OPERATIONS RESEARCH
Vol. 55, No. 5, September–October 2007, pp. 854–865
ISSN 0030-364X | EISSN 1526-5463 | 07 | 5505 | 0854

DOI 10.1287/opre.1070.0394

Storage-Space Capacitated Inventory System with (r, Q) Policies

Xiaobo Zhao, Fan Fan, Xiaoliang Liu
Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
{xbzhao@tsinghua.edu.cn, ffan02@mails.tsinghua.edu.cn, justinliu97@gmail.com}

Jinxing Xie
Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China,
jxie@math.tsinghua.edu.cn

We deal with an inventory system with limited storage space for a single item or multiple items. For the single-item system, customers' demand is stochastic. The inventory is controlled by a continuous-review (r, Q) policy. Goods are replenished to the inventory system with a constant lead time. An optimization problem with a storage-space constraint is formulated for computing a single-item (r, Q) policy that minimizes the long-run average system cost. Based on some existing results in the single-item (r, Q) policy without a storage-space constraint in the literature, useful structural properties of the optimization problem are attained. An efficient algorithm with polynomial time computational complexity is then proposed for obtaining the optimal solutions. For the multi-item system, each item possesses its particular customers' demand that is stochastic, its own (r, Q) policy that controls the inventory, and its individual lead time that is constant. An important issue in such inventory systems is the allocation of the storage space to the items and the values of r and Q for each item. We formulate an optimization problem with a storage-space constraint for multi-item (r, Q) policies. Based on the results in the single-item (r, Q) policy with a storage-space constraint, we find useful structural properties of the optimization problem. An efficient algorithm with polynomial time computational complexity is then proposed for obtaining undominated solutions.

Subject classifications: inventory/production: (r, Q) policy, stochastic demand; programming: storage-space constraint, algorithm.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received October 2004; revisions received May 2005, April 2006, August 2006; accepted August 2006. Published online in *Articles in Advance* August 3, 2007.

1. Introduction

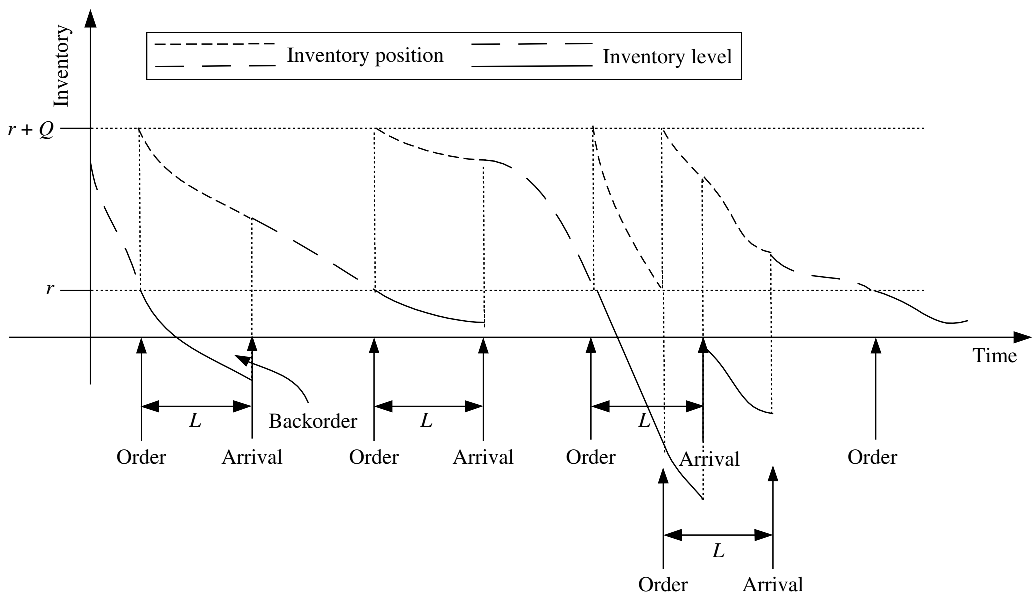
The continuous-review (r, Q) policy is popular in inventory management. The system is reviewed continuously, and whenever the inventory position drops to or below r, an amount of Q units of goods is issued to replenish the system. To be able to implement the policy, the system needs the storage space that can store the maximum inventory $r + Q$. In actual systems, usually, the resource such as the storage space is limited. (Many kinds of resources can be referred. In this paper, we refer the resource in particular to a class of storage spaces or resources like this.) Then, a major issue in planning and operations management is the optimal r and Q with capacitated storage space.

Most actual inventory systems are constructed for storing goods for multiple items. For instance, the distribution center of a chain store manages its inventory system for many items. Similar modes exist at wholesalers, third-party logistics centers, and department stores. Functionally, goods from different items exhibit individual characteristics in, for example, different weights, different shapes, and different volumes. A particular item may need distinct shelves and even special equipment for placing its goods. Therefore, each item must possess its own space to store the corresponding goods. The space to a particular item is then exclusively occupied and used by this item. Across different items, they are independent. A major issue in planning and operations management of such systems is the allocation of the storage space to the items, and the values of r and Q for each item with capacitated storage space.

The literature on inventory management is vast. Here we give a brief review on those works related to the continuous-review (r, Q) policy without a storage-space constraint. Federgruen and Zheng (1992) propose an efficient algorithm to calculate the optimal r and Q for single-item systems. In their model, the system cost consists of three terms: fixed setup cost, goods holding cost, and penalty cost for customer backorders. The objective is to minimize the long-run average system cost. Their work lays the foundation for our work in this paper. Other research on the single-item (r, Q) policy includes Hadley and Whitin (1963), Lau et al. (2002), Sahin (1979), Sivazlian (1974), Zheng (1992), and the references cited therein.

The issue we consider in this paper is important in two aspects. On the one hand, inventory systems under continuous-review (r, Q) policies (for either a single item

Figure 1. Inventories under the (r, Q) policy.



or multiple items) with a storage-space constraint widely exist in the real logistics field. On the other hand, to the authors' knowledge, the issue has not yet been dealt with in the literature. (Only a few works have dealt with resource-constraint problems for newsboy models or for periodic review order-up-to policies; see, for example, Erlebacher 2000, Lau and Lau 1996, and Hausman et al. 1998.)

This paper is organized as follows: In the next section, we provide preliminary results on the single-item (r, Q) policy without a storage-space constraint. Section 3 analyzes the single-item (r, Q) policy with a storage-space constraint and develops a procedure to obtain optimal solutions. Section 4 analyzes multi-item (r, Q) policies with a storage-space constraint and provides an algorithm for obtaining undominated solutions. Section 5 contains concluding remarks.

2. Preliminaries: Single-Item System Without Storage-Space Constraint

In this section, we introduce results related to the optimal single-item (r, Q) policy without a storage-space constraint. Based on them, storage-space constraint problems will be discussed in the subsequent sections.

2.1. System Description

Consider such a configuration as a manufacturer-inventory-customer system for only a single item. Goods are in discrete form or unit form. Consumption for goods by customers follows a renewal process and every customer demands a unit of goods. Demands that cannot be immediately fulfilled are backordered. The inventory is reviewed continuously and controlled by an (r, Q) policy. Two terminologies related to inventory are defined: inventory level and inventory position. The inventory level refers to the amount of goods on hand minus the number of backorders. (Note that the amount of goods on hand and the number of backorders cannot be simultaneously positive.) The manufacturer replenishes goods to the inventory system with a constant lead time after receiving an order from the inventory system. Then, the inventory position refers to the amount of the inventory level plus replenishments in transit.

We discuss the system in the steady state. Let L be the lead time, which is a constant. Denote by I^l the inventory level, which is a random variable, and by I^p the inventory position, which is also a random variable. Assume that the mean demand per unit time is λ. Moreover, let D represent the demand in the lead-time interval, which is a random variable with mean $\lambda \cdot L$.

The (r, Q) policy controls the inventory system in accordance with the following mechanism: whenever the inventory position I^p drops to or below r, the inventory manager issues an order to the manufacturer for an amount of Q units of goods to replenish, and the ordered goods arrive at the inventory system after a time delay L. Goods are consumed by customers. Figure 1 illustrates the operation of an inventory system controlled by an (r, Q) policy. It is known that, as shown in the figure, if no orders are currently outstanding, the inventory position I^p and the inventory level I^l are the same; otherwise their difference is just the total amount of goods ordered in outstanding at the moment. Moreover, the inventory position I^p takes values on $\{r+1, r+2, \ldots, r+Q\}$, whereas the inventory level I^l can be positive, zero, or negative (a positive value stands for the goods on hand, and a negative value refers to backorders).

