Journal of the Operations Research
Society of Japan
Vol. 34, No. 1, March 1991

A HEURISTIC ALGORITHM FOR THE DETERMINISTIC MULTI-PRODUCT INVENTORY SYSTEM WITH CAPACITY CONSTRAINT

Maw-Sheng Chern Kao-Chêng Lin Chin-Chern Chen
National Tsing Hua University

(Received March 22, 1988; Final November 26, 1990)

Abstract This paper investigates an m-product inventory system ($m \geq 3$) with a capacity constraint where products can have individual order intervals and orders be phased to reduce the maximum stock level of all the products on hand. The objective is then to find the optimal order quantity of each product by considering staggering time and order interval which minimizes the system cost per unit time. The problem is described in a non-linear integer programming problem which shows a very complicated nature to derive the solution analytically. Therefore, a heuristic algorithm is proposed and tested for its efficiency with various numerical examples as being superior to either the Lagrangian multiplier method or the fixed cycle method.

1. Introduction

This paper considers a multi-product inventory ordering problem of minimizing the average inventory cost subject to a capacity constraint. The Lagrangian multiplier method [3, 7] and fixed cycle method [2, 6, 8, 10] have been used for solving such a problem. However, the Lagrangian multiplier method implicitly assumes that all products will be ordered simultaneously, or at least within an arbitrarily small time interval. Thus, the warehouse space utilization is about 50% and it usually results in higher average cost per unit time. Although the fixed cycle method allows the phasing of orders for the different products, it assumes that all products have the same order interval. In some cases, this restriction may lead to a higher average cost per unit time than that of the Lagrangian multiplier method. Hartley and Thomas [4, 9] and Chen *et al.* [1] propose an alternative approach in which different products may have different order intervals and these orders may be phased in certain cases to avoid having the maximum stock levels of different products on hand at the same time (see, for example, Figure 2). For the two - product problem we may assume without loss of generality that one of the staggering times is zero and the maximum stock level over time can be expressed explicitly as a function of the order intervals and the nonzero staggering time [4, 9]. Moreover, we can minimize the maximum stock level with respect to the staggering time and express it as a function of the order intervals. The average cost per unit time can then be minimized by using such an expression.

However, for m-product problems, $m \geq 3$, it is difficult to derive a similar relationship among the maximum stock level, order intervals and staggering times. The problem rather reveals so complex nature that it is hard to solve. Therefore, this paper proposes a heuristic algorithm which is tested with various numerical examples for its superiority to the Lagrangian multiplier method and the fixed cycle method. It is shown that the proposed algorithm gives better solutions than the latter two methods.

2. Mathematical Formulation

Assumptions:

1. Demand can be approximated by a constant average demand over time.
2. Planning horizon is infinite.
3. Shortage is not allowed.
4. Delivery is instantaneously made.
5. Overall stock level is periodic.

Notation:

$c_i =$ the order cost of product i. (Assuming that the order cost is independent of the order quantity.)
$v_i =$ the volume of product i per unit.
$d_i =$ the demand of product i per unit time.
$h_i =$ the holding cost of one unit of product i per unit time.
$q_i =$ the order quantity of product i.
$t_i =$ the order interval of product i.($q_i = d_i t_i$.)
$s_i =$ the staggering time of product i.
$M =$ the warehouse capacity.

The total variable cost is the sum of order costs and holding costs. For given c_i, v_i, d_i, h_i, $i = 1, \cdots, m$, and M, we let Z be the total variable cost per unit time and $S_{\max}(\mathbf{t}, \mathbf{s})$ be the maximum stock level. Then, the problem is mathematically stated as follows:

$$\text{minimize } Z = \sum_{i=1}^{m} \frac{c_i}{t_i} + 0.5 \sum_{i=1}^{m} h_i d_i t_i, \tag{1}$$

$$\text{subject to } S_{\max}(\mathbf{t}, \mathbf{s}) \leq M, s_i \geq 0, t_i > 0, i = 1, \cdots, m, \tag{2}$$

where $\mathbf{t} = (t_1, \cdots, t_m)$ and $\mathbf{s} = (s_1, \cdots, s_m)$. The objective is to find $(\mathbf{t}, \mathbf{s})$ that minimizes Z and satisfies the constraint (2).

3. Staggering the Initial Orders

Let $\lfloor x_0 \rfloor$ denote the largest integer which is smaller than or equal to x_0. Then, given the order interval t_i and the staggering times $s_i, 0 \leq s_i < t_i, i = 1, \cdots, m$, the stock volume of product i is

$$f_i(t) = v_i d_i t_i + v_i d_i s_i - v_i d_i (t - \lfloor \frac{t - s_i}{t_i} \rfloor t_i).$$

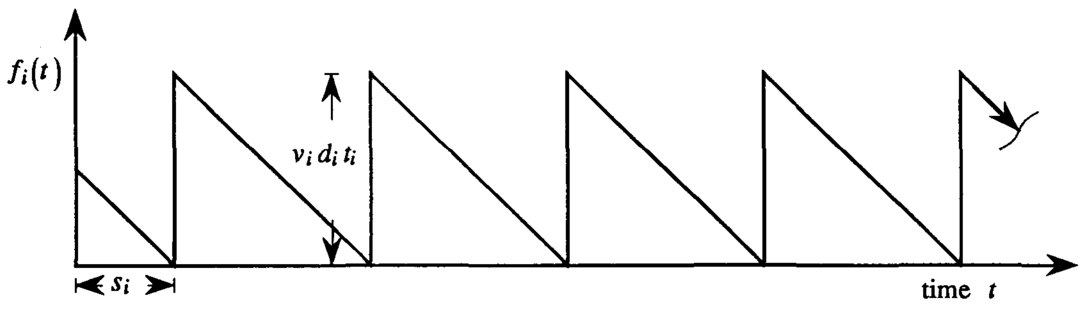

Figure 1. Stock level of product i.

