method for solving the Model 2, which we will then apply to the original problem of Model 1. For an item i, a period t is called a regeneration period if $I_{i,t-1} = 0$. We have the following proposition for the Model 2:

Proposition: Consider an optimal solution of Model 2 such that t_1 and t_2 $(t_2 > t_1)$ be two adjacent order periods for item i, and assume that t_1 is a regeneration period (i.e., $I_{i,t_1-1} = 0$). Then in this optimal solution, we have $x_{it_1} = \max\{L_{it_1}, \sum_{t=t_1}^{t_2-1} d_{it}\}$.

Proof: We consider two cases: t_1 is either a non-discount period or a discount period.

CASE 1: t_1 is a non-discount period. Thus, $L_{it_1} = 0$.

Since t_1 and t_2 are two adjacent order periods, $x_{it} = 0$ for $t = t_1 + 1, \ldots, t_2 - 1$. Now we show that $I_{i,t_2-1} = 0$. If this is not true, then we can construct a new feasible solution $(\bar{x}, \bar{I})$ such that $\bar{x}_{it_1} = x_{it_1} - I_{i,t_2-1}$ and $\bar{x}_{it_2} = x_{it_2} + I_{i,t_2-1}$. The new policy therefore has the same setup costs as the current solution, but smaller inventory costs. This contradicts the assumption. Consequently, $x_{it_1} = \sum_{t=t_1}^{t_2-1} d_{it}$.

CASE 2: t_1 is a discount period. We establish the result for two subcases:

Subcase 1: $L_{it_1} \leq \sum_{t=t_1}^{t_2-1} d_{it}$.

In this case, since there is no order placed between periods t_1 and t_2, we must order enough quantity for item i to meet the demands from period t_1 to $t_2 - 1$. Thus, following directly from CASE 1, we have $x_{it_1} = \sum_{t=t_1}^{t_2-1} d_{it}$.

Subcase 2: $L_{it_1} > \sum_{t=t_1}^{t_2-1} d_{it}$.

In this case, by the lower bound condition, we must order at least L_{it_1} for item i. If $x_{it_1} > L_{it_1}$ in the current optimal solution, then we can construct a new solution as $\bar{x}_{it_1} = L_{it_1}$ and $\bar{x}_{it_2} = x_{it_2} + x_{it_1} - L_{it_1}$. It is easy to see that the new solution has the same setup costs but lower inventory costs than the current solution. This contradicts the optimality of the current solution. ◇

The proposition indicates that for two adjacent order periods t_1 and t_2, if t_1 is a regeneration period, then we have $x_{it_1} = \max\{L_{it_1}, \sum_{t=t_1}^{t_2-1} d_{it}\}$. Since $L_{it_1} > \sum_{t=t_1}^{t_2-1} d_{it}$ may occur, it may be the case that $I_{i,t_2-1} = x_{it_1} - \sum_{t=t_1}^{t_2-1} d_{it} > 0$, and in which case the result of the proposition will not be valid for the next two adjacent order periods t_2 and t_3 (i.e., $x_{it_2} < \sum_{t=t_2}^{t_3-1} d_{it}$ may occur). Therefore, in order to use the proposition periodically for all pairs of adjacent order periods, we update d_{it_2} as $d_{it_2} - I_{i,t_2-1}$ (note that the resulting d_{it_2} may be negative). With these updated demands, we now have the order quantity $x_{it_1} = \max\{L_{it_1}, \sum_{t=t_1}^{t_2-1} d_{it}\}$ for any pair of adjacent order periods.

We now present the following dynamic programming recursion as a heuristic procedure for solving the Model 2.

Procedure A

Step 0 : Set $g(i,1)=0$ and $t_2=1$.

Step 1 : Let $t_2=t_2+1$ and calculate

$$g(i,t_2)=\min_{1\le t_1<t_2}\{g(i,t_1)+f(i,t_1,t_2)\}, \tag{14}$$

where for a pair of adjacent order periods t_1 and t_2, $f(i,t_1,t_2)$ is the total cost for item i incurred from period t_1 to period t_2-1 (including setup cost, inventory cost and the negative discount cost if applicable), i.e.,

$$f(i,t_1,t_2)=S_i+\sum_{t=t_1}^{t_2-1}h_iI_{it}-\alpha z_{t_1}v_ix_{it_1}\ . \tag{15}$$

The order quantity in the expression (15) is $x_{it_1}=\max\{L_{it_1},\sum_{t=t_1}^{t_2-1}d_{it}\}$, and the inventories are calculated as follows:

$$I_{it_1}=x_{it_1}-d_{it_1},\quad \text{and}\quad I_{it}=I_{i,t-1}-d_{it}\quad \text{for}\quad t=t_1+1,\ldots,t_2-1,$$

where d_{it_1} is the updated demand associated with $g(i,t_1)$ (see *Step 2* for the demand updating). If $x_{it_1}=0$, set $f(i,t_1,t)=\infty$ (so the period t_1 will not be an order period).

Step 2 : If $t_2=n+1$, then stop.