Note that at any time there can be more than one order outstanding. On the other hand, the feasible values of r are any finite integers (negative, zero, or positive), whereas those of Q must be finite positive integers.

Let Ω denote the value space of (r, Q) defined by

$$\Omega = \{(X, Y) \mid -\infty < X < +\infty,\ 1 \leqslant Y < +\infty,\ X \text{ and } Y \text{ are integers}\}. \quad (1)$$

2.2. Existing Results

All results in this subsection can be found in, for example, Federgruen and Zheng (1992), Hadley and Whitin (1963), and Sivazlian (1974).

In the steady state, the following relationship holds

$$I^l = I^p - D. \quad (2)$$

LEMMA 2.1. (a) *I^p is uniformly distributed on $\{r+1, r+2, \ldots, r+Q\}$;*
(b) *D is independent of I^p.*

Assume that the cost structure consists of a setup cost K per order from the inventory system to the manufacturer, an inventory holding cost of h per unit on hand held in the inventory system per unit time, and a backorder cost of p per unit of backorders for customers per unit time. Given the inventory position being y, the expectation of the holding and backordering costs per unit time is expressed as

$$G(y) = h\sum_{i=0}^{y}(y-i)\cdot \Pr\{D=i\} + p\sum_{i=y+1}^{\infty}(i-y)\cdot \Pr\{D=i\}. \quad (3)$$

The following result holds.

LEMMA 2.2. (a) *$G(y)$ is convex with respect to y, and $-G(y)$ is a unimodal function*;
(b) $\lim_{|y|\to\infty} G(y) = \infty$.

Under an (r, Q) policy, we can obtain the expectation of the system cost per unit time as

$$c(r, Q) = \frac{K\lambda}{Q} + \frac{1}{Q}\sum_{y=r+1}^{r+Q} G(y). \quad (4)$$

An optimization problem is then formulated as follows.

PROBLEM 2.1. Determine r and Q to minimize $c(r, Q)$, i.e.,

$$\min_{(r, Q)\in\Omega} c(r, Q). \quad (5)$$

An efficient algorithm is proposed to solve the above optimization problem (Federgruen and Zheng 1992; see also Zipkin 2000, Chapter 6), which is summarized as follows.

ALGORITHM 2.1.
Step 1. Find y^* that minimizes $G(y)$.
Step 2. Set $q_{\min} = y^*$, $q_{\max} = y^*$.
Step 3. Let $r = q_{\min} - 1$, $Q = q_{\max} - q_{\min} + 1$.
Step 4. If $\min\{G(q_{\min}-1), G(q_{\max}+1)\} \geqslant c(r, Q)$, then stop. Otherwise, go to the following step.
Step 5. If $G(q_{\min}-1) \leqslant G(q_{\max}+1)$, then $q_{\min} = q_{\min} - 1$. Otherwise, $q_{\max} = q_{\max} + 1$. Go to Step 3.

Denote the final resultant policy produced by the algorithm by $(\tilde{r}, \tilde{Q}) \in \Omega$, the optimal policy of Problem 2.1. In fact, the algorithm eventually evaluates and uses $\tilde{Q}$ smallest values on the convex function $G(y)$.

2.3. New Results

Other than the above existing results in the literature, we provide the following fundamental results that will be used in the subsequent sections.

LEMMA 2.3. (a) *For any given $Q \geqslant 1$, $c(r, Q)$ is convex with respect to r.*
(b) *For any given r, $c(r, Q)$ is convex with respect to Q.*

PROOF. (a) For any given $Q \geqslant 1$, consider

$$\delta(r) = c(r+1, Q) - c(r, Q).$$

It is sufficient to prove that $\delta(r)$ is increasing with respect to r. Substituting (4) into the above leads to

$$\delta(r) = \frac{1}{Q}[G(r+1+Q) - G(r+1)].$$

Then, the convexity of $G(y)$ in Lemma 2.2 implies that $\delta(r)$ is increasing as r increases.

(b) For any given r, consider

$$\delta(Q) = c(r, Q+2) + c(r, Q) - 2c(r, Q+1).$$

Substituting (4) into the above leads to

$$\delta(Q) = \frac{1}{Q(Q+1)(Q+2)}\Bigg[2K\lambda + 2\sum_{y=r+1}^{r+Q} G(y) + Q(Q+1)G(r+Q+2) - Q(Q+3)G(r+Q+1)\Bigg].$$

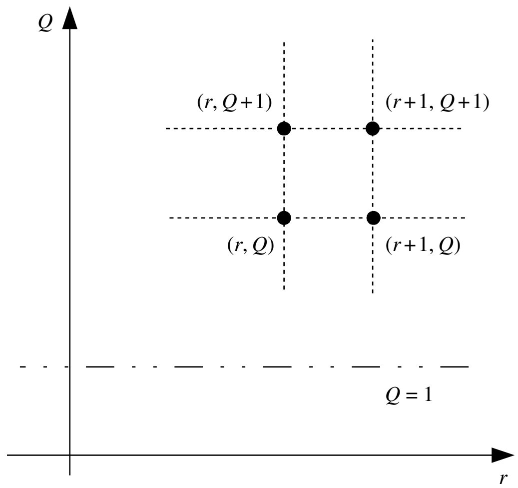
Define the following function:

$$F(Q) = 2\sum_{y=r+1}^{r+Q} G(y) + Q(Q+1)G(r+Q+2) - Q(Q+3)G(r+Q+1).$$

It can be verified that $F(1) \geqslant 0$ and $F(Q+1) - F(Q) \geqslant 0$ for all $Q \geqslant 1$. Thus, $F(Q) \geqslant 0$ for all $Q \geqslant 1$. This implies that $\delta(Q) \geqslant 0$ for all $Q \geqslant 1$. □

Consider four neighboring points (r, Q), $(r, Q+1)$, $(r+1, Q)$, and $(r+1, Q+1)$ (see Figure 2). Their relationship is stated in the following result.

Figure 2. Relationship of the system cost at the four neighboring points.



LEMMA 2.4. *For* $Q \geqslant 1$, $c(r,Q) - c(r+1,Q) \geqslant c(r,Q+1) - c(r+1,Q+1)$.

PROOF. It holds from (4) that

$$c(r,Q) - c(r+1,Q) = \frac{G(r+1) - G(r+1+Q)}{Q}.$$

It can be verified, by the convexity of $G(y)$, that for $Q \geqslant 1$,

$$\frac{G(r+1)-G(r+1+Q)}{Q} \geqslant \frac{G(r+1)-G(r+1+Q+1)}{Q+1},$$

which implies the result. □

The lemma indicates that the difference $c(r,Q) - c(r+1,Q)$ is decreasing along the Q-axis. It also indicates that the difference $c(r,Q) - c(r,Q+1)$ is decreasing along the r-axis, i.e., $c(r,Q) - c(r,Q+1) \geqslant c(r+1,Q) - c(r+1,Q+1)$.

3. Single-Item System with Storage-Space Constraint

Consider an inventory system for a single item controlled by an (r, Q) policy as described in the previous section, but the storage space is capacitated. Suppose that the amount of the storage space is w units and one unit of goods occupies one unit of the storage space. All other notation that will be used in the sequel possesses the same meaning as in the previous section.

The optimization problem is then formulated as follows.

PROBLEM 3.1. Determine r and Q to minimize $c(r,Q)$, subject to $r+Q \leqslant w$, i.e.,

$$\min_{(r,Q)\in\Omega} \quad c(r,Q), \tag{6}$$

$$\text{s.t.} \quad r+Q \leqslant w. \tag{7}$$

If the demand in the lead-time interval can be zero, i.e., $\Pr\{D=0\} > 0$, then the maximal inventory on hand can be $r+Q$. Therefore, constraint (7) corresponds to this maximal inventory on hand. The major reason for taking such a constraint is that it guarantees the implementation of the (r, Q) policy. (See also the discussions in Remarks 3.2 and 3.3 at the end of this section.)

