Journal of Business Theory and Practice
ISSN 2372-9759 (Print) ISSN 2329-2644 (Online)
Vol. 4, No. 2, 2016
www.scholink.org/ojs/index.php/jbtp

A New Recursive Dynamic Lot-Sizing Model with Multi-Level Discount

Fu C. Chyr[1*], Shu Y. Lin[1], Shan Y. Huang[2] & Jui C. Lu[3]

[1] Department of International Business, Chang Jung Christian University, Tainan, Taiwan

[2] Department of International Trade, Overseas Chinese University, Taiwan

[3] Department of Industrial Engineering and Management, National Kaohsiung University of Applied Sciences, Taiwan

[*] Fu-Chiao Chyr, E-mail: chyr@mail.cjcu.edu.tw

Abstract

The optimal solution of dynamic lot-sizing problem with multi-level discount is solved by integer programming model in the past papers. However, a complex computation and a large computer memory are generated. Due to the complexity and the big computer memory, the heuristic approaches and the variable neighborhood algorithm are usually adopted in the large-scale multi-level lot-sizing problem. This paper develops a model with a recursive relation between the adjacent periods, and to obtain an optimal solution when multi-level discount is considered. Four types of the feasible policies in the Dynamic Lot-Sizing model with multi-level discount are classified to develop the recursive relations. A few properties, theorems and algorithms are developed to show the recursion between the adjacent periods. The number of addition items will significantly be reduced. The recursive algorithm can significantly decrease the computational entries, comparison entries and computer memory and improve the computation efficiency.

Keywords

quantity discount, Dynamic Lot-Sizing, inventory management, supply chain management

1. Introduction

Wagner and Whitin presented a Dynamic Lot-Sizing (DLS) model to obtain the minimum total cost in inventory management. The disadvantage of Wagner-Whitin (WW) model is it suffers from a computational complexity in supply chain management. Some efficient algorithms to improve the computational efficiency of the WW model are developed (Federgruen & Tzur, 1991; Chyr, 1993). However, they do not discuss the treatment of quantity discount, which is encountered in real practice in purchasing or production.

Many researchers have studied the issue of multi-level discount by heuristic approach in purchasing or production management. A tabular approach with quantity discount has been presented. However, the solution obtained by the tabular approach is not always optimal. The heuristic approach with Least Unit Cost (LUC) has been discussed. Christoph (1989) presented lot-sizing technique in multiple discounts. Lee, Zhu and Ruangdet (1993) compared the performance of LUC, IPPA (Incremental Part Period Algorithm) and WW model with multi-level discount and rolling horizon.

A few researches study the optimal solutions of DLS problem with multi-level discount. However, those models yield a complex computation based on linear programming and need a large computer memory in supply chain system. A new recursion model to overcome those problems is developed in this paper. Callarman and Whybark presented a mixed integer-programming model with multi-level discount. Chung et al. (1987) adopted a tree search procedure to establish a DLS model and to obtain

an optimal order policy of it with multi-level quantity discount. Chyr (1990, 1999) presented a recursive model to improve the computation in the DLS model with single discount quantity in a purchasing system. By extending the model presented by Chyr (1999), the DLS model with multi-level discount is developed to obtain a good recursive relation in DLS model. Munson and Rosenblatt (2001) discussed lot size model based on quantity discounts from viewpoint of supply chain. Hu and Munson (2002) developed DLS rule for incremental quantity discounts by linear programming approach. Rubin and Benton (2003) published a paper entitled "evaluating jointly constrained order quantity complexities for incremental discounts" to show the complexity of quantity discounts. Due to the complexity of computation and big computer memory in supply chain management, Chang et al. (2010) used database with minimum computer memory to discuss the characteristics of large-scale multi-level lot-sizing problem. Yiyong et al. (2011) presented "a reduced variable algorithm" to solve multi-level lot-sizing problem. Findt et al. (2012) developed the heuristic approach of neighborhood search for lot-sizing problem. This paper established a new recursive relation of the single level lot-sizing model with multi-level discount to obtain optimal solution. This approach decreases computational entries and computer memory required in inventory management. The following assumptions with purchasing discount are adopted in this paper.

1) The single and independent item is involved.
2) The inventory is zero at the end period t in a planning horizon of *t* periods.
3) The cost of purchased items is a function of the number of units purchased.
4) The inventory costs vary from period to period.
5) Each order is received in a single delivery.
6) Stock-outs are not permitted in inventory management.