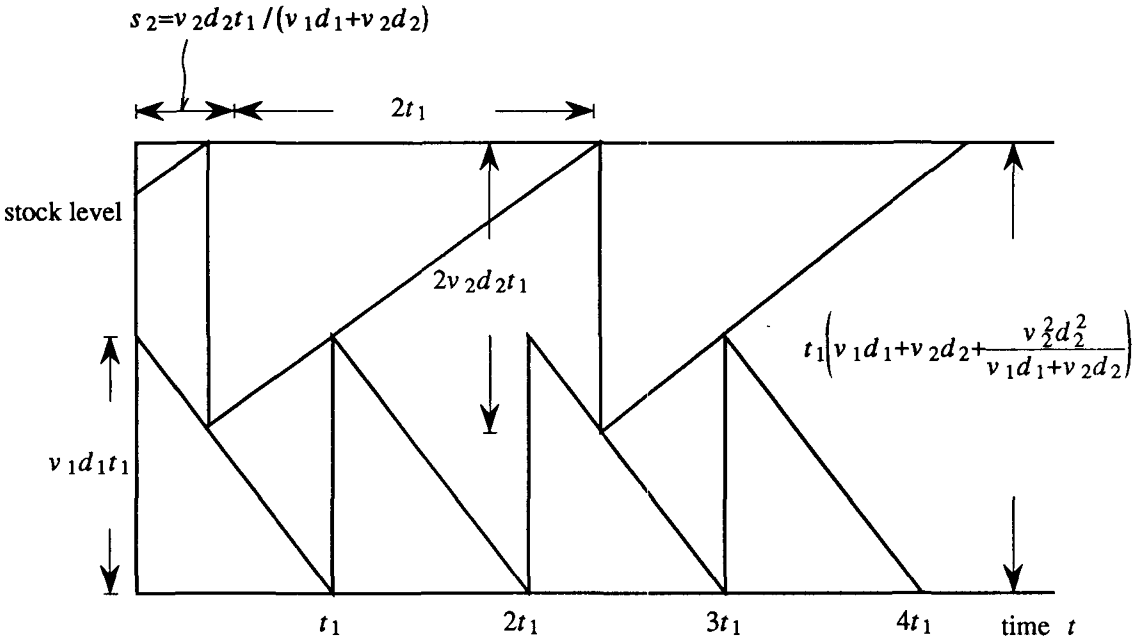

Figure 2. Stock level over time with $t_2 = 2t_1$[2].

If $t_i, i = 1,\cdots,m$, are rational numbers, then $f_i(t)$ is periodic with the period t_i and $\sum_{i=1}^m f_i(t)$ is also periodic with the period $T = \min\{t \mid t$ is an integral multiple of $t_i, i = 1,\cdots,m\}$.

Theorem 1. Let $t_i, i = 1,\cdots,m$, be the period of $f_i(t)$. Then

$$\begin{aligned} S_{\max}(\mathbf{t},\mathbf{s}) &= \max\{\sum_{i=1}^m f_i(t) \mid 0 \le t \le T\} \\ &= \sum_{i=1}^m v_id_it_i - \min\{S_k(s_1,\cdots,s_m) \mid k = 1,\cdots,m\}, \text{ where} \end{aligned}$$

$$\begin{aligned} S_k(s_1,\cdots,s_m) = \min_{\ell_{kk},\ell_{ki}} \{\sum_{i\neq k} v_id_i(s_k + \ell_{kk}t_k - s_i - \ell_{ki}t_i) \mid \ell_{ki} \in [0, n_i - 1], \ell_{kk} \in [0, n_k - 1], \\ n_i = T/t_i, \ell_{ki}, \ell_{kk} \text{ integer and } s_k + \ell_{kk}t_k - s_i - \ell_{ki}t_i \ge 0 \text{ for each } i.\}. \end{aligned}$$

Proof. It is clear that the maximum stock level will be attained immediately after one of the m products is ordered. Without loss of generality, we assume that the maximum stock level is attained at $t = \eta_k$ when the product k is ordered.