For Problem 3.1, we can first compute an optimal policy by relaxing constraint (7) by Algorithm 2.1; call it the relaxed policy $(\tilde{r}, \tilde{Q})$. This policy requires $\tilde{w} = \tilde{r} + \tilde{Q}$ units of the storage space. Then, check whether or not constraint (7) is satisfied. If it is true, then $(\tilde{r}, \tilde{Q})$ is the optimal policy of Problem 3.1. Otherwise, we need a procedure to obtain the optimal policy. In doing so, we first show several useful structural properties of the optimization problem.

LEMMA 3.1. *For* $Q \geqslant 1$ *and any given* w, *either* $-c(r,Q)$ *is a unimodal function on the line* $r+Q=w$, *or* $c(r,Q)$ *is increasing as* Q *increases on the line* $r+Q=w$.

PROOF. Substituting $r = w-Q$ into (4), we have the system cost function in terms of Q alone as

$$c(Q) = \frac{K\lambda}{Q} + \frac{1}{Q}\sum_{y=w-Q+1}^{w} G(y). \tag{8}$$

Consider that

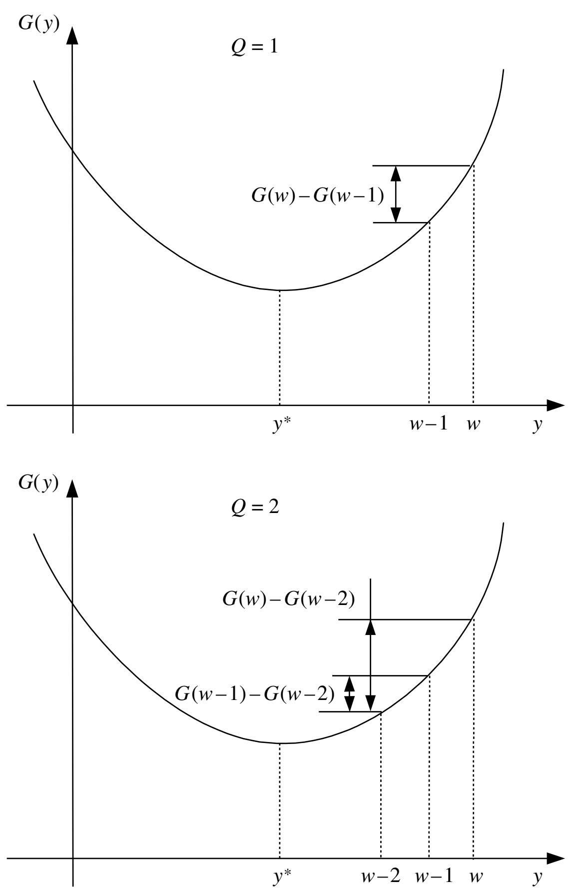
$$\begin{aligned}\delta(Q) &= c(Q) - c(Q+1)\\ &= \frac{K\lambda}{Q} + \frac{1}{Q}\sum_{y=w-Q+1}^{w} G(y) - \frac{K\lambda}{Q+1} - \frac{1}{Q+1}\sum_{y=w-Q}^{w} G(y)\\ &= \frac{1}{Q(Q+1)}\Bigg[K\lambda + \sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)]\Bigg].\end{aligned}$$

Because Q takes integer values larger than or equal to one, we can observe the behavior of $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)]$ with respect to Q. Assume that y^* is the minimizing point of the convex function $G(y)$, as shown in Figure 3 Consider two cases: $w > y^*$ and $w \leqslant y^*$ as follows.

Case 1. $w > y^*$. When $Q = 1$, $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)] = G(w) - G(w-1) > 0$. When $Q = 2$, $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)] = [G(w) - G(w-2)] + [G(w-1) - G(w-2)]$. Before $w-Q$ reaches y^*, $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)]$ is increasing as Q increases (see also Figure 3). After $w-Q$ becomes smaller than y^*, $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)]$ decreases as Q increases. Together with Lemma 2.2, we have $\sum_{y=w-\infty+1}^{w}[G(y) - G(w-\infty)] = -\infty$ due to $Q = \infty$. Taking into consideration that $0 \leqslant K\lambda < \infty$, we have that $\delta(Q)$ possesses a unique finite turning point Q^*, and $c(Q) - c(Q+1) \geqslant 0$ on the interval $[1, Q^*]$, whereas $c(Q) - c(Q+1) \leqslant 0$ on the interval $[Q^*, \infty)$.

Case 2. $w \leqslant y^*$. In this case, $\sum_{y=w-Q+1}^{w}[G(y) - G(w-Q)]$ is smaller than zero for any $Q \geqslant 1$ and is decreasing as Q increases. Taking into consideration that

Figure 3. Convex function $G(y)$.

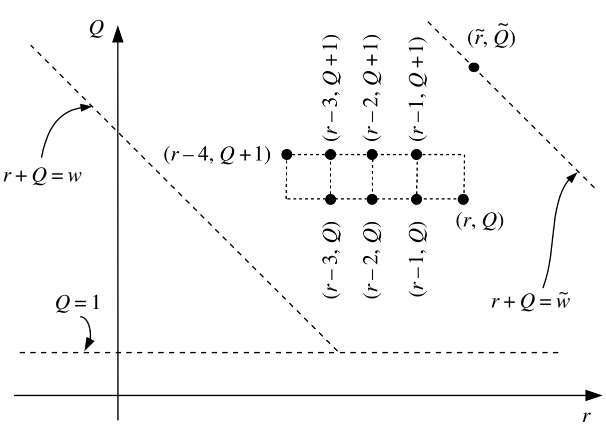


$0 \leqslant K\lambda < \infty$, Lemma 2.2 then implies that $\delta(Q)$ either possesses a unique turning point or is always negative. □

The solution space of Problem 3.1 is formed by $r + Q \leqslant w$ and $Q \geqslant 1$, as shown in Figures 4 and 5.

Definition 3.1. Point (r_1, Q_1) is superior to point (r_2, Q_2), denoted by $(r_1, Q_1) \prec (r_2, Q_2)$, if $c(r_1, Q_1) \leqslant c(r_2, Q_2)$.

Figure 4. Solution space and the superior property along the opposite direction of the r-axis.



The following result provides a superior property along the opposite direction of the r-axis. That is, if $(r, Q) \prec (r-1, Q+1)$, then $(r-1, Q) \prec (r-2, Q+1)$, $(r-2, Q) \prec (r-3, Q+1)$, $(r-3, Q) \prec (r-4, Q+1)$ and so forth (see also Figure 4).

Lemma 3.2. *For any* r *and any* $Q \geqslant 1$, *if* $(r, Q) \prec (r-1, Q+1)$, *then* $(r-n, Q) \prec (r-n-1, Q+1)$ *for all* $n \geqslant 1$.

Proof. It is sufficient to prove that if $(r, Q) \prec (r-1, Q+1)$, then $(r-1, Q) \prec (r-2, Q+1)$. Consider

$$\delta = [c(r-2, Q+1) - c(r-1, Q)] - [c(r-1, Q+1) - c(r, Q)].$$

After some algebra, we have

$$\delta = \frac{1}{Q(Q+1)}\{Q[G(r-1) - G(r)] - [G(r) - G(r+Q)]\}.$$

Then, the convexity of $G(y)$ implies that $\delta \geqslant 0$. Therefore, it holds that

$$c(r-2, Q+1) - c(r-1, Q) \geqslant c(r-1, Q+1) - c(r, Q).$$

On the other hand, by the condition $(r, Q) \prec (r-1, Q+1)$, that means $c(r-1, Q+1) - c(r, Q) \geqslant 0$, we have

$$c(r-2, Q+1) - c(r-1, Q) \geqslant 0,$$

which establishes the result. □

The following result provides a superior property along the opposite direction of the Q-axis. That is, if $(r, Q) \prec (r+1, Q-1)$, then $(r, Q-1) \prec (r+1, Q-2)$, $(r, Q-2) \prec (r+1, Q-3)$, and so forth (see also Figure 5). Recall that $(\tilde{r}, \tilde{Q})$ is the optimal solution of Problem 2.1 – an optimization problem without a storage-space constraint, which can be solved by Algorithm 2.1. Note that just $\tilde{Q}$ values of $G(y)$, i.e., $G(\tilde{r}+1), \ldots, G(y^*), \ldots, G(\tilde{r}+\tilde{Q})$, are used in calculating the optimal policy $(\tilde{r}, \tilde{Q})$.