2. Problems of the Wagner-Whitin model with Multi-Level Discount

The properties and the recursive relations of the conventional WW model cannot be applied to solve the case of multi-level discount in Wagner and Whitin research. Considering a planning horizon of T periods, in the undiscounted lot-sizing model, both demands and costs can change from period to period. Given T periods for the undiscounted lot-sizing problem, let Q_i be the quantity purchased in period i, D_i be the demand during period i, P be the unit purchasing cost and I_i be the inventory at the end of period i. Specifying $I_0 = I_T = 0$, we obtain

$$I_{i-1} + Q_i - I_i = D_i, i = 1,2,..., n, Q_i \geq 0, I_i \geq 0 .$$

Let h_i be the cost of carrying one unit in stock from period i-1 to the beginning of period i. Observe S_i is the ordering cost in period i and $P(Q_i)$ is the cost of purchasing Q_i units. Since the ordering cost is concave, so that the total costs of the dynamic lot-sizing model becomes

$$\text{Min} \left\{ \sum_{i=1}^{T} \left(\delta_i S_i + P(Q_i) + h_i I_i \right) \right\} \tag{1}$$

Where, $I_{i-1} + Q_i - I_i = D_i$, $P(Q_i) = P \times Q_i$

$$\delta_i = \begin{cases} 0, & \text{if } Q_i = 0 \\ 1, & \text{if } Q_i > 0 \end{cases}, S_i \geq 0, P > 0, Q_i \geq 0, I_i \geq 0, I_0 = I_T = 0, i = 1, 2, ..., T.$$

As has been shown in the WW model, there is the following property that

$$I_{i-1} Q_i = 0 \tag{2}$$

The equation (2) means that there exists an optimal program such that $I_{i-1}Q_i = 0$ for all i periods. If $IQ_i > 0$ holds, then it does not reschedule the purchase of I by including the quantity in Q_i. Can this property be used in the DLS model with multi-level discount? The result is it does not exist in the DLS model with multi-level discount. This is the first problem of solving the above model in inventory management.

Let $G(t)$ be the minimum total cost for periods 1 through t in all feasible policies at the end of period t. Wagner and Whitin have developed a standard forward dynamic programming formulation without quantity discount as follows:

$$G(t) = \text{Min}\left\{\begin{array}{l} S_t + G(t-1) \\ \text{Min}_{1\le j\le t-1}\left\{G(j-1) + S_j + \sum_{m=j+1}^{t}\sum_{k=m+1}^{t}(h_m D_k)\right\}\end{array}\right\} \quad (3)$$

where $G(1) = S_1$, $G(0) = 0$, $t = 1, 2, ..., T$.

Evans, Federgruen and Chyr have presented a few efficient algorithms to improve equation (3). However, they still do not discuss the case of multi-level discount.

Let QB_k, $k = 1, 2, ..., m$ denote multi-level discounts. The sequence of discount level orders is $QB_1 < QB_2 < QB_k < ... < QB_m$. Considering the multi-level discount in the dynamic lot-sizing model, the unit purchasing cost at period i is $P_{i,k}$ that the lot size of the order is in the kth discount range of $Q_i \ge QB_k$. The problem is to find an optimal order policy that minimizes the sum of ordering cost, inventory holding cost, and purchasing cost over the planning horizon. The DLS model of the multi- level discount can be formulated as follows:

$$\text{Min}\left\{\sum_{i=1}^{T}\left(\delta_i S_i + P(Q_i) + h_i I_i\right)\right\} \quad (4)$$

where $I_{i-1} + Q_i - I_i = D_i$, $i = 1,2,...,T.$, $P(Q_i) = P_{i,k} \times Q_i$, $k = 1,2, ..., m$.

$$\delta_i = \left\{\begin{array}{l} 0, \quad \text{if } Q_i = 0 \\ 1, \quad \text{if } Q_i > 0 \end{array}\right\}, S_i \ge 0, Q_i > 0, I_i 0, i = 1, 2, \ldots, T.$$

Equation (4) is different from equation (1). In the equation (4), the purchasing cost, $P(Q_i)$, is a function of Q_i and $P_{i,k}$. To solve equation (1), WW algorithm has been presented the equation (3). However, the equation (3) cannot be applied to solve equation (4). Considering multi-level discount, the algorithm of equation (3) should be revised to obtain the minimum total cost.

There are the following problems in solving the equation (4) of a dynamic lot-sizing model with multi-level discount:

1) The property of $I_{i-1}Q_i = 0$ cannot completely be used in solving the equation (4).

2) Can the recursive relation in the equation (3) be applied in solving the equation (4)?

Wagner and Whitin present that the property of $I_{i-1}Q_i = 0$ does not hold if the buying or production costs are not constant and identical for all periods. For the discounted problem, the recursive relation presented in the equation (3) has not been used in the literature.

Due to multi-level discount, it will result in many feasible solutions, which cannot satisfy the property of $I_{i-1}Q_i = 0$. How to find optimum set of solutions and to reduce the feasible solutions is a main consideration on developing multi-level discount model.

Let (Q_1, Q_2, Qi, Q_T) be a feasible policy that Q_i is a nonnegative integer and Q_1 is a positive integer. An example with four periods and two discount levels, given in Table 1, will illustrate the complexity of the feasible policy.