Otherwise, update demand $d_{it_2}=d_{it_2}-I_{i,t_2-1}$ for $g(i,t_2)$, and go to *Step 1.*

It can be seen that $g(i,t)$ is the total cost for item i incurred from period 1 to period $t-1$. Notice that the order quantity $x_{it_1}=\max\{L_{it_1},\sum_{t=t_1}^{t_2-1}d_{it}\}$ depends on the decisions taken prior to the period t_1. Therefore, the value of $f(i,t_1,t_2)$ also depends on the decisions taken prior to the period t_1. Since for each item i, the procedure takes $O(n^2)$ steps, the total complexity of Procedure A for solving m single-item lot size models of Model 2 is $O(mn^2)$. Using this outcome, we now seek a heuristic to solve the difficult Model 1 by designating the discount period and the associated lower bounds. Such a procedure is described in the following subsection.

3.2. The Basic Heuristic. We exploit the preceding results by designing a dynamic programming based heuristic to solve the multi-item lot size inventory model with discount. In this basic heuristic, we determine and fix the discount periods, find the associated lower bounds, and then use Procedure A to find the resulting solutions. Based on the solutions obtained, we determine if the given period should be a discount period or not and identify the next "most appropriate" discount period. The heuristic iterates until all periods are determined to be discount or non-discount periods. We describe the heuristic as follows:

Dynamic Programming Based Heuristic (DPH)

Step 0 : For each period t, we create three types of status: free, discount, non-discount. The status "free" is treated the same as the"non-discount" status, except that it has the future chance to be changed to the "discount" status. Both the "discount" and "non-discount" status are fixed, and cannot be changed later. Let n_1 be the number of discount periods, n_2 be the number of fixed discount or non-discount periods and n_3 be the maximum number of discount periods. Initially, all periods are set to "free" and

$$n_1 = n_2 = 0, \quad n_3 = \lfloor \sum_{t=1}^{n} \sum_{i=1}^{m} v_i d_{it} / \beta \rfloor.$$

Designate $x = \{x_{it}\}$ to be the current solution obtained by Procedure A and c to be the total true cost (including the set-up cost, inventory cost and the real discount benefit) corresponding to x. Likewise, define $\bar{x} = \{\bar{x}_{it}\}$ to be the best solution found so far and $\bar{c}$ to be the total true costs corresponding to $\bar{x}$.

Step 1 : Start with all periods "free" and all corresponding lower bounds zero. Use Procedure A to find the resulting solutions. Let $\bar{c} = c$ and $\bar{x} = x$.

Step 2 : Calculate $V_t = \sum_{i=1}^{m} v_i x_{it}$ for $t = 1, \ldots, n$. Let

$$t' = \underset{t \text{ is free}}{\mathbf{argmax}} \{V_t\}.$$

Assign t' the discount status. Identify the lower bound $L_{it'}$ for each item i using a specific procedure (which will be discussed in next section). If no qualified $L_{it'}$ found, then fix t' to the non-discount status and go to *Step 6*. Otherwise determine x and c using Procedure A.

Step 3 : Calculate new $V_{t'} = \sum_{i=1}^{m} v_i x_{it'}$. If $V_{t'} \geq \beta$, then go to *Step 4*. Otherwise, go to *Step 5*.

Step 4 : If $c < \bar{c}$, then let $\bar{x} = x$, $\bar{c} = c$, $n_1 = n_1 + 1$, and go to *Step 6*. Otherwise, assign t' the non-discount status, set the corresponding lower bounds as $L_{it'} = 0$, and go to *Step 6*.

Step 5 : Assign t' the non-discount status, and set the corresponding lower bounds to $L_{it'} = 0$. If $c < \bar{c}$, then let $\bar{x} = x$ and $\bar{c} = c$.

Step 6 : Let $n_2 = n_2 + 1$. If $n_1 < n_3$ and $n_2 < n$, go to *Step 2*. Otherwise, go to *Step 7*.

Step 7 : Terminate.

This heuristic initially assigns all periods "free" status in *Step 0*. Then it picks a free period with the largest purchase volume V_t as a candidate discount period in *Step 2*. If this candidate fails to meet the discount condition, we assign it a non-discount status in *Step 5*. If this candidate satisfies the discount condition but cannot improve the solution, then we also assign it a non-discount status in *Step 4*. Only those candidates that are "legitimate" and reduce the objective function will finally be accepted as a discount period. In *Step 4* and *Step 5*, we compare the cost of each legal move with the cost of the best solution available and update the best solution if the move reduces the total cost. The heuristic terminates when all periods are fixed as either discount or non-discount periods, or when we can not create any more discount periods. The overall computational complexity is $O(mn^3)$.

Note that even if a period is assigned a discount status, the discount benefit is not "compulsory", and it can decide if it is cost-effective to take advantage of the discount. The two associated scenarios for discount period t are: we can order item i to achieve a discount by satisfying $x_{it} \geq L_{it}$, or we do not order the item in this period ($x_{it} = 0$). The order policy is determined by Procedure A. Since we solve the multi-item problem separately for each item, the joint discount condition needs to be checked in *Step 3* to validate its discount status.

It should be pointed out that though the foregoing DPH is presented for one price breakpoint, the case of multiple price breakpoints is a straight forward extension. To handle the model with K discount levels, we need to extend the status of each period to free, non-discount, discount 1, ..., discount K. We also need to determine the multiple lower bound values as $L_{it}^1, \ldots, L_{it}^K$, corresponding to various price levels. We then attempt to assign the highest possible discount status to a free period with the highest business volume. If this fails for that period, we try to assign it the next possible (lower) discount status and so on until an appropriate discount status or non-discount status is assigned. Hence the multiple discount problem can be solved within a framework that directly extends the simpler DPH approach. However, the effectiveness of the general approach will inevitably decrease as the number of discount levels K increases, due to the greedy nature of the DPH and the difficulty in determining the appropriate lower bound values for multiple price breakpoints.