Let $\eta_k - s_k = \underline{\ell}_{kk} t_k$ where $\underline{\ell}_{kk} = \lfloor(\eta_k - s_k)/t_k\rfloor$ is a nonnegative integer.

$$\begin{aligned}\sum_{i=1}^{m} f_i(\eta_k) &= \sum_{i=1}^{m}[v_i d_i t_i + v_i d_i s_i - v_i d_i(\eta_k - \lfloor(\eta_k - s_i)/t_i\rfloor t_i)] \\ &= \sum_{i=1}^{m} v_i d_i t_i - \sum_{i\neq k} v_i d_i(\eta_k - s_i - \lfloor(\eta_k - s_i)/t_i\rfloor t_i) \\ &= \sum_{i=1}^{m} v_i d_i t_i - \sum_{i\neq k} v_i d_i(s_k + \underline{\ell}_{kk} t_k - s_i - \underline{\ell}_{ki} t_i) \\ &\quad \text{(where } \underline{\ell}_{ki} = \lfloor(s_k + \underline{\ell}_{kk} t_k - s_i)/t_i\rfloor \text{and } \underline{\ell}_{ki} \in [0, n_i - 1] \text{ for each } i.) \\ &= \sum_{i=1}^{m} v_i d_i t_i - \min_{(\ell_{ki})}\{\sum_{i\neq k} v_i d_i(s_k + \underline{\ell}_{kk} t_k - s_i - \ell_{ki} t_i) \mid \ell_{ki} \in [0, n_i - 1], \ell_{ki} \text{ integer}, \\ &\quad s_k + \underline{\ell}_{kk} t_k - s_i - \ell_{ki} t_i \geq 0 \text{ and } n_i = T/t_i \text{ for each } i.\} \\ &= \sum_{i=1}^{m} v_i d_i t_i - S_k(s_1, \cdots, s_m).\end{aligned}$$

Hence, the proof is completed. □

By Theorem 1, we can also formulate the problem (1) – (2) as a mixed-integer programming problem as follows:

$$\text{minimize } Z = \sum_{i=1}^{m} \frac{c_i}{t_i} + 0.5 \sum_{i=1}^{m} h_i d_i t_i, \tag{3}$$

$$\text{subject to } \sum_{i=1}^{m} v_i d_i t_i - \sum_{i\neq k} v_i d_i(s_k + \ell_{kk} t_k - s_i - \ell_{ki} t_i) \leq M, k = 1, \cdots, m, \tag{4}$$

$$s_k + \ell_{kk} t_k - s_i - \ell_{ki} t_i \geq 0, i = 1, \cdots, m, k = 1, \cdots, m, i \neq k, \tag{5}$$

$$0 \leq \ell_{ki} \leq n_i - 1, n_i = \frac{T}{t_i}, i = 1, \cdots, m, \tag{6}$$

$$s_i \geq 0, T > 0, t_i > 0, \ell_{ki} \text{ integer}, i = 1, \cdots, m, k = 1, \cdots, m. \tag{7}$$

Although a branch and bound method [5] can be designed for solving (3) – (7), the complexity of constraints reveals that it may take a large amount of computational time to solve this problem even when $m = 3$, because for some problem instances a large number of ℓ_{ki}'s must be enumerated. Therefore, some approaches have been proposed for solutions in the subspaces of (4) – (7). For example, the Lagrangian multiplier method searches for the optimal solution in the subspace of (4) – (7) with $s_1 = s_2 = \cdots = s_m = 0$ and the fixed cycle method searches for the optimal solution in the subspace with $t_1 = t_2 = \cdots = t_m$. In this paper, however, we propose an algorithm which searches for a solution of (3) – (7) such that s_i and $t_i, i = 1, \cdots, m$ satisfy some specific relationships.

For $m = 2$, we may assume without loss or generality that $s_1 = 0$. By Theorem 1,

$$\max_{0\leq t\leq T}\{f_1(t) + f_2(t)\} = v_1 d_1 t_1 + v_2 d_2 t_2 - \min\{S_1(0, s_2), S_2(0, s_2)\}. \tag{8}$$

It has been shown [4] that for given t_1 and t_2, the optimal solution of (8) satisfies the following condition

$$s_2 = \frac{T}{n_1 n_2} \frac{v_2 d_2}{v_1 d_1 + v_2 d_2}, \tag{9}$$

and the maximum stock level is

$$\max_{0\le t\le T}\{f_1(t)+f_2(t)\} = v_1d_1t_1 + v_2d_2t_2 - \frac{T}{n_1n_2}\frac{v_1d_1v_2d_2}{v_1d_1+v_2d_2}. \tag{10}$$

In this case, the optimal solution of (1) – (2) can then be solved by using (9) – (10) [9].

For $m \ge 3$, it is difficult to derive a similar relationship among the optimal staggering times s_i and the order intervals $t_i, i = 1,\cdots,m$, as that in (9). Therefore, we propose a heuristic approach for scheduling the initial orders. We denote that

$$T_{ij} = T_{ji} = n_{ij}t_i = n_{ji}t_j = \min\{t \mid t \text{ is an integral multiple of } t_i \text{ and } t_j.\},$$

where n_{ij} and n_{ji} are co-prime, i.e. $\gcd(n_{ij}, n_{ji}) = 1$. For given k and $(r_i)_{i\ne k}, \sum_{i\ne k} r_i = 1$, we consider product i and r_i proportion of product k as a two-product problem and let the staggering times be of the form as that in (9). Therefore, if we let

$$s_i = \frac{T_{ki}}{n_{ki}n_{ik}}\frac{v_id_i}{r_iv_kd_k+v_id_i}, i = 1,\cdots,m, i\ne k, \tag{11}$$

and $s_k = 0$, then by (9) – (10) the maximum stock level is not greater than

$$\min\{\sum_{i=1}^{m} v_id_it_i - \sum_{i\ne k}\frac{T_{ki}}{n_{ki}n_{ik}}\frac{r_iv_kd_kv_id_i}{r_iv_kd_k+v_id_i} \mid \sum_{i\ne k} r_i = 1, r_i \ge 0 \text{ for each } i.\}. \tag{12}$$