Table 1. Value of Parameters Given in the Example for 4 Periods

Demand, D_i	$D_1 = 50$, $D_2 = 40$, $D_3 = 120$, $D_4 = 130$
Discount level, QB_k	$QB_1 = 100$, $QB_2 = 200$
Unit purchasing cost	$P = 10$, when $Q_i < 100$,
	$P_1 = 9$, when $100 \le Q_i < 200$,
	$P_2 = 8$, when $Q_i \ge 200$
Ordering cost, S_i	$S_i = 10$, $i = 1, 2, 3, 4$
Holding cost per unit for each period, h_i	$h_i = 1$, $i = 1, 2, 3, 4$

Suppose that there are 3 periods and that the demands are $D_1 = 50$, $D_2 = 40$ and $D_3 = 120$. Then (50, 40, 120) is a feasible policy of $Q_1 = 50$, $Q_2 = 40$ and $Q_3 = 120$ for a planning horizon of 3 periods. This policy satisfies the property of $I_{i-1}Q_i = 0$ presented in equation (2) without considering the quantity discount. Defining $QB_1 = 100$ as the first discount level. To obtain the first discount benefits, ordering the quantities, 100, at period 1 and period 2 is adopted. The feasible policy, (100, 100, 10), is rearranged the order quantity in each period to obtain the first discount benefits in period 1 and period 2. The policy (200, 5, 5) is also a feasible solution to obtain a minimal cost. Besides, the policies such as (50, 100, 60), (100, 30, 80) are the feasible solutions. The policies, mentioned as (100, 100, 10), (200, 5, 5), (50, 100, 60) and (100, 30, 80), are the feasible solutions in the multi-level discount model, but they cannot satisfy the property of $I_{i-1}Q_i = 0$ and the equation (3). To obtain the benefits of multi-level discount, many possible policies such as (100, 100, 10) and (50, 100, 60) are generated at period 3. These possible policies generated at period 3 yield the complexity of solving the minimum total cost.

3. Decreasing Feasible Policies

There are many possible policies, which are generated by the factor of multi-level discount. Let QB_k be the *kth* discount level. The following theorems can reduce the size of feasible policy set.

Theorem 1:

Let Q_i be the quantity purchased at period i for $i > 1$. If a purchasing policy with $Q_i \neq 0$, $Q_i \neq D_i$, $Q_i \neq QB_k$ and $Q_i \neq (\sum_{a=1}^{r} D_a - \sum_{a=1}^{n-1} Q_a)$ for $r = n, n+1, T$. holds, then the policy with Q_i does not need to be considered for optimal solution.

Proof: The proof is shown in the appendix.

Theorem 1 shows that the quantity purchased in period i ($i > 1$), Q_i, will be one of the $Q_i = 0$, $Q_i = D_i$, $Q_i = QB_k$, or $Q_i = (\sum_{a=1}^{r} D_a - \sum_{a=1}^{n-1} Q_a)$ for $r = n, n+1, \ldots, T$. Theorem 1 will reduce a lot of possible policies such as (200, 5, 5) and (100, 30, 80) mentioned in section 2. It is important to find the optimal solution with less feasible solutions. Using theorem 1, the total cost of the policy (200, 5, 5) is higher than that of the policy (200, 0, 10). The policy (200, 5, 5) does not need to be considered for optimal solution, since $Q_2 = 5$ is not equal to 0, D_i, QB_k, or $(\sum_{a=1}^{3} D_a - \sum_{a=1}^{1} Q_a)$. The total cost of

the policy (100, 30, 80) is higher than that of the policy (100, 0, 110). Theorem 1 eliminates many feasible solutions for solving the lot size with multi-level discount. It means that the quantity purchased, Q_i, at period i ($i > 1$) will be one of the $Q_i = 0$, $Q_i = D_i$, $Q_i = QB_k$, or $Q_i = (\sum_{i=1}^{r} D_i - \sum_{i=1}^{n-1} Q_i)$ for $r = n, n + 1, \ldots, T$.

4. Types of All the Feasible Policies

An ordering policy at period t means that the amount ordered at period t is larger then zero. An inventory policy at period t means that the amount ordered at period t is equal to zero. Adopting the inventory policy, the demands D_i are purchased at period j where $j < i$. For a discounted program, the properties of all the feasible policies presented by theorem 1 can be classified into four types of feasible policies. The first type of policies is that we order D_t at period t, which is an extending policy from period t-1. The second type of policies is rearranging order quantities in the first type of policies at period t due to satisfying the multi-level discount. The third type of policies is that we hold an inventory policy at period t, which is an extending policy from period t-1. The fourth type of policies is rearranging order quantities in the third type of policies at period t due to satisfying the multi-level discount. Using the example mentioned in section 2, we list all the feasible policies as seen in Table 2 and indicate the type of policies.

Table 2. Types of the All the Feasible Policies before Improvement

(1) (50, 40, 120, 130)*
(2) (100, 40, 120, 80)
(2) (50, 100, 120, 70)
(2) (50, 40, 200, 50)
(1) (100, 40, 70, 130)*
(2) (100, 100, 70, 70)
(2) (100, 40, 100, 100)
(2) (200, 40, 70, 30)
(1) (100, 100, 10, 130)*
(2) (200, 100, 10, 30)
(2) (100, 200, 10, 30)
(2) (100, 100, 100, 40)
(1) (50, 100, 60, 130)*
(2) (100, 100, 60, 80)
(2) (50, 200, 60, 30)
(2) (50, 100, 100, 90)
(1) (90, 0, 120, 130)*
(2) (100, 0, 120, 120)
(2) (200, 0, 120, 20)
(2) (90, 0, 200, 50)
(1) (100, 0, 110, 130)
(2) (200, 0, 110, 30)
(2) (100, 0, 200, 40)
(1) (200, 0, 10, 130)*