4. Lower Bound L_{it} Calculations

An important component of this heuristic is the determination of the lower bound values L_{it} for discount periods. There are a variety of ways to determine the lower bound values in order to meet the aggregate discount condition. In this section, we first introduce a simple proportional calculation, then describe several more refined calculations.

Let t' be the current period at which the lower bounds are to be set, $x_{it'}$ the solution before period t is assigned the discount status, and $V_{t'}$ the total purchase volume corresponding to $x_{it'}$. Then, the simplest proportional calculation is described as follows.

Simplest Proportional Calculation (PC1) for Lower Bounds

Step 1 : Calculate $L_{it'} = x_{it'}\beta / V_{t'}$ for $i = 1, \ldots, m$.

PC1 has the appeal of being very simple and intuitive but may produce a number of infeasible lower bounds when $V_{t'} < \beta$. A more elegant and feasible lower bound calculation is presented as follows.

Proportional Calculation (PC2) for Lower Bounds

Step 1 : If $V_{t'} \geq \beta$, then set $L_{it'} = x_{it'}\beta / V_{t'}$ for $i = 1, \ldots, m$, and terminate. Otherwise, go to *Step 2*.

Step 2 : Let O_i be the maximum possible order quantity for item i at period t'. Then we have

$$O_i = \sum_{t=t'}^{n} d_{it} \quad \text{for} \quad i = 1, \ldots, n.$$

If $\sum_{i=1}^{m} v_i O_i \geq \beta$, then set

$$L_{it'} = O_i \beta / \sum_{i=1}^{m} v_i O_i \quad \text{for} \quad i = 1, \ldots, m.$$

Otherwise, the calculation fails to determine lower bound values that satisfy the discount condition. Terminate.

This calculation, while accounting for conditions beyond those considered by PC 1, may produce infeasible lower bounds since O_i is not the true value of the maximum order quantity for item i. If any periods after t' have discount status and lower bound requirements, the order with quantity O_i for item i at period t' may be infeasible. The preceding calculation also fails to take the item price into consideration. We favor an item with higher future business volume $v_i O_i$, by assigning it a larger lower bound, since this item is more likely to meet a large lower bound constraint. The following more refined calculation addresses these issues.

Proportional Calculation Considering Price (PC3) for Lower Bounds

Step 1 : If $V_{t'} \geq \beta$, then set $L_{it'} = \lceil x_{it'} \beta / V_{t'} \rceil$ for $i = 1, \ldots, m,$ and terminate. Otherwise, go to *Step 2*.

Step 2 : Let O_i represent the maximum order quantity for item i at period t'. Then we have

$$O_i = \sum_{t=t'}^{n} (d_{it} - L_{it}) \quad \text{for} \quad i = 1, \ldots, n.$$

If $\sum_{i=1}^{m} v_i O_i < \beta$, then the calculation fails to determine the lower bounds, terminate. Otherwise, go to *Step 3*.

Step 3 : Let

$$i' = \underset{1 \leq i \leq m}{\mathbf{argmax}} \{ v_i (O_i - x_{it'}) \}.$$

Find the period t'' which is the first order period for item i' after t' ($x_{i't''} > 0$). Update $x_{i't'} = x_{i't'} + d_{i't''}$, $x_{i't''} = x_{i't''} - d_{i't''}$, and $V_{t'} = V_{t'} + v_{i'} d_{i't''}$. If $V_{t'} \geq \beta$, then go to *Step 4*. Otherwise, repeat *Step 3*.

Step 4 : Set $L_{it'} = x_{it'} \beta / V_{t'}$ for $i = 1, \ldots, m.$ Terminate.

PC3 considers the item price in *Step 3*. It aggressively assigns a relatively larger lower bound to an item with a higher future business volume. PC3 provides a legitimate lower bound for each item because ordering $L_{it'}$ at period t' is feasible. However, the O_i value in PC3 is limited to the most recent solution $\{x_{it}\}$ and is not the real maximum order quantity for item i. In case of $\sum_{i=1}^{m} v_i O_i < \beta$ (which signals that PC3 could not determine the lower bounds), we may decrease the order for the periods before t' to increase O_i if $I_{i,t'-1} > 0$. To do this, we must determine lower bound quantities that can be "transferred" from the previous orders to later ones, in order to allow the corresponding lower bound constraints and demand constraints to remain satisfied. An advanced calculation which incorporates this kind of adjustment is given below.

Advanced Proportional Calculation (PC4) for Lower Bounds

Step 1 : If $V_{t'} \geq \beta$, then set $L_{it'} = \lceil x_{it'} \beta / V_{t'} \rceil$ for $i = 1, \ldots, m$, and terminate. Otherwise, go to *Step 2*.

Step 2 : Let e_{it} be the quantity of item i which can be transferred from the previous orders to period t, and let t_1 and t_2 be two adjacent order periods, that is,

$$x_{it_1} > 0, \quad x_{it_2} > 0, \quad \text{and} \quad x_{it} = 0 \quad \text{for} \quad t_1 < t < t_2.$$

Thus $e_{it_2} = e_{it_1} + x_{it_1} - \max\{L_{it_1}, \sum_{t=t_1}^{t_2-1} d_{it}\}$.