Lemma 3.3. *For* $r + Q \leqslant \tilde{w}$ *and* $r \leqslant \tilde{r}$, *if* $(r, Q) \prec (r+1, Q-1)$, *then* $(r, Q-n) \prec (r+1, Q-n-1)$ *for all* $n \geqslant 1$.

Proof. It is sufficient to prove that if $(r, Q) \prec (r+1, Q-1)$, then $(r, Q-1) \prec (r+1, Q-2)$. According to the relationship between $G(\tilde{r}+1)$ and $G(\tilde{r}+\tilde{Q})$, we consider two different cases.

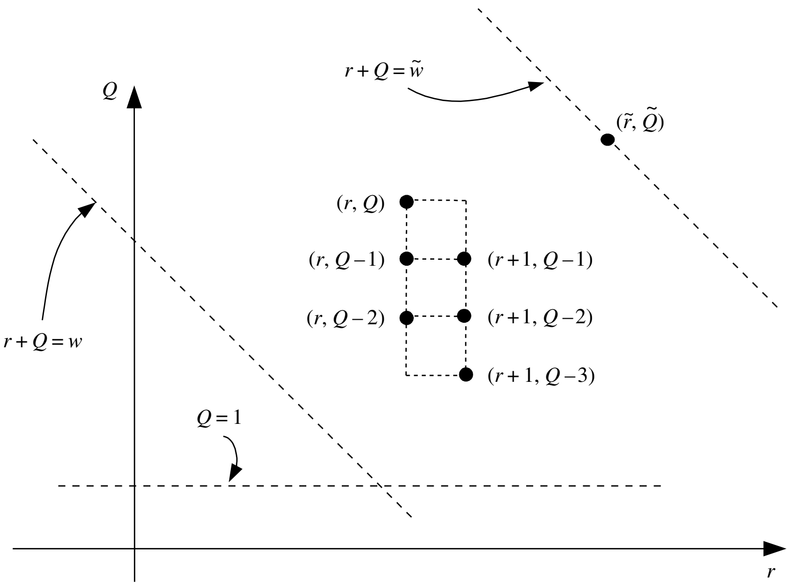
Case 1. Assume that $G(\tilde{r}+1) \geqslant G(\tilde{r}+\tilde{Q})$. Consider δ defined by

$$\delta = [c(r+1, Q-2) - c(r, Q-1)].$$

From the condition $(r, Q) \prec (r+1, Q-1)$, i.e., $c(r, Q) \leqslant c(r+1, Q-1)$, it follows that

$$K\lambda \geqslant -\sum_{y=r+2}^{r+Q} G(y) + (Q-1)G(r+1).$$

Figure 5. Solution space and the superior property along the opposite direction of the Q-axis.



Then, we have

$$\delta = \frac{K\lambda}{(Q-2)(Q-1)} + \frac{1}{(Q-2)(Q-1)} \cdot \sum_{y=r+2}^{r+Q-1} G(y) - \frac{1}{Q-1} G(r+1)$$
$$\geqslant \frac{1}{(Q-2)(Q-1)} [G(r+1) - G(r+Q)].$$

Because $\tilde{r}+1 \leqslant y^*$, the convexity of $G(y)$ indicates that $G(r+1) \geqslant G(\tilde{r}+1)$ for $r \leqslant \tilde{r}$. On the other hand, it is clear that either $r+1 \leqslant y^* \leqslant r+Q \leqslant \tilde{w}$ or $r+1 \leqslant r+Q \leqslant y^* \leqslant \tilde{w}$. Then, the condition $G(\tilde{r}+1) \geqslant G(\tilde{r}+\tilde{Q})$ together with the convexity of $G(y)$ implies that $G(r+1) \geqslant G(r+Q)$ and thus $\delta \geqslant 0$. Therefore, it holds that

$$c(r+1, Q-2) \geqslant c(r, Q-1).$$

Case 2. Assume that $G(\tilde{r}+1) \leqslant G(\tilde{r}+\tilde{Q})$. In this case, it must hold that $G(\tilde{r}) \geqslant G(\tilde{r}+\tilde{Q})$. (Recall that Algorithm 2.1 generates an optimal solution $(\tilde{r}, \tilde{Q})$, which implies that $\tilde{Q}$ smallest values on the convex function $G(y)$ are $G(\tilde{r}+1), G(\tilde{r}+2), \ldots, G(\tilde{r}+\tilde{Q})$. Therefore, $G(\tilde{r}) \geqslant \max\{G(\tilde{r}+1), \ldots, G(\tilde{r}+\tilde{Q})\} \geqslant G(\tilde{r}+\tilde{Q})$.) Under the hypothesis $G(\tilde{r}) \geqslant G(\tilde{r}+\tilde{Q})$, the result can be proved for $r \leqslant \tilde{r}-1$ by the same method as in Case 1. Now we discuss the situation $r = \tilde{r}$. Consider

$$\delta = c(\tilde{r}+1, Q-2) - c(\tilde{r}, Q-1)$$
$$= \frac{1}{Q-2} [c(\tilde{r}, Q-1) - G(\tilde{r}+1)].$$

On the other hand, Algorithm 2.1 generates the optimal solution $(\tilde{r}, \tilde{Q})$ to Problem 2.1, which implies that $c(\tilde{r}, \tilde{Q}-1) > G(\tilde{r}+\tilde{Q})$ (and the algorithm is stopped due to $c(\tilde{r}, \tilde{Q}) \leqslant G(\tilde{r})$ or $c(\tilde{r}, \tilde{Q}) \leqslant G(\tilde{r}+\tilde{Q}+1)$). Then, it is clear that

$$c(\tilde{r}, Q-1) > c(\tilde{r}, \tilde{Q}-1) > G(\tilde{r}+\tilde{Q}) \geqslant G(\tilde{r}+1).$$

Therefore, it holds that

$$c(\tilde{r}+1, Q-2) - c(\tilde{r}, Q-1) \geqslant 0,$$

and the lemma is proved. □

The following results provide monotone properties of $c(r, Q)$ along the opposite directions of the r-axis and the Q-axis.

Lemma 3.4. (a) *For $r \leqslant \tilde{r}$ and $Q \leqslant \tilde{Q}$, $c(r, Q)$ is nondecreasing as r decreases for a given Q.*

(b) *For $r \leqslant \tilde{r}$ and $Q \leqslant \tilde{Q}$, $c(r, Q)$ is nondecreasing as Q decreases for a given r.*

Proof. (a) It follows from Lemma 2.4 that if $(r+1, Q+1) \prec (r, Q+1)$, then $(r+1, Q) \prec (r, Q)$. Lemma 2.3 indicates that

$$(\tilde{r}, \tilde{Q}) \prec (\tilde{r}-1, \tilde{Q}) \prec (\tilde{r}-2, \tilde{Q}) \prec \cdots,$$

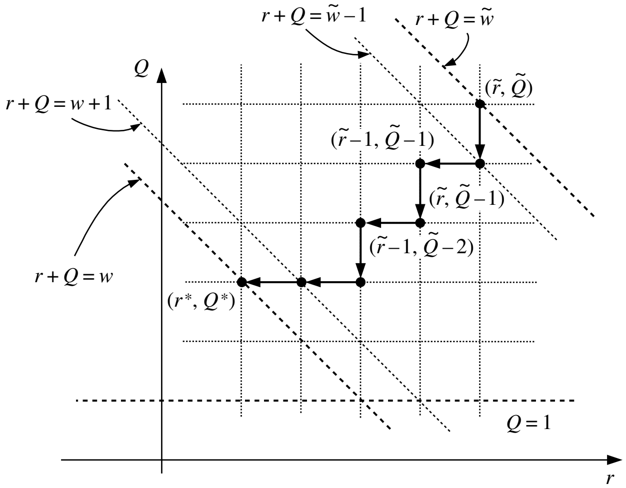
and

$$(\tilde{r}, \tilde{Q}) \prec (\tilde{r}, \tilde{Q}-1) \prec (\tilde{r}, \tilde{Q}-2) \prec \cdots.$$

Using the above relationships repeatedly, it holds that $(\tilde{r}-n, \tilde{Q}) \prec (\tilde{r}-n-1, \tilde{Q})$ for any $n \geqslant 0$, and furthermore we have $(\tilde{r}-n, \tilde{Q}-k) \prec (\tilde{r}-n-1, \tilde{Q}-k)$ for any $0 \leqslant k \leqslant \tilde{Q}-1$.