			(2) (200, 0, 100, 40)
			(1) (50, 160, 0, 130)*
			(2) (50, 200, 0, 90)
			(1) (100, 110, 0, 130)*
			(2) (100, 200, 0, 40)
			(1) (200, 10, 0, 130)*
			(2) (200, 100, 0, 40)
			(1) (210, 0, 0, 130)*
		(1) (50, 40, 120)	(3) (50, 40, 250, 0)*
		(2) (100, 40, 70)	(4) (50, 200, 90, 0)
		(2) (100, 100, 10)	(4) (200, 40, 100, 0)
		(2) (50, 100, 60)	(4) (200, 100, 40, 0)
			(3) (100, 40, 200, 0)*
			(4) (100, 200, 40, 0)
			(3) (100, 100, 140, 0)*
			(3) (50, 100, 190, 0)*
		(1) (90, 0, 120)*	(3) (90, 0, 250, 0)*
		(2) (100, 0, 110)	(3) (100, 0, 240, 0)
		(2) (200, 0, 10)	(3) (200, 0, 140, 0)*
	(1) (50, 40)	(3) (50, 160, 0)	(3) (50, 290, 0, 0)*
		(4) (100, 110, 0)	(3) (100, 240, 0, 0)*
		(4) (200, 10, 0)	(3) (200, 140, 0, 0)
(1) (50)	(3) (90, 0)	(3) (210, 0, 0)	(3) (340, 0, 0, 0)

Note. (1), (2), (3), and (4) indicate the type of feasible policies.

The symbol of * indicates that these entries can be eliminated after using recursion in section 5.

Table 2 illustrated the properties of the feasible policy. The policy (50, 40, 120, 130) presented in Table 2 is a feasible policy for a planning horizon of 4 periods. This feasible policy has a preceding feasible policy expressed by (50, 40, 120) at period 3. To obtain the discount benefits, we can order the quantities, 100, at period 1 or period 2. Furthermore, the quantity, 200, purchased at period 3 is also a feasible alternative. The policies (100, 40, 120, 80), (50, 100, 120, 70), and (50, 40, 200, 50) are generated at period 4 to obtain the discount benefits. The preceding policy of (100, 40, 120, 80) does not exist at period 3. Therefore, the former policy (50, 40, 120, 130) at period 4 has a good relation with the preceding policy (50, 40, 120) at period 3. But the latter policies (100, 40, 120, 80), (50, 100, 120, 70), and (50, 40, 200, 50) generated at period 4 will be treated as a new case. The latter policies will yield an added complexity of the discounted case over the undiscounted case.

Furthermore, both the policies with a preceding one and the policies generated at period t will be further divided into two types:

1) The policy at period t is a reordering policy, such as (50, 40, 120, 130) and (100, 40, 120, 80).

2) The policy at period t is an inventory policy, such as (50, 40, 250, 0) and (50, 200, 90, 0).

Generally speaking, the feasible policy can be classified into four cases mentioned above. These cases act as the base of developing a multi-level discount model with recursion.

5. Finding the Minimum Total Cost with New Recursion

5.1 List of Notations

In this section, a recursive procedure will be developed to construct a dynamic programming model and to reduce computation. The following notations are adopted.

n: the number of ordering periods;

D_t: demand at period t, (t = 1, 2, …, T), $D_t \geq 0$;

S_t: ordering cost at period t, (t = 1, 2, …, T), $S_t > 0$;

h_t: unit holding cost from period t-1 to period t for the inventory at the end of period t-1, $h_t > 0$;

$K_{t,new}$: the number of feasible solutions generated at period t due to discount factor;

P: the original unit cost purchased;

P_t: the unit purchasing cost at period t;

QB_k: the kth quantity discount level;

P_{t,QB_i}: the discounted unit purchasing cost at period t;

Q_t: ordering quantity at period t, (t = 1, 2, …, T), $Q_t \geq 0$;

r_i: the ith discount rate, where $P_{t,QB_i} = P \times r_i$;

$G(t)$: the minimum total cost for periods 1 through t in all feasible policies at the end of period t, $G(0) = 0$;

$T(t, k)$: the total cost of the kth feasible policy at period t;

$g(t, j)$: the minimum total cost of the inventory policy at period t, which has the final order at period j, $j < t$, and has a preceding policy, $g(t\text{-}1, j)$, in period t-1;

$g_n(t, j, k)$: the total cost of the new kth inventory policy generated at period t, which has not a preceding policy at period t-1 and has a final ordering period j;

$g(t, t)$: the minimum total cost of the reordering policy at period t, which has a preceding policy at period t-1;

$g_n(t, t, k)$: the total cost of the new kth reordering policy generated at period t, which has not a preceding policy at period t-1;

$G(t, t)$: the minimum total cost for periods 1 through t in all reordering policy at period t;

$G(t, j)$: the minimum total cost for periods 1 through t in all inventory policy at period t, which has the final order at period j, $j < t$.