Start with $t_1 = 1$ and $e_{it_1} = 0$. Find t_2 and calculate e_{it_2}. If $t_2 = t'$, go to *Step 3*. Otherwise, update $t_1 = t_2$ and repeat *Step 2*.

Step 3 : Calculate

$$O_i = e_{it'} + \sum_{t=t'}^{n} (d_{it} - L_{it}).$$

If $V = \sum_{i=1}^{m} v_i O_i < \beta$, then it is impossible to satisfy the discount condition, terminate. Otherwise, go to *Step 4*.

Step 4 : Calculate $L_{it'} = x_{it'} \beta / V_{t'}$ for $i = 1, \ldots, m$. If all $L_{it'} \leq O_i$, all lower bounds are feasible, terminate. Otherwise, go to *Step 5*.

Step 5 : Set

$$L_{it'} = \min\{L_{it'}, O_i\} \quad \text{for} \quad i = 1, \ldots, m,$$

$$V = \sum_{i=1}^{m} v_i(O_i - L_{it'}),$$

$$b = \beta - \sum_{i=1}^{m} v_i L_{it'}.$$

Step 6 : Set $L_{it'} = L_{it'} + (O_i - L_{it'})b/V$ for $i = 1, \ldots, m$. Terminate.

In PC4, the sequence from *Step 1* to *Step 3* calculates the real O_i values, and the sequence from *Step 4* to *Step 6* assigns the lower bounds according to the previous solution $x_{it'}$ and future business volume. The lower bounds are feasible, because $L_{it'} \leq O_i$.

The foregoing four calculations may produce significantly different lower bound values and hence impact the performance of the dynamic programming algorithm described in Section 3. We compare the individual performance of these lower bound calculations in Section 6.

5. Fine Tuned Solution Adjustment Approaches and a Speedup Method

Here we introduce two fine tuned methods to alter the solution obtained by the dynamic programming model for each discount period. The adjustment of solutions is subject to the discount conditions and demand balance conditions. The ideas are based on the following observations: in discount period t_1, if $x_{it_1} > 0$, x_{it_1} is either greater than L_{it_1} or equal to L_{it_1}. In case $x_{it_1} = L_{it_1}$, the lower bound is binding. By Proposition 2, this must occur when $I_{i,t_2-1} = L_{it_1} - \sum_{t=t_1}^{t_2-1} d_{it} > 0$ where t_2 is the next order period after t_1. On the other hand, the existence of some items such that $x_{it_1} > L_{it_1}$, can create some extra "credits" for meeting the discount condition because $\sum_{i=1}^{m} v_i x_{it_1} > \sum_{i=1}^{m} v_i L_{it_1} = \beta$. We can decrease x_{it_1} somewhat for binding items while still meeting the discount condition. In this way, we can reduce the inventory costs for items by reducing the inventory costs for items by reducing their order quantity, but will lose some discount benefits since smaller quantities can enjoy the discount. We describe the first fine tuned method that addresses this tradeoff.

Fine Tuned Post-Solution Method (FT1) for discount period t_1

Step 1 : Let IB be the set of items whose lower bound is binding ($x_{it_1} = L_{it_1}$). Calculate $Credit = V_{t_1} - \beta$.

Step 2 : If $Credit = 0$ or $IB = \phi$, terminate. Otherwise, do the following:
Let $t(i)$ be the next order period after period t_1 for item i. Choose

$$i' = \mathop{\mathbf{argmax}}_{i \in IB} \{h_i(t(i) - t_1) - \alpha v_i + u(i)\alpha v_i\},$$

where $u(i) = 1$ if $t(i)$ is the discount period, and 0 otherwise. Calculate

$$\begin{aligned} q &= \min\{ Credit/v_{i'}, I_{i't(i')-1} \} \\ x_{i't_1} &= x_{i't_1} - q \\ x_{i't(i')} &= x_{i't(i')} + q \\ Credit &= Credit - v_{i'}q \\ IB &= IB - \{i'\} \\ L_{i't_1} &= x_{i't_1}. \end{aligned}$$

Update $I_{i',t}$. Repeat *Step 2*.

FT1 chooses the item with the maximum savings that results from reducing the order quantity by one unit. Note the savings must be positive or else we can achieve a better solution by increasing the order quantity to $\sum_{t=t_1}^{t(i')} d_{i't}$. We reduce the order quantity as much as possible while satisfying the discount condition ($Credit \geq 0$) and the demand balance condition. This process is repeated until no item that yields such a savings can be found or no $Credit$ is left.

The FT1 procedure can be further enhanced by the idea of oscillation. At the end of FT1, we choose some items for which we can increase or reduce the order quantity by a reasonable amount without affecting the next order period. We increase the order quantity for one item and reduce order quantity for another item in order to reduce the total costs. More specifically, we look for two items that respectively yield the maximum and minimum savings when the order quantity is decreased by one unit. Then we find a feasible amount to decrease the first item and increase the second while satisfying the discount condition. We accept the changes if these two changes will reduce total cost. The details are described in the algorithm (FT2) below. Note that FT2 is executed after FT1, thus we start at *Step 3*.