(b) The proof can be established similar to the above. □

Figure 6. Resultant path by the algorithm.



Now, we are ready to propose our algorithm for solving Problem 3.1 when constraint (7) is active. The principle is as follows (see also Figure 6). First, utilize Algorithm 2.1 to find the relaxed policy $(\tilde{r}, \tilde{Q})$ by relaxing constraint (7). Then, starting from point $(\tilde{r}, \tilde{Q})$, move toward the line $r+Q=w$ step by step. At the first step, consider two points $(\tilde{r}-1, \tilde{Q})$ and $(\tilde{r}, \tilde{Q}-1)$, which are neighbors of point $(\tilde{r}, \tilde{Q})$ and are on the line $r+Q=\tilde{w}-1$. If $(\tilde{r}, \tilde{Q}-1) \prec (\tilde{r}-1, \tilde{Q})$, then move to point $(\tilde{r}, \tilde{Q}-1)$. Based on this point, compare its two neighboring points $(\tilde{r}-1, \tilde{Q}-1)$ and $(\tilde{r}, \tilde{Q}-2)$, which are on the line $r+Q=\tilde{w}-2$. If $(\tilde{r}-1, \tilde{Q}-1) \prec (\tilde{r}, \tilde{Q}-2)$, then move to point $(\tilde{r}-1, \tilde{Q}-1)$. The procedure proceeds until the line $r+Q=w$ is reached. Formally, it is summarized as the following algorithm.

Algorithm 3.1.

Step 1. Find the relaxed policy $(\tilde{r}, \tilde{Q})$ by Algorithm 2.1.
Step 2. Let $r^* = \tilde{r}$ and $Q^* = \tilde{Q}$. Set $N = \tilde{w} - w$.
Step 3. If $N \leqslant 0$, then stop.
Step 4. If $c(r^*-1, Q^*) \leqslant c(r^*, Q^*-1)$, then let $r^* = r^* - 1$. Otherwise, let $Q^* = Q^* - 1$.
Step 5. $N = N - 1$. Go to Step 3.

Denote by $\vec{P}$ the directed path from point $(\tilde{r}, \tilde{Q})$ to point (r^*, Q^*) generated by the algorithm. Lemma 3.4 straightforwardly leads to the following result.

Lemma 3.5. *$c(r, Q)$ is increasing along the path $\vec{P}$.*

For any given w $(\leqslant \tilde{w})$, the algorithm generates the directed path by $N = \tilde{w} - w$ steps. The optimality of the resultant policy (r^*, Q^*) is given by the following proposition.

Proposition 3.1. *For any given w $(\leqslant \tilde{w})$, the algorithm generates an optimal policy (r^*, Q^*) to Problem* 3.1.

Proof. The result holds obviously for $w = \tilde{w}$. Consider $w = \tilde{w} - 1$. The unimodal property on the line $r+Q=\tilde{w}$ in Lemma 3.1 implies that $(\tilde{r}, \tilde{Q}) \prec (\tilde{r}-1, \tilde{Q}+1)$ and $(\tilde{r}, \tilde{Q}) \prec (\tilde{r}+1, \tilde{Q}-1)$. Starting from point $(\tilde{r}, \tilde{Q})$, the algorithm reaches one of two points $(\tilde{r}-1, \tilde{Q})$ and $(\tilde{r}, \tilde{Q}-1)$ that are on the line $r+Q=\tilde{w}-1$. The unimodal property or the increasing property on the line $r+Q=\tilde{w}$ in Lemma 3.1, together with Lemmas 3.2 and 3.3, implies that the resultant point (r^*, Q^*) is superior to all other points on the line $r+Q=\tilde{w}-1$. From Lemma 3.5, it is known that the optimal solution must be on the line $r+Q=\tilde{w}-1$. Therefore, the resultant (r^*, Q^*) is the optimal policy to Problem 3.1. For $w \leqslant \tilde{w}-2$, the conclusion can be easily proved by the induction method, and we omit its detail here. □

In fact, for any given w $(\leqslant \tilde{w})$, the solution space is formed by $r+Q \leqslant w$ and $Q \geqslant 1$. The algorithm reaches the point with the minimum system cost on the line $r+Q=w$. Lemma 3.5 implies that we do not need to consider further any point in $r+Q<w$. In other words, the optimal point must be on the line $r+Q=w$. Moreover, from Lemmas 3.1, 3.2, and 3.3, we know that $r^* \leqslant \tilde{r}$ and $Q^* \leqslant \tilde{Q}$.

Proposition 3.1 implies that the path $\vec{P}$ is formed by a set of "optimal points." That is, if the amount of the storage space is $\tilde{w}$, then the first point on the path $\vec{P}$ is the optimal solution; if the amount of the storage space is $\tilde{w}-1$, then the second point on the path is optimal solution; and if the amount is $\tilde{w}-2$, then the third point on the path is the optimal solution. We shall call path $\vec{P}$ the *optimal path* hereafter.

It is clear that the calculations for obtaining the optimal solution by the algorithm cause a polynomial time computational complexity with respect to $\tilde{r}$, $\tilde{Q}$, r^*, and Q^*.

Example 3.1. Consider a problem with the storage space $w = 31$. The consumption of goods by customers follows a Poisson stream. Other parameters are listed in Table 1.

Using Algorithm 3.1, we obtain the optimal solution of the example $(r^*, Q^*) = (9, 22)$ and the corresponding system cost $c(9, 22) = 856.756$. In contrast, we provide the relaxed policy and the corresponding system cost, which are $(\tilde{r}, \tilde{Q}) = (11, 48)$ and $c(11, 48) = 608.133$, respectively.

Remark 3.1. Problem 3.1 is formulated with the assumption such that one unit of goods occupies one unit of the storage space. In general, we can assume that one unit of goods occupies s units of the storage space; s can be any real number. Then, the problem becomes

$$\min_{(r, Q) \in \Omega} \quad c(r, Q),$$
$$\text{s.t.} \quad s \cdot (r+Q) \leqslant w.$$

Table 1. Parameters of the example (single item).

λ	K	L	h	p
13	1,042	1	13	247

It is clear that the above problem is equivalent to the following:

$$\min_{(r,Q)\in\Omega} c(r,Q),$$

$$\text{s. t. } r+Q \leqslant \left\lfloor \frac{w}{s} \right\rfloor,$$

where $\lfloor x \rfloor$ represents the largest integer less than or equal to x. Therefore, provided that we replace w by $\lfloor w/s \rfloor$, Algorithm 3.1 can obtain optimal solutions to the above generalized optimization problem.

REMARK 3.2. If a system satisfies that $\Pr\{D=d\}=0$ for $d=0,1,\ldots,k-1$ and $\Pr\{D=k\}>0$, then the demand in the lead-time interval is at least k. In such a case, the inventory on hand is at most $r+Q-k$. The constraint becomes $r+Q\leqslant w+k$, but this is the same form as (7) by regarding $w+k$ as a new w. Thus, constraint (7) does not lose the generality.

REMARK 3.3. More generally, we can find the largest value k ($\geqslant 0$) that satisfies $\Pr\{D\geqslant k\}=1$. The storage space can be treated as $w+k$ to determine the optimal policy (r,Q). Theoretically, the inventory on hand is at most $r+Q-k$. If the probability that the inventory on hand is at this maximal value is very small, then the ratio of the utilization of the storage space may be low. Such a situation may be improved to some extent. Suppose that v ($\geqslant 0$) is the largest value satisfying $\Pr\{D\geqslant v\}\geqslant \alpha$ for a given α ($\leqslant 1$). It is obvious that $v=k$ if $\alpha=1$ and $v\geqslant k$ if $\alpha\leqslant 1$. If we treat the storage space as $w'=w+v$ and then determine the corresponding optimal policy (r',Q'), a shortage of the storage space exists to implement the policy. In other words, the risk exists such that goods ordered in outstanding cannot enter the storage when they arrive at the inventory system due to the shortage of the storage space. We call α a safety coefficient that states "goods ordered in outstanding can safely enter the storage." Then, $\Pr\{D\geqslant v\}\geqslant\alpha$ implies that

$$\Pr\{\text{goods ordered in outstanding can safely enter the storage}\}\geqslant\alpha.$$

If α is taken very close to one, the policy can be implemented safely enough while the ratio of the utilization of the storage space may be raised and the operating cost may be reduced. For Example 3.1, if we take $\alpha=0.999$, then $v=3$ and $w'=w+v=34$, that is about 9.7% larger than the actual storage space $w=31$. The resultant policy (r',Q') is $(10,24)$ with the system cost $c(10,24)=783.071$, that is about 8.6% reduced than that in the example. If we implement this policy under the actual storage space 31, the probability that goods ordered in outstanding can safely enter the storage will be larger than 0.999.