5.2 Calculating Total Cost of each Feasible Policy

Using the notations mentioned above, the total cost of each feasible policy could be formulated as follow:

$$T(t,k) = n \times S_t + \sum_{i=1}^{t} (Q_i \times P_{t,QB_i}) + \sum_{j=2}^{t} ((\sum_{k=1}^{j} Q_k - \sum_{k=1}^{j} D_k) \times h_j) \quad (5)$$

Equation (5) is suitable for all of the feasible policies and can be used to calculate the total cost of the new kth policy, $g_n(t, t, k)$ and $g_n(t, j, k)$, generated at period t. The total cost of $g(t, j)$ and $g(t, t)$ can also be computed by equation (5). Since $g(t, j)$ and $g(t, t)$ will have a good recursive relation between period t-1 and period t in the latter discussion, the above equation will not be applied to $g(t, j)$ and $g(t, t)$.

5.3 Recursive Relations between Period t-1 and Period t for g(t, j) and g(t, t)

We can develop a good recursive relation to simplify the computation of $g(t, j)$ and $g(t, t)$. The first, $g(t, t)$ can be formulated as follow:

$$g(t,t) = G(t\text{-}1)+S_t+D \times P_{t,QB_i} \tag{6}$$

where $t = 1, 2, T$, $P_{t,QB_i} = \left\{ \begin{array}{l} P_t, \text{ if } D_t < QB_1 \\ P_{t,QB_i}, \text{ if } QB_{i-1} \le D_t < QB_i \end{array} \right\}$, $G(0) = 0$.

$g(t, t)$ denotes the minimum total cost of a reordering policy at period t that has a preceding policy at period t-1. The added total cost at period t will be expressed as $S_t + D_t \times P_{t,QB_i}$. Since $G(t\text{-}1)$ is the minimum total cost for periods 1 through t-1 in all feasible policies at the end of period t-1, only one $g(t, t)$ at period t will be computed by the equation (6). This recursive relation presented in the equation (6) eliminates many feasible policies, only one entry of the reordering policy exists at period t.

The total cost of an inventory policy at period t, $g(t, j)$, which has a preceding policy in period t-1, can be computed as follow:

$$g(t, j) = G(t\text{-}1, j) + D_t \times \sum_{v=j+1}^{t} h_v + D_t \times P_{t,QB_i} - (\sum_{v=j}^{t-1} D_v) \times (1 - r_i) \times P \tag{7}$$

where $0< r_i < 1$, if $\sum_{v=j}^{t-1} D_v < QB_i$ and $\sum_{v=j}^{t} D_v \ge QB_{i-1}$

$r = 1$, if $\sum_{v=j}^{t} D_v < QB_{i-1}$

The above equation presents a recursive relation between $g(t, j)$ and $G(t\text{-}1, j)$. Only the added total cost $D_t \times \sum_{v=j+1}^{t} h_v$, $D_t \times P_{t,QB_i}$ and $-(\sum_{v=j}^{t-1} D_v)(1 - r_i)P$ should be computed in period t. This recursive relation eliminates many computations of the feasible inventory policy at period t. It needs only one computation between $G(t\text{-}1, j)$ and $g(t, j)$. If $\sum_{v=j}^{t} D_v < QB_{i-1}$ holds, then $(\sum_{v=j}^{t-1} D_v)(1 - r_i)P$ is equal to zero. These are computational improvements.

5.4 An Algorithm of Computing the Minimum Total Cost

In fact, all the feasible policies consist of $g(t, t)$, $g_n(t, t, k)$, $g(t, j)$ and $g_n(t, j, k)$. Using the equations (5), (6) and (7), a forward dynamic programming algorithm can be outlined as follow:

(1) Initialization: $G(0) = 0$, $G(1, 1) = G(1)$.

(2) Recursion: This step is divided into four cases:

Case1 The period t adopts a reorder policy, which has a preceding policy at period t-1. The minimum total cost of these policies at period t is computed as $g(t, t)$ presented by equation (6).

Case2 The period t adopts an inventory policy, which has a preceding policy at period t-1. The total cost of each policy at period t is computed as $g(t, j)$ presented by equation (7).

Case3 Considering multi-level discount, the new inventory policies, which have not a preceding policy, are rearranging order quantities in each period. The total cost of the new *kth* inventory policy at period t is computed as $g_n(t, j, k)$ presented by equation (5).

Case4 Considering multi-level discount, the new reordering policies, which have not a preceding policy, are rearranging order quantities in each period. The total cost of the new *kth* reordering policy at period t is computed as $g_n(t, t, k)$ presented by equation (5).

(3) To determine $G(t, t)$

$$G(t, t) = \text{Min} \left\{ \begin{array}{l} g(t, t) \\ g_n(t, t, k) \end{array} \right\} \tag{8}$$

where k is positive integer.

(4) To determine $G(t, j)$

$$G(t, j) = \text{Min} \begin{Bmatrix} g(t, j) \\ g_n(t, j, k) \end{Bmatrix}$$

where $j < t$ and k > 0 (9)

(5) To determine the minimum total cost of $G(t)$ at period t

$$G(t) = \text{Min} \begin{Bmatrix} G(t, t) \\ G(t, j) \end{Bmatrix} \quad (10)$$

where j = 1, 2, …, t-1.