Enhanced Fine Tuned Method (FT2) for discount period t_1 (follow FT1)

Step 3 : Let IF be the set of items such that $I_{i,t(i)-1} > 0$. Find

$$i_1 = \operatorname*{argmax}_{i \in IF} \{h_i(t(i) - t_1) - \alpha v_i + u(i)\alpha v_i\} \quad \text{and}$$

$$i_2 = \operatorname*{argmin}_{i \in IF} \{h_i(t(i) - t_1) - \alpha v_i + u(i)\alpha v_i\},$$

where $u(i) = 1$ if $t(i)$ is the discount period, and 0 otherwise. If $i_1 = i_2$ terminate. Otherwise, calculate

$$q_1 = I_{i_1,t(i_1)-1}.$$

$$q_2 = \max\{d_{i_2,t(i_2)} - I_{i_2,t(i_2)-1}, 0\}.$$

$$V = \min\{q_1 v_{i_1}, q_2 v_{i_2}\}.$$

$$q_1 = \min\{q_1, V/v_{i_1}\}.$$

$$q_2 = \min\{q_2, V/v_{i_2}\}.$$

Step 4 : If $q_1[h_{i_1}(t(i_1) - t_1) - \alpha v_{i_1} + u(i_1)\alpha v_{i_1}] \leq q_2[h_{i_2}(t(i_2) - t_1) - \alpha v_{i_2} + u(i_2)\alpha v_{i_2}]$, terminate. Otherwise, let

$$\begin{aligned} x_{i_1,t_1} &= x_{i_1,t_1} - q_1 \\ x_{i_1,t(i_1)} &= x_{i_1,t(i_1)} + q_1 \\ x_{i_2,t_1} &= x_{i_2,t_1} + q_2 \\ x_{i_2,t(i_2)} &= x_{i_2,t(i_2)} - q_2. \end{aligned}$$

Update I_{i_1t} and I_{i_2t}, and go to *Step 3*.

FT2 terminates if it can not find two desirable items (*Step 3*) or the change cannot result in a saving for total cost (*Step 4*). It is easily understood that the minimum savings/cost of reducing one unit order quantity is exactly the same as the maximum savings/cost of adding one unit order quantity to that item. In FT2, a potential problem may occur if $t(i_1) \neq t(i_2)$ and $t(i_2)$ is a discount period. The discount condition will be violated if there are not enough "credits" at period $t(i_2)$. The solution to this is to add the following remedy to calculate the correct q_2 in *Step 3*:

If $t(i_1) \neq t(i_2)$ and $t(i_2)$ is the discount period, then $q_2 = \min\{q_2, (V_{t(i_2)} - \beta)/v_{i_2}\}$.

Speedup Method (SM)

When a candidate discount period is initiated in our heuristic, the dynamic programming model must be re-solved to evaluate the candidate discount period. If we can preserve some useful information from the previous solution, and thus reduce the work of solving the dynamic model, the heuristic will be more efficient. We propose a speedup method for solving the dynamic programming model in *Step 2* of the DPH. Given the candidate discount period t', the most recent solution x_{it} and the solution $\bar{x}_{it}$ to be determined, we describe the method as follows:

SM: If $I_{i,t'-1} = 0$, let $\bar{x}_{it} = x_{it} \quad \forall t = 1, \ldots, t'-1,$ and solve the Model 2 from period t' to period $n+1$ to determine the remaining $\bar{x}_{it}$.
Otherwise, solve the Model 2 from period 1 to period $n+1$ to determine $\bar{x}_{it}$.

Note that the speedup method based on the partial dynamic programming model is not equivalent to the complete dynamic programming model. In SM, it is assumed that we have to order at period t'. This assumption imposes certain restrictions on the recursion (14), though its occurrence is rare in most applications. The tradeoff between efficiency and solution quality needs to be considered when applying the SM approach.

6. Computational Experiments

Computational tests have been conducted on two randomly generated problem sets of different sizes. These test problems are labeled as "$(m \times n)$". The demand d_{it} is generated uniformly in multiples of ten from the interval [10,100]. The setup cost S_i is generated uniformly in multiples of five in the interval [60,180], while the inventory cost h_i is a random integer in the interval [3,8]. The item price v_i is also a random integer in the interval [1,10]. α is set to 10%. The first set consists of 12 test problems for the single item case. The discount breakpoint β for this set is defined by $\beta = \max\{\lfloor \sum_{t=1}^{n} v_1 d_{1t} \times 2/n \rfloor, v_1 \times dmax\}$ where $dmax$ is maximum demand over all periods. The second set contains 10 problems with three or five items. The discount breakpoint is defined by $\beta = \lfloor \sum_{i=1}^{m} \sum_{t=1}^{n} v_i d_{it} \times 1.6/n \rfloor$. We use PC2 for the basic algorithm (DPH) unless otherwise stated. We select these parameters to provide more tradeoffs between the setup and inventory costs, and the discount benefits, thus to produce more difficult problem instances. The costs reported are total costs according to (1) unless otherwise stated. All data can be obtained from the authors for test and comparison purposes.