REMARK 3.4. There may exist other approaches to solve Problem 3.1. For example, we can assign a very large penalty for the maximal inventory on hand $r+Q$ exceeding the amount of the storage space w. This penalty can be taken as a convex function and be added to the system cost. Through such a transformation of a constrained problem into an unconstrained problem, Algorithm 2.1 can be applied to solve the problem. Nevertheless, the approach developed in this section provides a feasibility to solve multi-item systems defined in the next section.

4. Multi-Item System with Storage-Space Constraint

We now consider a multi-item inventory system with capacitated storage space. Customers' demands are stochastic with distributions that can be different from item to item, and they are independent across different items. Assume that the number of items is M. The amount of the storage space is W units, which are allocated to the items. For $m=1,\ldots,M$, assume that one unit of goods in item m occupies s_m units of the storage space.

Other notation in this section possesses the same meaning as in the previous sections, but each one may be attached a subscript m as the index of item m.

Let $\mathbf{s}=(s_1,\ldots,s_M)$, $\mathbf{r}=(r_1,\ldots,r_M)'$, and $\mathbf{Q}=(Q_1,\ldots,Q_M)'$. For a vector $\mathbf{x}=(x_1,\ldots,x_M)'$, denote $\mathbf{x}^+=(x_1^+,\ldots,x_M^+)'$, where $x_m^+=\max\{0,x_m\}$. For item m $(=1,\ldots,M)$, w_m units of the storage space are allocated to it and used exclusively by it. Note that w_m $(=s_m\cdot(r_m+Q_m)^+)$ is known only after the decision variables r_m and Q_m are determined.

An optimization problem is then formulated as follows.

PROBLEM 4.1. For $m=1,\ldots,M$, determine r_m and Q_m to minimize $C(\mathbf{r},\mathbf{Q})=\sum_{m=1}^{M}c_m(r_m,Q_m)$, subject to $\sum_{m=1}^{M}w_m=\sum_{m=1}^{M}s_m\cdot(r_m+Q_m)^+=\mathbf{s}\cdot(\mathbf{r}+\mathbf{Q})^+\leqslant W$, i.e.,

$$\min_{(r_m,Q_m)\in\Omega} C(\mathbf{r},\mathbf{Q}), \tag{9}$$

$$\text{s.t. } \mathbf{s}\cdot(\mathbf{r}+\mathbf{Q})^+\leqslant W. \tag{10}$$

The solution of the problem is denoted by $(\mathbf{r},\mathbf{Q})$, which is called the *system policy*.

The above optimization problem is different from standard resource allocation problems in the literature (see, for example, Ibaraki and Katoh 1988). Usually, for standard resource-allocation problems, the objective function is formed by a set of functions that are separable, each of which consists of a single decision variable. Whereas, in our problem, two decision variables, r_m and Q_m, are included in a function $c_m(r_m,Q_m)$. It is obvious, by referring to Ibaraki and Katoh (1988), that the problem is NP-hard. Thus, no existing polynomial algorithms can be applied to the optimization problem for finding the optimal solutions. Nevertheless, we can apply the principle of solving a standard resource-allocation problem to find an undominated solution for our problem.

Referring to Fox (1966), Kao (1976), and Yuceer (1998), an undominated solution is defined as follows.

Definition 4.1. A solution $(\mathbf{r}^*, \mathbf{Q}^*)$ is an undominated solution if for all $(\mathbf{r}, \mathbf{Q})$ with $(r_m, Q_m) \in \Omega$ $(m = 1, \ldots, M)$,

$$C(\mathbf{r}, \mathbf{Q}) < C(\mathbf{r}^*, \mathbf{Q}^*) \Rightarrow \mathbf{s} \cdot (\mathbf{r} + \mathbf{Q})^+ > \mathbf{s} \cdot (\mathbf{r}^* + \mathbf{Q}^*)^+,$$
$$C(\mathbf{r}, \mathbf{Q}) = C(\mathbf{r}^*, \mathbf{Q}^*) \Rightarrow \mathbf{s} \cdot (\mathbf{r} + \mathbf{Q})^+ \geqslant \mathbf{s} \cdot (\mathbf{r}^* + \mathbf{Q}^*)^+.$$

The following result is a key property for developing an algorithm to solve Problem 4.1.

Proposition 4.1. *For an item m and any given* w_m $(\leqslant \tilde{w}_m)$, $c_m(r_m, Q_m)$ *is increasing and convex along the optimal path* $\vec{P}_m$.

Proof. Algorithm 3.1 generates the optimal path $\vec{P}_m$ starting from point $(\tilde{r}_m, \tilde{Q}_m)$ and ending at point (r_m^*, Q_m^*). From Lemma 3.5, it straightforwardly follows that $c_m(r_m, Q_m)$ is increasing along the optimal path. On the other hand, the optimal path is formed by $N_m + 1$ points, where $N_m = \tilde{w}_m - w_m$. We index these points as $0, 1, \ldots, N_m$, and the (r_m, Q_m) at point n is denoted by (r_{m_n}, Q_{m_n}). Then, for the convexity to hold, it is sufficient to prove that for any three consecutive points n, $n+1$, $n+2$,

$$c_m(r_{m_{n+2}}, Q_{m_{n+2}}) - c_m(r_{m_{n+1}}, Q_{m_{n+1}}) \geqslant c_m(r_{m_{n+1}}, Q_{m_{n+1}}) - c_m(r_{m_n}, Q_{m_n}).$$

Consider any three consecutive points on the optimal path. If they are all on a horizontal line or all on a vertical line, Lemma 2.3 indicates that $c_m(r_m, Q_m)$ is convex at these three points. If the three consecutive points are neither all on a horizontal line nor all on a vertical line, they can be in two forms as shown in Figure 7. For the form in Figure 7(a), it is sufficient to prove that

$$c_m(r_m - 1, Q_m - 1) - c_m(r_m - 1, Q_m) \geqslant c_m(r_m - 1, Q_m) - c_m(r_m, Q_m).$$

Note that the form in Figure 7(a) implies that $(r_m - 1, Q_m) \prec (r_m, Q_m - 1)$. Therefore, it holds that

$$c_m(r_m - 1, Q_m - 1) - c_m(r_m - 1, Q_m) \geqslant c_m(r_m - 1, Q_m - 1) - c_m(r_m, Q_m - 1).$$

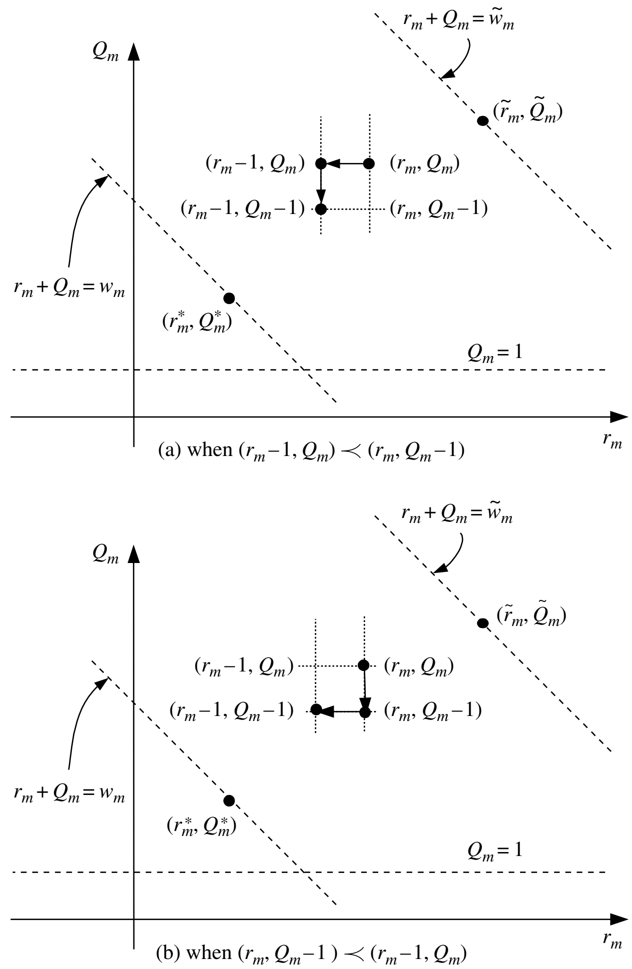
Lemma 2.4 indicates that

$$c_m(r_m - 1, Q_m - 1) - c_m(r_m, Q_m - 1) \geqslant c_m(r_m - 1, Q_m) - c_m(r_m, Q_m).$$