When $m = 2$, (11) is reduced to (9) and (12) is reduced to (10) with $k = 1, s_1 = 0, r_1 = 0, n_{12} = n_1$ and $n_{21} = n_2$. Moreover, in Figure 3 and Figure 4, we consider a three-product problem. For fixed r, if we let the staggering times be $s_1 = 0, s_2 = (T_{12}/n_{12}n_{21})[v_2d_2/(rv_1d_1+v_2d_2)]$ and $s_3 = (T_{13}/n_{13}n_{31})[v_3d_3/((1-r)v_1d_1+v_3d_3)]$, then the stock volume of product 2 with $r100\%$ proportion of product 1 is not greater than

$$rv_1d_1t_1 + v_2d_2t_2 - \frac{T_{12}}{n_{12}n_{21}}\frac{rv_1d_1v_2d_2}{rv_1d_1+v_2d_2}$$

and the stock volume of product 3 with $(1-r)100\%$ proportion of product 1 is not greater than

$$(1-r)v_1d_1t_1 + v_3d_3t_3 - \frac{T_{13}}{n_{13}n_{31}}\frac{(1-r)v_1d_1v_3d_3}{(1-r)v_1d_1+v_3d_3}.$$

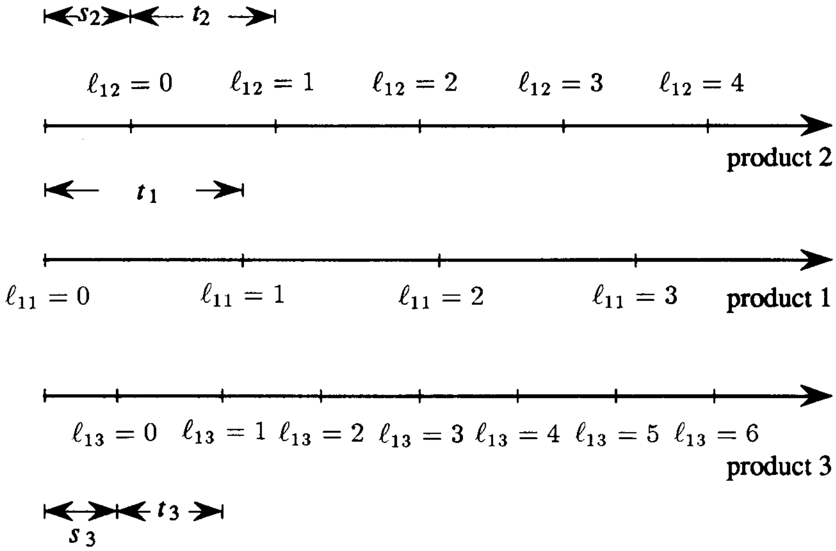

Figure 3. Staggering times and order intervals of three products.

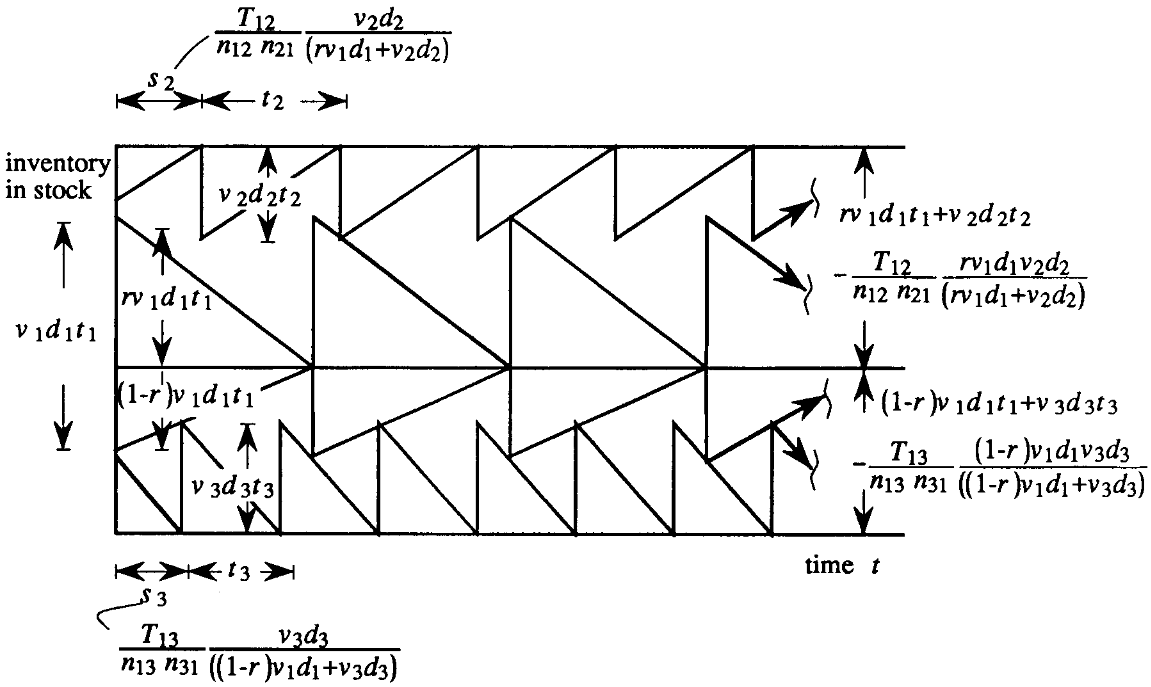

Figure 4. Stock level of three-product inventory over time.