We first test our algorithm on a special case of the model — the one item discount problem which is a well-studied problem. Numerous other algorithms exist in the literature. Federgruen and Lee (1990) developed a dynamic programming algorithm which can solve the one item discount problem to optimality. Bergman and Silver (1993) found that a modification of the Silver-Meal heuristic significantly outperforms the other four heuristics in term of the solution quality and computational time. Therefore we compare our algorithm (DPH) with these two leading algorithms on 12 test problems in the first set. Note that the inventory cost expressions in these two algorithms are slightly different from that in our model, so we adjust their algorithms accordingly to test our model. In addition, since all three algorithms can solve the test problems in negligible time with today's computer hardware, we only report the solution costs in Table 1. (The Federgruen and Lee algorithm and the DPH heuristic both have complexity $O(n^3)$ while the Bergman and Silver heuristic has complexity $O(n)$).

Problem	DPH	Federgruen & Lee	Bergman & Silver
(1×5)	2320.0	2320.0	2320.0
(1×10)	4240.0	4240.0	4251.2
(1×20)	12626.5	12626.5	12670.5
(1×25)	7550.0	7550.0	7553.8
(1×30)	7347.0	7347.0	7350.0
(1×40)	15106.0	15106.0	15206.0
(1×50)	15060.0	15060.0	15175.0
(1×60)	20140.0	20140.0	20195.5
(1×70)	11740.0	11740.0	11740.0
(1×80)	26262.0	26262.0	26386.0
(1×90)	21100.0	21100.0	21252.8
(1×100)	39971.0	39924.0	40281.0

Table 1. Computational results on the first problem set

From Table 1 we find that our DPH heuristic can produce very high quality solutions for the single item case. Specifically, compared with the optimal solutions produced by the Federgruen and Lee algorithm, it yields optimal solutions to 11 out of 12 problems and only fails in one remaining case with very small error (around 0.1%). Compared to the Bergman and Silver heuristic, it produces ten better solutions and ties in the remaining two cases. In sum, the results indicate that our DHP approach, though not specially designed for the one item discount problem, can obtain optimal or near-optimal solutions for this problem also.

We next test our DPH method for the multi-item discount problem, for which it is designed. We concentrate all subsequent experiments on the ten test problems in problem set 2. The evaluation of DPH is not as straightforward as in the one item case, since there are no known heuristics available to compare against for the multi-item model. One effective way to evaluate the solution quality of a heuristic is to compare it with a strong lower bound by solving an associated problem relaxation. However, in this case, the easiest and most commonly used LP relaxation lower bound is extremely weak. For example, for the smallest test problem (1×5), the LP relaxation solution is 1593.375 while the optimal solution of this problem is 2320.0.

We can construct a better lower bound by relaxing the business volume constraint. The relaxed problem is created in this instance by allowing the discount to be applied to all quantities purchased. The lower bound can be obtained by solving m one-item lot size models separately, and then reducing the resulting cost by applying the discount. To evaluate the quality of this lower bound, we consider a solution where the discount is disabled, i.e., the solution to the m separate classic one-item lot-size models without involving a discount. Obviously, this solution represents a feasible solution (or upper bound solution) for our original problem (marked as UB), and it has the same order schedule and quantities (and therefore the same setup and inventory costs) as the lower bound solution. For these two solutions, let $C1$ be the sum of set-up and inventory costs, and $C2$ be the total values of all items purchased. Then the ratio of the cost of the lower bound solution to the cost of the feasible solution is $(C1 + (1 - \alpha)C2)/(C1 + C2)$ which is clearly greater than $1 - \alpha$. This shows that the ratio of the lower bound cost to the optimal cost is approximately between $1 - \alpha$ and 1.0, that is, between 0.9 and 1.0 for our test problems.

For practical purposes, we also compare our heuristic solution with a feasible solution derived by employing the commercial package CPLEX (CPLEX MIP 3.0) to solve the mixed integer programming model in Section 2. However, in the second test set, except for four very small problems (marked by *), most problems cannot be solved optimally by CPLEX even after running for an exceedingly long CPU time on a powerful DEC ALPHA machine with 200MHZ. For example, we ran CPLEX on a moderate multi-item problem 3×25 for over 200 hours (more than 8 days), but could not obtain a solution even reasonably close to the DPH result. We therefore (pragmatically) compare the best feasible solutions obtained by CPLEX, after allowing it to run for a significant period, with the DPH solutions in Table 2.

Problem	CPLEX		DPH	
$(m \times n)$	Solution	CPU (hour)	Solution	CPU (second)
(3×5)	7047.00*	0.03	7102.40	0
(3×10)	9205.00*	0.23	9261.00	0
(3×25)	26656.05	28.26	24513.55	0
(3×50)	66784.00	28.5	50582.09	1.2
(3×100)	174665.00	24.75	108238.21	18.2
(5×5)	8136.00*	0.15	8300.00	0
(5×10)	15945.42*	12.98	16080.00	0
(5×25)	40097.10	40.75	29135.09	0
(5×50)	88991.64	69.65	70190.79	2.4
(5×100)	280928.50	103.32	195329.73	33.2

Table 2. Solutions of CPLEX and DPH on the second problem set

From Table 2, we observe that our DPH approach can get solutions that are close to the optimal on the four small problems (the ones marked by asterisks), and solves these problems so rapidly that the time does not even register in a fraction of a second. On the remaining problems, DPH are significantly, often dramatically, superior to those obtained by CPLEX. Moreover, the DPH approach runs from 4,800 to over 100,000 times faster than CPLEX to obtain these superior solutions. This suggests that the branch and bound approach employed by CPLEX, although generally acknowledged to be among the best devised, is impractical for real world applications of the multi-item discount problem.