(6) Going to step (2) to proceed next period until the end of planning horizons.

Actually, the minimum total cost for periods 1 through t in all feasible policies at the end of period t, $G(t)$, can be formulated as

$$G(t) = \text{Min}\{G(t,t), G(t,j)\} = \text{Min} \begin{Bmatrix} \min \{g(t,t), g_n(t,t,k)\} \\ \min \{g(\text{t}, \text{j}), g_n(t, j, k)\} \end{Bmatrix} \quad (11)$$

where $G(0) = 0$, $j = 1, 2, \ldots, t\text{-}1$., $t = 1, 2, \ldots, T$.

The above equations mentioned in (6) ~ (10) yield a good recursive relationship. Using the equation (5) and (7), we obtain the equation (9), which has a good recursion between period t-1 and period t. Using the equation (5) and (6), the recursive equation (8) can be developed. The policy $g(t, t)$ means that the added total cost at period t depends on D_t, P_{t,QB_1} and the ordering cost S_t. The policy $g(t, t)$ at period t will be computed by the extension of the policy ,$G(t\text{-}1)$, at period t-1. The policy, $g(t, j)$, at period t also addresses that it has a good relationship with the preceding policy $g(t\text{-}1, j)$ at period t-1. As for $g_n(t, t, k)$ and $g_n(t, j, k)$, a general equation (5) is offered. Finally, the minimum total cost can be computed by a single expression presented by equation (11). These recursions result in computational improvements.

Let $K_{t,new}$ denote the number of feasible solutions generated at period t due to multi-level discount. Using equation (6) through (11), many computation entries, addition items and comparison items are reduced. The reduction number of feasible solutions resulted by equation (6) through (11) is listed in Table 3.

Table 3. The Numbers of Feasible Solution Reduction at Period *t* after Improvement

	Original method at period t	Using equation (6) through (11) at period t	Reduction
The number of feasible solutions	$2^{t\text{-}1} + K_{t\text{-}1,new} + K_{t,new}$	$t + K_{t,new}$	$2^{t\text{-}1} - t + K_{t\text{-}1,new}$

Table 3 shows the computational efficiency of adopting the new recursive method developed in this paper.

5.5Example

The recursive algorithm is demonstrated by using a problem with four periods and two discounts level, given in Table 1. This example illustrates the complexity of the discounted case over the undiscounted case, and shows the efficiency of the recursive relation mentioned in equations (6)-(11). The results of the recursive relationship are shown in Table 4.

Table 4. Reducing Feasible Policies Shown in Table 2 based on Recursive Relations between the Adjacent Periods

	(2) (100, 40, 120, 80) g_n(4, 4, 1) 3370 (2) (50, 100, 120, 70) g_n(4, 4, 2) 3340 (2) (50, 40, 200, 50) g_n(4, 4, 3) 3120
	(2) (100, 100, 70, 70) g_n(4, 4, 4) 3460 (2) (100, 40, 100, 100) g_n(4, 4, 5) 3270 (2) (200, 40, 70, 30) g_n(4, 4, 6) 3440
	(2) (200, 100, 10, 30) g_n(4, 4, 7) 3400 (2) (100, 200, 10, 30) g_n(4, 4, 8) 3300 (2) (100, 100, 100, 40) g_n(4, 4, 9) 3390
	(2) (100, 100, 60, 80) g_n(4, 4, 10) 3550 (2) (50, 200, 60, 30) g_n(4, 4, 11) 3300 (2) (50, 100, 100, 90) g_n(4, 4, 12) 3340
	(2) (100, 0, 120, 120) g_n(4, 4, 13) 3210 (2) (200, 0, 120, 20) g_n(4, 4, 14) 3330 (2) (90, 0, 200, 50) g_n(4, 4, 15) 3160
	(1) (100, 0, 110, 130) g_n(4, 4) 3150 (2) (200, 0, 110, 30) g_n(4, 4, 16) 3380 (2) (100, 0, 200, 40) g_n(4, 4, 17) 3080

				(2) (200, 0, 100, 40) g_n(4, 4, 18) 3280 (2) (50, 200, 0, 90) g_n(4, 4, 19) 3230 (2) (100, 200, 0, 40) g_n(4, 4, 20) 3280 (2) (200, 100, 0, 40) g_n(4, 4, 21) 3380
			(1) (50, 40, 120) g_n(3, 3) 2010 (2) (100, 40, 70) g_n(3, 3, 1) 2130 (2) (100, 100, 10)g_n(3, 3, 2) 2090 (2) (50, 100, 60) g_n(3, 3, 3) 2090 (2) (100, 0, 110) g_n(3, 3, 4) 1970 (2) (200, 0, 10) g_n(3, 3, 5) 1980	(4) (50, 200, 90, 0) g_n(4, 3, 1) 3320 (4) (200, 40, 100, 0) g_n(4, 3, 2) 3360 (4) (200, 100, 40, 0) g_n(4, 3, 3) 3420 (4) (100, 200, 40, 0) g_n(4, 3, 4) 3320 (3) (100, 0, 240, 0) g_n(4, 3) 3030 ▲
		(1) (50, 40) g_n(2, 2) 920	(3) (50, 160, 0) g_n(3, 2) 2080 (4) (100, 110, 0) g_n(3, 2, 1) 2080 (4) (200, 10, 0) g_n(3, 2, 2) 1990	(3) (200, 140, 0, 0) g_n(4, 2) 3410
	(1) (50) g_n(1, 1) 510	(3) (90, 0) g_n(2, 1) 950	(3) (210, 0, 0) g_n(3, 1) 1970	(3) (340, 0, 0, 0) g_n(4, 1) 3400
Minimum total cost	510	920	1970	3030
Optimal policy	(50)	(50, 40)	(100, 0, 110) or (210, 0, 0)	(100, 0, 240, 0)
Period	1	2	3	4