For each $k, 1 \leq k \leq m$, we consider a simplified problem which is obtained from (1) – (2) by replacing (2) with an approximation of capacity constraint by using (12).

$$\text{minimize} \sum_{i=1}^{m} \frac{c_i}{t_i} + 0.5 \sum_{i=1}^{m} h_i d_i t_i, \tag{13}$$

$$\text{subject to } \sum_{i=1}^{m} v_i d_i t_i - \sum_{i \neq k} \frac{T_{ki}}{n_{ki} n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i} \leq M, \tag{14}$$

$$T_{ij} = T_{ji} = n_{ij} t_i = n_{ji} t_j = \min\{t \mid t \text{ is an integral multiple of } t_i \text{ and } t_j.\}, \tag{15}$$

$$\sum_{i \neq k} r_i = 1, r_i \geq 0, t_i > 0, n_{ik} > 0, n_{ki} > 0, n_{ik}, n_{ki} \text{ integer}. \tag{16}$$

For each k, we will solve (13) – (16) for an approximate solution of (1) – (2). We then select the best one of these m solutions. In the following sections, a procedure is proposed for solving (13) – (16).

4. A Heuristic Method

For given k, (13) – (16) can be formulated as follows:

$$\text{minimize } W = (c_k + \sum_{i \neq k} \frac{n_{ik}}{n_{ki}} c_i) \frac{1}{t_k} + 0.5(h_k d_k + \sum_{i \neq k} \frac{n_{ki}}{n_{ik}} h_i d_i) t_k, \tag{17}$$

$$\text{subject to } (v_k d_k + \sum_{i \neq k} \frac{n_{ki}}{n_{ik}} v_i d_i - \sum_{i \neq k} \frac{1}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i}) t_k \leq M, \tag{18}$$

$$\sum_{i \neq k} r_i = 1, r_i \geq 0, t_k > 0, n_{ik} > 0, n_{ki} > 0, n_{ik}, n_{ki} \text{ integer}, \tag{19}$$

where (17) and (18) are derived from (13) and (14) respectively. For given $(n_{ik})_{i \neq k} = (\underline{n}_{ik})_{i \neq k}$, we can further convert this mixed-integer programming problem into a simplified problem by minimizing the objective function W with respect to t_k. Note that W is a convex function of t_k. For $i \neq k$, let $a_i = n_{ki}/\underline{n}_{ik}$. The solution of $\partial W / \partial t_k = 0$ is

$$\underline{t}_k = \sqrt{2(c_k + \sum_{i \neq k} \frac{c_i}{a_i}) / (h_k d_k + \sum_{i \neq k} a_i h_i d_i)}.$$

Substituting t_k by $\underline{t}_k$ in (17), we have

$$W = \Phi(a_1, \cdots, a_{k-1}, a_{k+1}, \cdots, a_m) = \sqrt{2(c_k + \sum_{i \neq k} \frac{c_i}{a_i})(h_k d_k + \sum_{i \neq k} a_i h_i d_i)}.$$

Consider the problem

$$\text{minimize}_{a_i \geq 0} \ \Phi(a_1, \cdots, a_{k-1}, a_{k+1}, \cdots, a_m) \tag{20}$$

and let $(a_i^*)_{i \neq k}$ be an optimal solution of (20). Because W is a convex function of $(n_{ki})_{i \neq k}$, the optimal solution $(n_{ki})_{i \neq k}$ associated with $(a_i^*)_{i \neq k}$ would be $n_{ki} = \lfloor \underline{n}_{ik} a_i^* \rfloor$ or $\lfloor \underline{n}_{ik} a_i^* \rfloor + 1$ for each $i \neq k$. For given $(n_{ki})_{i \neq k}$ where $n_{ki} = \lfloor \underline{n}_{ik} a_i^* \rfloor$ or $\lfloor \underline{n}_{ik} a_i^* \rfloor + 1$, we then search for an optimal solution $(r_i^*)_{i \neq k}$ which minimizes the left-hand side of (18) with $t_k = \underline{t}_k$ and $(n_{ik})_{i \neq k} = (\underline{n}_{ik})_{i \neq k}$.