We further compare our DPH solution with associated LB and UB solutions in Table 3. In order to more accurately investigate the performance of the algorithms, we remove the constant term ($C2 = \sum_{i=1}^{m} \sum_{t=1}^{n} v_i x_{it}$) from the total costs, since this cost is not intended for optimization. Therefore, the revised costs now are only composed of setup and inventory costs, and negative discount terms. We list the revised costs of DPH and UB in Table 3 (marked as DPH$^+$ and UB$^+$) and compare their ratios.

Problem	Lower Bound (LB)		Upper Bound (UB)		Revised Cost		
$(m \times n)$	Solution	LB/DPH	Solution	DPH/UB	DPH^+	UB^+	DPH^+/UB^+
(3×5)	6563.00	92.41	7150.00	99.33	1232.40	1280.00	96.28
(3×10)	8741.00	94.39	9370.00	98.84	2971.00	3080.00	96.46
(3×25)	23261.00	94.89	24900.00	98.45	8123.55	8510.00	95.46
(3×50)	48205.00	95.30	51830.00	97.59	14332.09	15580.00	91.99
(3×100)	101262.00	93.55	109480.00	98.87	26058.21	27300.00	95.45
(5×5)	7725.00	93.07	8300.00	100.00	2550.00	2550.00	100.00
(5×10)	15146.00	94.19	16265.00	98.86	4890.00	5075.00	96.35
(5×25)	27766.00	95.30	29445.00	98.95	12345.09	12655.00	97.55
(5×50)	66773.00	95.13	71220.00	98.55	25720.79	26750.00	96.15
(5×100)	182290.00	93.32	196660.00	99.32	51629.73	52960.00	97.49

Table 3. Comparisons of DPH with LB and UB

In the comparisons with the lower bound in Table 3, we find that the average value of LB/DPH is 0.942. Considering that the average value of LB/CPLEX* (for the four optimal cases) is 0.945, and the average value of DPH/CPLEX* (for the four optimal cases) is 1.041, we surmise that the DPH solutions are close to the optimal solutions for all problems. Compared with the upper bound solutions which simply ignore the discounts, our DPH take advantage of the discounts and finds nine better solutions out of ten problems. The average improvement is 98.88% for the total costs and 96.32% for the revised costs. (These DPH^+ solutions can be further improved as we show subsequently.) These results demonstrate the potential of DPH as a good practical heuristic.

Fine Tuned and Accelerated Versions of the DPH Method.

We now investigate the impact of various lower bound L_{it} calculations and fine tuned procedures as well as a speedup method. Again, in order to more accurately investigate the performance, we use the revised cost subsequently, i.e., we remove the constant term $(\sum_{i=1}^{m} \sum_{t=1}^{n} v_i x_{it})$ from the subsequent costs, since this cost is incurred by all variants of DPH. We continue to refer to our initial solution simply as DPH, and identify the variants by their description labels. Since we find that most of these variants take the same magnitude of computational times as the original DPH, we will not list the CPU times in these comparisons except in the case of SM, for which speed is an important factor to consider.

The solutions obtained by other proportional lower bound calculations (previously identified as PC1, PC3, PC4) are presented in Table 4. For ease of comparison, we only report the ratio of the corresponding solution costs to those by the original DPH approach (which uses PC2).

Problem	PC1/DPH	PC3/DPH	PC4/DPH
(3×5)	1.000	0.958	1.000
(3×10)	1.000	1.000	1.000
(3×25)	1.000	1.006	1.000
(3×50)	1.004	1.019	1.005
(3×100)	1.000	0.973	1.000
(5×5)	1.000	0.953	0.975
(5×10)	1.000	1.000	1.000
(5×25)	1.011	0.997	1.000
(5×50)	1.000	0.990	0.995
(5×100)	1.000	0.988	1.000

Table 4. Comparisons from Proportional Calculations for Lower Bounds

From Table 4, we find that the PC3 variant appears to be the best. We conjecture that business volume may be an important factor in determining the lower bounds. On the other hand, the feasibility of the lower bound is not sensitive. In our experiment, even the simplest lower bound calculation, PC1, that has the potential to produce many infeasible lower bounds, still obtains solutions that are consistently very close to those of PC2 (which is the default setting for the original DPH). The more advanced procedures driven by lower bound feasibility as embedded in PC4, do not appear necessary in practice.

We next incorporate the two FT methods in DPH and compare the results with DPH. The outcomes are presented in Table 5.

Problem	FT1/DPH	FT2/DPH
(3×5)	0.981	0.954
(3×10)	1.000	1.000
(3×25)	0.994	0.994
(3×50)	0.979	0.979
(3×100)	0.977	0.976
(5×5)	1.000	1.000
(5×10)	1.000	1.000
(5×25)	0.980	0.980
(5×50)	0.988	0.992
(5×100)	0.982	0.984

Table 5. Solutions of Fine Tuned methods for Lower Bounds

We see from Table 5 that both of these two fine tuned procedures can effectively improve the solution. Although FT2 is a more comprehensive heuristic than FT1, it does not always beat FT1. This is because FT1 and FT2, are similar in nature, and can be trapped in a local optimum. FT2 is a generally "stronger" local search method than FT1 for the relaxed problem where the discount and non-discount status is fixed via lower bound assignment. However, since we fix the discount or non-discount status at each iteration in DPH, the stronger FT2 heuristic may not always yield a better final solution than FT1. This finding suggests that incorporating a more intelligent search methodology such as Tabu Search into our DPH heuristic may be a basis for generating still better solutions.