Note. (1), (2), (3), and (4) indicate the type of the feasible policy.

Table 4 lists all feasible solutions in the example after improvement. All feasible solutions are classified into four cases, such as $g(t, t)$, $g_n(t, t, k)$, $g(t, j)$ and $g_n(t, j, k)$. A recursive relationship between two adjacent periods is obviously shown in Table 2 and Table 4. Comparing Table 2 and Table 4, a few original policies of "*" existed in Table 2 are eliminated in Table 4. There are 29 feasible policies in Table 4 and 47 feasible policies in Table 2. The reduction rate of feasible solution in Table 4 is 23.8% in a 4-period planning horizon. Then as the planning horizons increase, the number of feasible policies reduced will significantly increase. Table 3 shows the improvement of feasible solution reduction.

As seen in Table 4, a few feasible solutions are eliminated to improve the computational complexity by recursion. The label (Q_1, $\delta_2 Q_2$, $\delta_I Q_i$) indicates an order policy from periods 1 through i. $\delta_i = 1$ denotes that the period i adopts a reorder policy. In other words, $\delta_i = 0$ shows that the period i adopts an inventory policy. The figure under each (Q_1, $\delta_2 Q_2$, $\delta_i Q_i$) gives the corresponding total cost and the recursive computation process expressed by equation (6) and (7). Table 4 shows the optimal order policy and the corresponding cost for each period at the bottom of each period.

The optimal policy of this example, listed at the bottom of Table 4, is to order 100 units in period 1 to cover the demands from period 1 through 2, 240 units in period 3 to cover the demands from period 3 through 4. Comparing Table 2 and Table 4, we can understand that the numbers of computation at period 4 in Table 4 are obviously less than that in Table 2. All the entries indicated as "*" in Table 2 can be eliminated in computing. It is a computational improvement.

6. Conclusions

The multi-level discount model developed in this paper obviously has a good recursive relation between the adjacent periods to simplify the computations and reduce the required memory in the computer. Many numbers of feasible solutions are cut at period t after improvement. This is obviously efficient than the past literatures.

In developing the dynamic lot-sizing model with multi-level discount, all the total cost of the feasible policies at period t in purchasing or production management are classified into four cases to establish the recursive relations and reduce the computer memory, such as $g(t, t)$, $g_n(t, t, k)$, $g(t, j)$ and $g_n(t, j, k)$. Both $g(t, t)$ and $g(t, j)$, which have a preceding policy at period t-1, can establish a recursive relationship shown in equation (6) and equation (7) between period t and period t-1. This recursive relation presented in this paper can simplify the computations. Besides, multi-level discount model may yield a new total cost of the feasible policy, $g_n(t, t, k)$, and $g_n(t, j, k)$, computed as equation (5). Finally, a forward dynamic lot-sizing model is expressed as equation (11) to improve the computation efficiency.

References

Chang, D. S., Chyr, F. C., & Yang, F. C. (2010). Incorporating a database approach into the large-scale multi-level lot sizing problem. *Computers & Mathematics with Applications*, *60*(9), 2536-2547.

Christoph, O. B. (1989). McLaren's order moment lot-sizing technique in multiple discounts. *Production and Inventory Management Journal*, *30*, 44-47.

Chung, C. S., Chiang, D. T., & Lu, C. Y. (1987). An optimal algorithm for the quantity discount problem. *Journal of Operations Management-Combined Issue*, *7*, 165-177.

Chyr, F. C. (1990). A new approach to the dynamic lot size model. *International Journal of Engineering Costs and Production Economic*, *20*, 255-263.

Chyr, F. C. (1993). A faster solution of the dynamic lot size model based on cost path concept. *Journal of the Chinese Institute of Industrial Engineers*, *10*, 149-154.

Chyr, F. C., Huang, S. T., & Lai, S. D. (1999). A dynamic lot-sizing model with quantity discount. *International Journal of Production Planning and Control*, *10*, 67-75.

Federgruen, A., & Tzur, M. (1991). A simple forward algorithm to solve general dynamic lot sizing models with n periods in O(n log n) or O(n) time. *Management Science*, *37*, 909-925.

Findt, M. L., Simon, S., & David, P. (2012). A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *European Journal of Operation Research*, *218*(3), 614-623.