Thus, if

$$(v_k d_k + \sum_{i \neq k} a_i^* v_i d_i - \sum_{i \neq k} \frac{1}{\underline{n}_{ik}} \frac{r_i^* v_k d_k v_i d_i}{r_i^* v_k d_k + v_i d_i}) \underline{t}_k \leq M, \tag{21}$$

then

$$\min\{\Phi \mid n_{ki} = \lfloor \underline{n}_{ik} a_i^* \rfloor, \lfloor \underline{n}_{ik} a_i^* \rfloor + 1, i = 1, \cdots, m, i \neq k \ ,$$

$$(v_k d_k + \sum_{i\neq k} \frac{n_{ki}}{\underline{n}_{ik}} v_i d_i - \sum_{i\neq k} \frac{1}{\underline{n}_{ik}} \frac{r_i^* v_k d_k v_i d_i}{r_i^* v_k d_k + v_i d_i}) \underline{t}_k \leq M\}$$

yields an optimal solution for (17) - (19) with $t_k = \underline{t}_k$ and $(n_{ik})_{i\neq k} = (\underline{n}_{ik})_{i\neq k}$. Otherwise, by the property of complementary slackness, the capacity constraint (18) is tight. If the capacity constraint is tight, we let $\underline{t}_k' = M/\xi$ where

$$\xi = v_k d_k + \sum_{i\neq k} a_i v_i d_i - \sum_{i\neq k} \frac{1}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i}.$$

Substituting t_k by $\underline{t}_k'$ in (17), we have

$$\begin{aligned} W &= \Psi(a_1, \cdots, a_{k-1}, a_{k+1}, \cdots, a_m, r_1, \cdots, r_{k-1}, r_{k+1}, \cdots, r_m) \\ &= (c_k + \sum_{i\neq k} \frac{c_i}{a_i}) \frac{\xi}{M} + 0.5(h_k d_k + \sum_{i\neq k} a_i h_i d_i) \frac{M}{\xi}. \end{aligned}$$

Then, (17) - (19) can be written as follows:

$$\text{minimize } \Psi(a_1, \cdots, a_{k-1}, a_{k+1}, \cdots, a_m, r_1, \cdots, r_{k-1}, r_{k+1}, \cdots, r_m), \tag{22}$$

$$\text{subject to } \sum_{i\neq k} r_i = 1, r_i \geq 0, a_i \geq 0, \text{for all } i. \tag{23}$$

Similarly, we solve (22) - (23) for a solution $(a_i^*, r_i^*)_{i\neq k}$.

Hence, for given $(\underline{n}_{ik})_{i\neq k}$ and $(a_i^*)_{i\neq k}$, if the capacity constraint (21) is satisfied with $(r_i)_{i\neq k} = (r_i^*)_{i\neq k}$, then an approximate solution of (13) - (16) can be obtained by solving

$$\begin{aligned} &\min\{\Phi \mid n_{ki} = \lfloor \underline{n}_{ik} a_i^* \rfloor, \lfloor \underline{n}_{ik} a_i^* \rfloor + 1, i = 1, \cdots, m, i \neq k, \\ &(v_k d_k + \sum_{i\neq k} \frac{n_{ki}}{\underline{n}_{ik}} v_i d_i - \sum_{i\neq k} \frac{1}{\underline{n}_{ik}} \frac{r_i^* v_k d_k v_i d_i}{r_i^* v_k d_k + v_i d_i}) \underline{t}_k \leq M\}. \end{aligned} \tag{24}$$

Otherwise, a solution is obtained by solving

$$\begin{aligned} &\min\{\Psi \mid n_{ki} = \lfloor \underline{n}_{ik} a_i^* \rfloor, \lfloor \underline{n}_{ik} a_i^* \rfloor + 1, i = 1, \cdots, m, i \neq k, \\ &(v_k d_k + \sum_{i\neq k} \frac{n_{ki}}{\underline{n}_{ik}} v_i d_i - \sum_{i\neq k} \frac{1}{\underline{n}_{ik}} \frac{r_i^* v_k d_k v_i d_i}{r_i^* v_k d_k + v_i d_i}) \underline{t}_k \leq M\}. \end{aligned} \tag{25}$$

We conclude that a solution (t_i^*) of (13) - (16) can be obtained by solving either (24) or (25) for all possible values of $\underline{n}_{ik}$. In order to solve (24) or (25) efficiently, we will derive an upper bound of n_{ik} for each $i \neq k$ in the next section.

5. An Upper Bound of n_{ik}

Let

$$\mu(B) = \min\{\sum_{i=1}^{m} \frac{c_i}{t_i} + 0.5 \sum_{i=1}^{m} h_i d_i t_i \mid \sum_{i=1}^{m} v_i d_i t_i \leq M + B, t_i > 0, i = 1, \cdots, m\}. \tag{26}$$

If $B = 0$, then $\mu(0)$ is a convex program which can be solved by the Lagrangian multiplier method [3]. Assume that λ^* is an optimal multiplier of (26) with $B = 0$. Then

$$\mu(B) \geq \mu(0) - \lambda^* B \text{ for any } B \geq 0 \text{ [9]}.$$

For given k, let

$$B_k = \sum_{i \neq k} \frac{t_k}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i}$$

and U be the objective value associated with a feasible solution of (1) – (2) such that $\mu(B_k) \leq U \leq \mu(0)$. Then an upper bound u_{ik} of n_{ik} can be derived by the following inequality

$$U \geq \mu(0) - \lambda^* B_k.$$

Note that the total volume of product k can not be greater than M, i.e. $t_k \leq \frac{M}{v_k d_k}$. Therefore,

$$U \geq \mu(0) - \lambda^* (\sum_{i \neq k} \frac{1}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i})(\frac{M}{v_k d_k}),$$

$$\begin{aligned} \mu(0) - U &\leq \lambda^* (\sum_{i \neq k} \frac{1}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k})(\frac{M}{v_k d_k}) \\ &\leq \lambda^* (\frac{1}{n_{ik}} v_i d_i + \sum_{j \neq i,k} v_j d_j)(\frac{M}{v_k d_k}), i \neq k. \end{aligned}$$

Hence,

$$n_{ik} \leq \lambda^* v_i d_i [(\mu(0) - U)(\frac{v_k d_k}{M}) - \lambda^* \sum_{j \neq i,k} v_j d_j]^{-1} = u_{ik}, i \neq k. \quad (27)$$

6. Algorithms and Computational Results

Based on the theoretical results presented in the previous sections, we now propose a heuristic algorithm for solving the problem (1) – (2).