Thus, the result follows. For the other form as shown in Figure 7(b), the result can be proved similarly. □

Because the objective function (9) is a separable function, a marginal procedure can be devised based on Proposition 4.1. (A marginal procedure is introduced in, for example, Fox 1966, Kao 1976, and Ibaraki and Katoh 1988.) Denote by $\vec{P}_m(n_m)$ the optimal path with $n_m + 1$ points starting from point $(\tilde{r}_m, \tilde{Q}_m)$ and ending at point (r_m, Q_m) generated by Algorithm 3.1 for item m. Let $c_m(\vec{P}_m(n_m))$ represent the cost of item m at the end point (r_m, Q_m) of the optimal path $\vec{P}_m(n_m)$. Let $\widetilde{W} = \sum_{m=1}^{M} \tilde{w}_m = \sum_{m=1}^{M} s_m \cdot (\tilde{r}_m + \tilde{Q}_m)^+$. An allocation scheme is proposed in the following algorithm, where a contributive item m means that its current r_m and Q_m satisfy $r_m + Q_m > 0$. (See also the discussions in Remark 4.2 at the end of this section.)

Figure 7. Convexity of the systems cost along the optimal path $\vec{P}_m$.



Algorithm 4.1.

Step 1. Find the relaxed system policy, i.e., $(\tilde{r}_m, \tilde{Q}_m)$ for all $m = 1, \ldots, M$ by Algorithm 2.1.

Step 2. Set $n_m = 0$ and $(\tilde{r}_m, \tilde{Q}_m) \rightarrow \vec{P}_m(0)$ for all $m = 1, \ldots, M$. Set $N = \widetilde{W} - W$.

Step 3. If $N \leqslant 0$, then stop.

Step 4. Use Algorithm 3.1 to find the index m among contributive items with the smallest value of

$$\frac{c_m(\vec{P}_m(n_m + 1)) - c_m(\vec{P}_m(n_m))}{s_m}.$$

Then, let $n_m = n_m + 1$ for the above mentioned m.

Step 5. $N = N - s_m$. Go to Step 3.

Let (r_m^*, Q_m^*) be the end point corresponding to the final resultant optimal path $\vec{P}_m(n_m)$ for $m = 1, \ldots, M$. With Proposition 4.1, the following result is obvious and thus we omit its proof. (The principle of the proof is similar to, for example, Fox 1966 and Kao 1976.)

PROPOSITION 4.2. *For any given W $(\leqslant \widetilde{W})$, the algorithm generates an undominated solution $(\mathbf{r}^*, \mathbf{Q}^*)$ formed by policies (r_m^*, Q_m^*) for all $m = 1, \ldots, M$ to Problem* 4.1.

The key for the above proposition to hold is "along the optimal path $\vec{P}_m$" described in Proposition 4.1, not along all increasing and convex paths.

It is clear that the calculations for obtaining the undominated solution by the algorithm cause a polynomial time computational complexity with respect to $\tilde{r}_m$, $\widetilde{Q}_m$, r_m^*, and Q_m^* over $m = 1, \ldots, M$.

The following result is straightforward.

COROLLARY 4.1. *If Algorithm* 4.1 *is terminated with the storage space to be used up, then the resultant solution is an optimal solution of Problem* 4.1.

One of the examples for the storage space to be used up by the algorithm is such that $s_m = 1$ for all $m = 1, 2, \ldots, M$. By the corollary, the algorithm always generates optimal solutions for such cases.

Although the undominated solution obtained by the algorithm may not be an optimal solution, we can easily evaluate the quality of the undominated solution. Recall that the algorithm starts from the initial solution $(\tilde{\mathbf{r}}, \tilde{\mathbf{Q}})$ and ends at the final solution $(\mathbf{r}^*, \mathbf{Q}^*)$. Suppose that $(\tilde{\mathbf{r}}, \tilde{\mathbf{Q}})$, $(\mathbf{r}^1, \mathbf{Q}^1), \ldots, (\mathbf{r}^n, \mathbf{Q}^n), (\mathbf{r}^*, \mathbf{Q}^*)$ is the solution sequence step-by-step produced by the algorithm. Denote by $(\bar{\mathbf{r}}, \bar{\mathbf{Q}})$ the optimal solution of the optimization problem.

Similar to Fox (1966), we have the following lemma:

LEMMA 4.1. *It holds that*

$$C(\mathbf{r}^n, \mathbf{Q}^n) \leqslant C(\bar{\mathbf{r}}, \bar{\mathbf{Q}}) \leqslant C(\mathbf{r}^*, \mathbf{Q}^*). \tag{11}$$

Define a relative error by

$$\Delta^* = \frac{C(\mathbf{r}^*, \mathbf{Q}^*) - C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})}{C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})}. \tag{12}$$

Then, we can give the following bound for evaluating the quality of the undominated solution.

COROLLARY 4.2. *It holds that*

$$\Delta^* \leqslant \frac{C(\mathbf{r}^*, \mathbf{Q}^*) - C(\mathbf{r}^n, \mathbf{Q}^n)}{C(\mathbf{r}^n, \mathbf{Q}^n)}. \tag{13}$$

More valuable information can be revealed. For example, we can give an estimation for the system cost at the optimal solution $C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})$ and provide quality evaluation for the estimation. Formally, consider the midvalue between $C(\mathbf{r}^n, \mathbf{Q}^n)$ and $C(\mathbf{r}^*, \mathbf{Q}^*)$ as the estimation for $C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})$, i.e.,

$$\bar{C} = \frac{1}{2}[C(\mathbf{r}^*, \mathbf{Q}^*) + C(\mathbf{r}^n, \mathbf{Q}^n)]. \tag{14}$$

Table 2. Parameters of the example (multiple items).

Item number m	s_m	λ_m	K_m	L_m	h_m	p_m
1	5.4	30	120	3	6	70
2	3.4	40	100	2	6	63
3	5.5	40	110	3	9	87
4	5.2	50	120	2	8	75
5	4.3	50	110	2	6	67
6	5.7	40	130	2	9	92
7	5.4	30	130	3	6	73
8	10.1	70	130	3	9	89
9	2.4	30	120	3	5	55
10	5.2	40	120	2	8	80
11	5.8	50	120	2	7	74
12	4.2	50	120	2	8	92
13	6.6	40	100	3	9	83
14	0.7	50	110	3	6	60
15	8.3	60	110	2	8	92
16	3.5	40	100	3	7	73
17	4.8	60	100	3	8	77
18	4.3	50	100	2	7	65
19	2.6	50	110	2	5	49
20	5.3	40	120	2	8	80
21	3.7	60	110	3	5	47
22	4.3	60	130	2	8	77
23	6.6	60	130	2	9	82
24	3.7	30	120	3	5	46
25	2.3	50	130	3	6	60
26	2.9	60	100	2	8	80
27	4.5	70	110	3	6	56
28	5.3	60	140	2	7	69
29	3.7	50	130	2	5	55
30	2.4	50	100	2	7	69

Define a relative error as

$$\bar{\Delta} = \frac{|\bar{C} - C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})|}{C(\bar{\mathbf{r}}, \bar{\mathbf{Q}})}. \tag{15}$$

Then, the following is the quality evaluation for the estimation.

COROLLARY 4.3. *It holds that*

$$\bar{\Delta} \leqslant \frac{(1/2)[C(\mathbf{r}^*, \mathbf{Q}^*) - C(\mathbf{r}^n, \mathbf{Q}^n)]}{C(\mathbf{r}^n, \mathbf{Q}^n)}. \tag{16}$$

EXAMPLE 4.1. Consider a problem relatively close to scales in actual systems. The number of items is 30, i.e., $M = 30$. The amount of the storage space $W = 16{,}000$. The consumptions of goods by customers follow Poisson streams. The parameters of the example are summarized in Table 2.