The results of the speedup method SM are provided in Table 6. In addition to comparing solutions, we also investigate their computation times.

Problem	Cost	CPU time
$(m \times n)$	SM/DPH	SM / DPH
(3×5)	1.000	1.000
(3×10)	1.000	1.000
(3×25)	1.000	0.677
(3×50)	1.005	0.556
(3×100)	1.000	0.476
(5×5)	1.000	1.000
(5×10)	1.000	1.000
(5×25)	1.000	1.000
(5×50)	0.999	0.511
(5×100)	1.000	0.500

Table 6. Performance of the SM

Our findings disclose that SM can speed up DPH significantly without deteriorating the solution quality. (SM generates the worse solution than DPH in only one case, and then still obtains outcome exceedingly close to that of DPH). Because of its speed, SM is appealing to be employed in large planning and scheduling systems, where the multi-item lot size problem with discount is required to be solved frequently.

It is interesting to examine the effects of combining the various enhanced components. We designate HYBRID to be the dynamic programming based heuristic that incorporates PC3, FT1 and SM, and compare this variant with the original DPH approach. The results are presented in Table 7.

Problem	Cost			CPU Time
$(m \times n)$	HYBRID	DPH	$HYBRID/DPH$	HYBRID/DPH
(3×5)	1181.00	1232.4	0.958	1.000
(3×10)	2924.00	2971.10	0.984	1.000
(3×25)	7993.93	8123.55	0.984	0.677
(3×50)	14303.40	14332.09	0.998	0.556
(3×100)	24781.00	26058.21	0.951	0.444
(5×5)	2431.00	2550.00	0.953	1.00
(5×10)	4890.00	4890.00	1.000	1.000
(5×25)	12314.00	12345.09	0.997	0.750
(5×50)	25182.86	25720.79	0.979	0.511
(5×100)	50703.58	51629.73	0.982	0.500

Table 7. Performance of the HYBRID

From Table 7, we observe that for most test problems, HYBRID yields better solutions than applying PC3 and FT1 separately. The results are very consistent with those in Table 4–6. For the four small problems where the optimal solutions are known, HYBRID improves three cases by obtaining solutions very close to the optimal costs.

7. Conclusion

This paper provides the first study of the deterministic dynamic multi-item lot size problem with a joint business volume discount. We formulate the problem as a mixed integer programming model, but observe that this model cannot be solved within a reasonable time period by current optimization procedures. To solve the problem heuristically, we introduce a set of lower bounds to approximately replace the complicating side constraints, then solve the revised model using a dynamic programming based approach. We show how this approach can also be extended to solve the more complicated problem of multiple discount levels. Various lower bound calculations and solution post-optimization procedures as well as a speedup method are proposed and their performances are investigated. The experiments demonstrate the effectiveness of the heuristic and its promise as a practical tool.

Although this research focuses on the deterministic dynamic lot size model without backlogging, it is not difficult to apply our methodology in broader contexts, which incorporate backlogging, non-zero trial inventories, etc. More numerical experiments are required to validate the robustness of our heuristic in various environments.

An interesting avenue for investigation is to embed our approach in a meta-heuristic procedure such as Tabu Search to reduce the remaining optimality gap in cases where it still exists.

Acknowledgements

We thank anonymous referees and the associate editor for their helpful comments on earlier versions of this paper.

Reference

Bregman, R.L., "An Experimental Comparison of MRP Purchase Discount Methods," *Journal of Operational Research Society,* 42 (1991), 235-245.

Bregman, R.L. and E.A. Silver, "A Modification of the Silver-Meal Heuristic to Handle MRP Purchase Discount Situations," *Journal of Operational Research Society,* 44 (1993), 717-723.

Buffa, E.S. and J.G. Miller, *Production-Inventory Systems: Planning and Control,* Irwin, Homewood, Illinois, 1979.

Federgruen, A. and C. Lee, "The Dynamic Lot Size Model with Quantity Discount," *Naval Research Logistics,* 37 (1990), 707-713.

Hadley, G. and T.M. Whitin, *Analysis of Inventory Systems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

Johnson, L.A. and D.C. Montgomery, *Operations Research in Production Planning, Scheduling, and Inventory Control,* John Wiley & Sons, New York, 1974.

Sadrian, A. and Y. Yoon, "Business Volume Discount: A New Perspective on Discount Pricing Strategy," *Journal of Purchasing and Materials Management,* 28 (1992), 43-46.

Sadrian, A. and Y. Yoon, "A Procurement Decision Support System in Business Volume Discount Environments," *Operations Research,* 42 (1994), 14-23.

Silver, E.A. and R. Peterson, *Decision Systems for Inventory Management and Production Planning,* 2nd ed., John Wiley & Sons, New York, 1985.

Wagner, H.M. and T.M. Whitin, "A Dynamic Version of the Economic Lot Size Model," *Management Science,* 5 (1958), 89-96.