Algorithm 1:

Step 1: Solve (1) without capacity constraint (2) and obtain the Economic Order Quantity (EOQ) solution $s_i = 0, t_i = \sqrt{2c_i / h_i d_i}, i = 1, \cdots, m$. If the EOQ solution satisfies (2), then it is optimal and stop. Otherwise, go to Step 2.

Step 2: Solve (26) with $B = 0$ for the optimal Lagrangian multiplier λ^* and let $U = \mu(0)$.

Step 3: For $k = 1, \cdots, m$,perform Step 4 to Step 9.

Step 4: If a feasible solution of (1) – (2) with the objective value $U', \mu(B_k) \leq U' < U$, is available, then U is replaced by U'. For each $i, i \neq k$, calculate the upper bound u_{ik} of n_{ik} according to (27).

Step 5: For each $(\underline{n}_{ik})_{i \neq k}, 1 \leq \underline{n}_{ik} \leq u_{ik}, i \neq k$, perform Step 6 to Step 8.

Step 6: Solve $(a_i^*)_{i \neq k}$ and $(r_i^*)_{i \neq k}$ such that Φ is minimized. If (21) is satisfied with $(a_i)_{i \neq k} = (a_i^*)_{i \neq k}$ and $(r_i)_{i \neq k} = (r_i^*)_{i \neq k}$, then solve (24) for given $(\underline{n}_{ik})_{i \neq k}$ and go to Step 8. Otherwise, go to Step 7.

Step 7: Solve $(a_i^*)_{i \neq k}$ and $(r_i^*)_{i \neq k}$, such that Ψ is minimized and solve (25) for given $(\underline{n}_{ik})_{i \neq k}$.

Step 8: If a feasible solution of (1) – (2) with the objective value $U', \mu(B_k) \leq U' < U$, is obtained, then U is replaced by U' and $u_{ik}, i \neq k$, are updated by using (27).

Step 9: Set $L_k = U$.

Step 10: Output the heuristic solution with the objective value $L^* = \min\{L_k \mid k = 1, \cdots, m\}$.

A reasonable way to improve the current feasible solution (t_i, s_i) is to search for a better solution in its neighborhood. In the following, we propose such an algorithm.

Algorithm 2:

Step 1: Give a positive constant δ and positive integers ζ and N. δ/ζ is the step size for the neighborhood search and N is the maximum number of iterations allowed in the algorithm. Set $t_i^* = t_i, s_i^* = s_i, i = 1, \cdots, m$ and $I = 0$ (I is a counter of the number of iterations.)

Step 2: Set $I = I + 1$. If $I \leq N$, perform Step 3 and Step 4 for all the combinations of integer values $\phi_1, \cdots, \phi_m$ such that $|\phi_i| \leq \zeta, i = 1, \cdots, m$. Otherwise, go to Step 5.

Step 3: Set $t_i = t_i^* + \phi_i(\delta/\zeta), i = 1, \cdots, m$. If the objective value Z associated with (t_i) is less than that associated with (t_i^*), then go to Step 4. Otherwise, check the next combination of integer values $\phi_1, \cdots, \phi_m$.

Step 4: Calculate the n_{ij}'s and solve

$$\min_k \min_{(r_i)} \{(v_k d_k + \sum_{i \neq k} \frac{n_{ki}}{n_{ik}} v_i d_i - \sum_{i \neq k} \frac{1}{n_{ik}} \frac{r_i v_k d_k v_i d_i}{r_i v_k d_k + v_i d_i}) t_k \mid \sum_{i \neq k} r_i = 1, r_i \geq 0\}.$$

If the optimal value is not greater than M for $k = k_0$ and $(r_i)_{i \neq k_0} = (r_i^*)_{i \neq k_0}$, then set $s_{k_0}^* = 0, s_i^* = (t_{k_0}/n_{ik_0})[v_i d_i/(r_i^* v_{k_0} d_{k_0} + v_i d_i)], i = 1, \cdots, k_0 - 1, k_0 + 1, \cdots, m$ and $t_i^* = t_i, i = 1, \cdots, m$, and go to Step 2. Otherwise, check the next combination of integer values $\phi_1, \cdots, \phi_m$.

Step 5: Output the solution (t_i^*) and $(s_i^*), i = 1, \cdots, m$.

In the following, we consider three-product problems with various warehouse capacities. These problems are solved by three different approaches and their computational results are shown in Table 3.

Table 1. A three-product problem.

i	1	2	3	
c_i	50	50	50	$m = 3$
h_i	10	4	16	$M = 15000$
d_i	1000	1000	2000	
v_i	50	20	80	

Table 2. The solutions of EOQ method and the proposed algorithm.