Hu, J., & Munson, C. L. (2002). Dynamic demand lot-sizing rules for incremental quantity discounts. *Journal of the Operational Research Society*, *53*, 855-868.

Lee, J., Ristroph, J. H., Zhu, Z., & Ruangdet, M. (1993). Performance evaluation of lot-sizing methods with multiple quantity discounts in a rolling horizon environment. *Production and Inventory Management Journal*, *34*, 50-57.

Munson, C. L., & Rosenblatt, M. J. (2001). Coordinating a three-level supply chain with quantity discounts. *IIE Transaction*, *33*, 371-382.

Rubin, P. A., & Benton, W. C. (2003). Evaluating jointly constrained order quantity complexities for incremental discounts. *European Journal of Operational Research*, *149*, 557-570.

Yiyong, X., Ikou, K., Qiuhong, Z., & Rengian, Z. (2011). A reduced variable neighborhood search algorithm for uncapacitated multi-level lot-sizing problem. *European Journal of Operational Research*, *214*(2), 223-231.

Appendix

Theorem 1: Let Q_i be the quantity purchased at period i for $i > 1$. If a purchasing policy of $Q_i \neq 0$, $Q_i \neq D_i$, $Q_i \neq QB_k$, and $Q_i \neq (\sum_{a=1}^{r} D_a - \sum_{a=1}^{n-1} Q_a)$ for $r = n, n + 1, \ldots, T.$ holds, then the policy with Q_i does not need to be considered for optimal solution.

Proof: If the quantity purchased in period i ($i > 1$), Q_i, has the property of $Q_i \neq 0$, $Q_i \neq D_i$, $Q_i \neq QB_k$, and $Q_i \neq (\sum_{a=1}^{r} D_a - \sum_{a=1}^{n-1} Q_a)$ for $r = n, n + 1, \ldots, T.$, we can find another policy which has a lower total cost than that. Using the equation (5) to calculate the total cost of the feasible policy mentioned, the following cases can be demonstrated.

1. If $Q_i > QB_k, Q_i \neq D_i, Q_i \neq 0, Q_i \neq (\sum_{a=1}^{a \geq n} D_a - \sum_{a=1}^{n-1} Q_a), i < j$ and $Q_j \geq QB_k$ hold, then $(Q_1, \ldots, Q_{i-1}, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, Q_{i-1}, QB_k, \ldots, Q_j + (Q_i - QB_k), \ldots, Q_T)$ exists. For an example mentioned in section 2, the total cost of purchasing policy (50, 120, 50, 120) is larger than that of purchasing policy (50, 110, 50, 130).

2. If $Q_i > QB_k, Q_i \neq D_i, Q_i \neq 0, Q_i \neq (\sum_{i=1}^{r \geq n} D_i - \sum_{i=1}^{n-1} Q_i), i < j, Q_j < QB_k$ and $Q_j + (Q_i -$

$QB_k) \geq QB_k$ hold, then $(Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, QB_k, \ldots, Q_j + (Q_i - QB_k), \ldots, Q_T)$ exists.

3. If $Q_i > QB_k, Q_i \neq D_i, Q_i \neq 0, Q_i \neq (\sum_{i=1}^{r \geq n} D_i - \sum_{i=1}^{n-1} Q_i), i < j, Q_j < QB_k$ and $Q_j + (Q_i - QB_k) < QB_k$ hold, then we can find $(Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, QB_k, \ldots, Q_j + (Q_i - QB_k), \ldots, Q_T)$ or $(Q_1, \ldots, Q_i + Q_j, \ldots, 0, \ldots, Q_T) < (Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T)$.

4. If $Q_i < QB_k, Q_i \neq 0, Q_i \neq (\sum_{i=1}^{r \geq n} D_i - \sum_{i=1}^{n-1} Q_i), i < j$ and $Q_i > D_i$ hold, then $(Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, D_i, \ldots, Q_j + (Q_i - D_i), \ldots, Q_T)$ exists.

5. If $Q_i < QB_k, Q_i \neq 0, Q_i > (\sum_{i=1}^{k \geq n} D_i - \sum_{i=1}^{n-1} Q_i)$ and $Q_i < D_i$ hold, then $(Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, 0, \ldots, Q_j + Q_i, \ldots, Q_T)$ exists.

6. If $Q_i < QB_k, Q_i \neq D_i, Q_i \neq 0,$ and $Q_i > (\sum_{i=1}^{k \geq n} D_i - \sum_{i=1}^{n-1} Q_i)$ hold, then $(Q_1, \ldots, Q_i, \ldots, Q_j, \ldots, Q_T) > (Q_1, \ldots, (\sum_{i=1}^{k \geq n} D_i - \sum_{i=1}^{n-1} Q_i), \ldots, Q_j + Q_i - (\sum_{i=1}^{k \geq n} D_i - \sum_{i=1}^{n-1} Q_i), \ldots, Q_T)$ exists.

If the other cases satisfy the conditions of $Q_i \neq QB_j$, $Q_i \neq D_i$, $Q_i \neq 0$, and $Q_i \neq (\sum_{i=1}^{k \geq n} D_i - \sum_{i=1}^{n-1} Q_i)$, we still can find a policy, which has a lower total cost than the original one.