Using Algorithm 4.1, we obtain a undominated solution of the example and the corresponding system cost, which are shown in Table 3. After the calculation of the first step in Algorithm 4.1, it is known that the relaxed system policy needs the storage space $\widetilde{W} = \sum_{m=1}^{30} \tilde{w}_m = \sum_{m=1}^{30} s_m \cdot (\tilde{r}_m + \widetilde{Q}_m)^+ = 23{,}043.20$. Thus, $N = \widetilde{W} - W = 7{,}043.20$. On the other hand, the total amount of the storage space used by the undominated solution is $\sum_{m=1}^{30} s_m \cdot$

Table 3. Computational results of the example (multiple items).

Item number m	(r_m^*, Q_m^*)	$c_m(r_m^*, Q_m^*)$
1	(78, 12)	824.23
2	(71, 15)	569.91
3	(110, 13)	912.13
4	(88, 15)	917.32
5	(89, 16)	748.32
6	(71, 13)	880.89
7	(79, 12)	823.07
8	(102, 14)	9,594.50
9	(83, 16)	443.26
10	(70, 13)	832.37
11	(86, 14)	1,053.84
12	(93, 16)	756.49
13	(107, 11)	1,114.53
14	(148, 30)	365.39
15	(103, 13)	1,548.12
16	(113, 15)	611.36
17	(168, 16)	990.23
18	(88, 15)	781.85
19	(89, 19)	558.39
20	(70, 13)	832.37
21	(161, 19)	891.72
22	(109, 18)	899.20
23	(104, 16)	1,273.27
24	(76, 14)	638.99
25	(142, 22)	577.24
26	(114, 19)	626.45
27	(190, 19)	1,113.76
28	(104, 18)	1,085.61
29	(87, 18)	730.69
30	(94, 18)	528.84
Total	$\sum_{m=1}^{30} s_m \cdot (r_m^* + Q_m^*)^+ = 15{,}996.2$	$C(\mathbf{r}^*, \mathbf{Q}^*) = \sum_{m=1}^{30} c_m(r_m^*, Q_m^*) = 33{,}524.34$

$(r_m^* + Q_m^*)^+ = 15{,}996.2 < 16{,}000$. Therefore, from Corollary 4.1, the obtained undominated solution may not be an optimal solution. Nevertheless, we can provide the bound of the relative error according to Corollary 4.2. At the step just before the termination of the algorithm, the system cost is $C(\mathbf{r}^n, \mathbf{Q}^n) = 33{,}435.34$ and the storage space used is $\sum_{m=1}^{30} s_m \cdot (r_m^n + Q_m^n)^+ = 16{,}006.3$. The bound of the relative error is then

$$\Delta^* \leqslant \frac{33{,}524.34 - 33{,}435.34}{33{,}435.34} = 0.27\%,$$

which shows a very high-quality solution to be generated by the algorithm. Moreover, we can take

$$\bar{C} = \frac{1}{2}[C(r^*, Q^*) + C(r^n, Q^n)] = \frac{33{,}524.34 + 33{,}435.34}{2} = 33{,}479.84$$

as an estimation of the system cost at the optimal solution. Then, from Corollary 4.3, the quality evaluation for the above estimation is

$$\bar{\Delta} \leqslant \frac{1}{2} \cdot \frac{33{,}524.34 - 33{,}435.34}{33{,}435.34} = 0.14\%.$$

Remark 4.1. Similar to Remark 3.3 for the single-item case. If the inventory on hand rarely reaches its theoretical maximum, the ratio of the utilization of the storage space may be low. For such a case, we can define a safety coefficient α_m ($\leqslant 1$) for item m. Let v_m ($\geqslant 0$) be the largest value that satisfies $\Pr\{D_m \geqslant v_m\} \geqslant \alpha_m$. Denote $u_m = \min\{v_m, \tilde{w}_m\}$. The storage space can then be enlarged to $W' = W + \sum_{m=1}^{M} s_m \cdot u_m$. Suppose that the algorithm produces a system policy with (r_m', Q_m') to item m. Then, the actual allocation of the storage space to item m is $w_m = s_m \cdot (r_m' + Q_m') - s_m \cdot u_m$. If we implement the policy (r_m', Q_m') under the actual storage space w_m for item m, the probability that goods ordered in outstanding can safely enter the storage space will be larger than or equal to α_m.

Remark 4.2. Eventually, when constraint (10) is active, every time Algorithm 4.1 finds an item m and moves one step along its optimal path $\vec{P}_m$, that results in the smallest increase of the system cost. Moving one step along the optimal path $\vec{P}_m$ means that the corresponding r_m or Q_m is reduced so as to reduce the occupancy for the storage space by the item. Once the current r_m and Q_m reach $r_m + Q_m \leqslant 0$, which indicates that the item cannot contribute to reduce the occupancy for the storage space by further reducing its r_m or Q_m, Step 4 in Algorithm 4.1 will not consider this item (or such items) anymore. We refer the remaining items to contributive items. For the case of an enlarged W' as described in Remark 4.1, a contributive item m means that its current r_m and Q_m satisfy $r_m + Q_m > v_m$.

5. Conclusions and Future Directions

The storage space is one of typical classes of resources in actual inventory systems. For single-item systems, with polynomial time computational complexity, the algorithm can solve the optimization problem and easily obtain optimal solutions for large-scale systems. For multi-item systems, each item is allocated a fixed amount of storage space that is exclusively used by the item for placing its goods. Such an operating mode widely exists in the goods distribution field (for example, the distribution center of a chain store, wholesalers, third-party logistics centers, and department stores). The algorithm, with polynomial time computational complexity, can solve the optimization problem for multi-item systems and obtain undominated solutions for large-scale systems. The quality of the obtained undominated solutions can be easily evaluated.

Another class of resources in inventory systems is, for example, capitals and investments, which is different from the resource considered in this paper. In such cases, the resource can be commonly shared across different items, rather than exclusively used in part by a specified item. Then, an optimization problem for multi-item (r, Q) policies with a constraint of such a resource becomes more complex, and will be a direction for future research.

Acknowledgments

This research was supported by the Natural Science Foundation of China under grant 70325004 and was partially supported by grant 70532004.

References

Erlebacher, S. J. 2000. Optimal and heuristic solutions for the multiple newsvendor problem with a single capacity constraint. *Production Oper. Management* **9** 303–318.

Federgruen, A., Y. Zheng. 1992. An efficient algorithm for computing an optimal (r, Q) policy in continuous review stochastic inventory systems. *Oper. Res.* **40** 808–813.

Fox, B. 1966. Discrete optimization via marginal analysis. *Management Sci.* **13** 210–216.

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

Hausman, W. H., H. L. Lee, A. X. Zhang. 1998. Joint demand fulfillment probability in a multi-item inventory system with independent order-up-to policies. *Eur. J. Oper. Res.* **109** 646–659.

Ibaraki, T., N. Katoh. 1988. *Resource Allocation Problems*. MIT Press, Cambridge, MA.

Kao, E. P. 1976. On incremental analysis in resource allocations. *Oper. Res. Quart.* **27** 759–763.

Lau, H., A. H. Lau. 1996. The newsstand problem: A capacitated multiple-product single-period inventory problem. *Eur. J. Oper. Res.* **94** 29–42.

Lau, A. H.-L., H.-S. Lau, L. W. Robinson. 2002. Convenient expressions for computing the exact annual cost of a continuous-review (Q, R) system with backordering. *J. Oper. Res. Soc.* **53** 655–663.

Sahin, I. 1979. On the stationary analysis of continuous review (s, S) inventory system with constant lead times. *Oper. Res.* **27** 717–728.

Sivazlian, B. D. 1974. A continuous-review (s, S) inventory system with arbitrary interarrival distribution between unit demand. *Oper. Res.* **22** 65–71.

Yuceer, U. 1998. Marginal allocation algorithm for nonseparable functions. *INFOR* **37** 97–113.

Zheng, Y.-S. 1992. On properties of stochastic inventory systems. *Management Sci.* **38** 87–103.

Zipkin, P. H. 2000. *Foundations of Inventory Management*. McGraw-Hill, New York.