Solution Method	t_1	t_2	t_3	Total Cost	Maximum Stock Level
EOQ	0.1000	0.1581	0.0559	3421.11	17106
Algorithm 1	0.1106	0.1659	0.0553	3427.20	15000
	$s_1 = 0.0000$	$s_2 = 5.5 \times 10^{-9}$		$s_3 = 0.0385$	

Table 3. Solution inventory costs from four algorithms.

Warehouse Capacity M	Lagrangian Multiplier Method (A)	Fixed Cycle Method (B)	Algorithm 1 (C)	Algorithm 1& Neighborhood Search	(C)/(A) 100%	(C)/(B) 100%
100	292644.04	265447.78	249574.47	249574.47	85.28	94.02
200	146337.02	132743.39	124804.86	124804.86	85.29	94.02
300	97574.68	88517.25	83222.83	83222.83	85.29	94.02
400	73198.51	66410.69	62437.69	62437.69	85.30	94.02
500	58576.81	53151.94	49971.31	49971.31	85.31	94.02
600	48832.34	44317.12	41664.31	41664.31	85.32	94.01
700	41874.86	38010.24	35734.09	35734.09	85.34	94.01
800	36659.25	33283.33	31289.37	31289.37	85.35	94.01
900	32604.89	29609.73	27834.98	27834.98	85.37	94.01
1000	29363.40	26673.45	25073.81	25073.81	85.39	94.00
2000	14831.70	13531.69	12713.21	12713.21	85.72	93.95
3000	10054.47	9237.75	8671.37	8671.37	86.24	93.87
4000	7715.85	7155.77	6709.22	6709.22	86.95	93.75
5000	6352.69	5958.57	5578.96	5578.96	87.82	93.63
6000	5477.23	5203.77	4864.61	4864.61	88.82	93.48
7000	4880.49	4701.75	4387.95	4387.95	89.91	93.33
8000	4457.93	4357.73	4059.85	4057.85	91.07	93.16
9000	4151.49	4119.05	3830.77	3830.77	92.27	93.00
10000	3926.34	3954.10	3666.44	3666.44	93.38	92.73
11000	3760.31	3842.77	3558.34	3558.34	94.63	92.59
12000	3638.62	3771.66	3487.85	3487.85	95.86	92.48
13000	3551.03	3731.48	3448.28	3448.28	97.11	92.41
14000	3490.24	3715.61	3428.29	3428.29	98.23	92.27
15000	3450.89	3714.84	3427.20	3427.20	99.94	92.26
16000	3428.96	3714.84	3427.00	3427.00	99.94	92.25
17000	3421.38	3714.84	3421.38	3421.36	100.00	92.10
17106	3421.11	3714.84	3421.11	3421.11	100.00	92.09

In Table 2 and Table 3, we note that if $M \geq 17106$, then the problem can be solved without capacity constraint. In this case, both the EOQ method and the Lagrangian multiplier method yield the optimal solution. It can be seen that as the restriction on the warehouse capacity gets tighter, the fixed cycle method is often better that the Lagrangian multiplier method. However, for some problem instances the fixed cycle method may never generate an optimal solution for any given $M > 0$. In fact, these two methods do not solve the problem in an optimal way. Thus, when faced with the warehouse capacity restriction, it is essential to use the staggering policy and allow different products to have different order intervals. In Table 3, we also note that the proposed algorithm is significantly better than the two methods mentioned above.

Acknowledgements

The authors would like to thank the Editor and two anonymous referees for their helpful comments and suggestions.

References

[1] C.-C. Chen, M.-S. Chern and K.-C. Lin: Capacitated Multiple Products Inventory System, *Recent Developments in Production Research* (Editor: Anil Mital), Elsevier Science Publishers B.V., Amsterdam (1988), 194–199.

[2] S.K. Goyal: A Note on 'Multi-Product Inventory Situations with one Restriction', *Journal of the Operational Research Society*, Vol. 29, (1978), 269–271.

[3] G. Hadley and J.M. Whitin, *Analysis of Inventory System*, Prentice-Hall, Englewood Cliffs, N.J., 1963.

[4] R. Hartley and L.C. Thomas: The Deterministic, Two-Product, Inventory System with Capacity Constraint, *Journal of the Operational Research Society*, Vol. 33 (1982), 1013–1020.

[5] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD, 1978.

[6] E. Page and K.J. Paul: Multi-Product Inventory Situations with one Restriction, *Operational Research Quarterly*, Vol. 27 (1976), 815–834.

[7] H. Pirkul and O.A. Aras: Capacitated Multiple Item Ordering Problem with Quantity Discount, *IIE Transactions*, Vol. 17 (1985), 206–211.

[8] H.J. Rosenblatt: Multi-Item Inventory System with Budgetary Constraint: A Comparison Between the Lagrangian and Fixed Cycle Approach, *International Journal of Production Research*, Vol. 19 (1981), 331–339.

[9] L.C. Thomas and R. Hartley: An Algorithm for the Limited Capacity Inventory Problem with Staggering, *Journal of the Operational Research Society*, Vol. 34 (1983), 81–85.

[10] K. Zoller: Deterministic Multi-Item Inventory Systems with Limited Capacity, *Management Science*, Vol. 24 (1977), 451–455.

Maw-Sheng Chern
Dept. of Industrial Engineering
National Tsing Hua University
Hsinchu (30043)
Taiwan