A Class of Random Algorithms for Inventory Cycle Offsetting

Ernest Croot
Department of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.
ecroot@math.gatech.edu

Kai Huang
School of Management
Binghamton University, State University of New York
PO Box 6000, Binghamton, New York 13902-6000, U.S.A.
huang@binghamton.edu

August 28, 2008

Abstract

The inventory cycle offsetting problem (ICP) is a strongly NP-complete problem. We study this problem from the view of probability theory, and rigorously analyze the performance of a specific random algorithm for this problem; furthermore, we present a "local search" algorithm, and a modified local search, which give much better results (the modified local search gives better results than plain local search), and leads to good solutions to certain practical instances of ICP, as we demonstrate with some numerical data. The regime where the random algorithm is rigorously proved to work is when the number of items is large, while the time horizon and unit volumes are not too large. Under such natural hypotheses, the Law of Large Numbers, and various quantitative refinements (such as Bernstein's inequality)

come into play, and we use these results to show that there *always* exist good solutions, not merely that good solutions holds with high probability.

1 Introduction

In a multi-item inventory system, the basic problem is the timing and quantity of replenishment orders. Suppose all items have constant demand rates, and the inventory cycles are given. The decision of cycle offsets is known as the Inventory Cycle Offsetting Problem (ICP).

Let k be the index for items, where $k \in \{1, \cdots, K\}$. The inventory cycle of item k is q_k. Within a base cycle framework (c.f., Murphy et al. (2003)), we assume q_k's are natural numbers greater than 1. The volume of resource consumed or occupied by one unit item k in unit time is d_k. They are also called "unit volumes". The integer decision variable is $0 \leq \delta_k < q_k$, the replenishment time for item k. We define the following periodic function for each item k:

$$f_k(t) = q_k - t \quad \forall 0 \leq t < q_k. \tag{1}$$

For all integers t, let h be the unique integer such that $0 \leq t + hq_k < q_k$ and $f_k(t) = f_k(t + hq_k)$.

The ICP problem is to efficiently determine

$$M_T(q_1, ..., q_K; d_1, ..., d_K) \ := \ \underset{\delta_1,\delta_2,...,\delta_K}{\text{Min}} \ \underset{\substack{0\leq t\leq T-1 \\ t\in\mathbb{Z}}}{\text{Max}} \ S(t; \delta_1, ..., \delta_K),$$

where $S(t; \delta_1, ..., \delta_K) = \sum_{k=1}^{K} d_k f_k(t + \delta_k)$, and to determine a choice for $\delta_1, ..., \delta_K$ where this Minimum-Maximum is attained. In the literature, the ICP problem is also called the staggering problem (c.f., Gallego et al. (1992), Gallego et al. (1996)).

The significance of this problem comes from two aspects. First, in many multi-item inventory systems, there exists a resource constraint, which requires that the resource occupied at any time along a finite or infinite time horizon does not exceed a given capacity. Such a resource can be the maximum money tied up in the inventory (c.f., Rosenblatt (1981)), or the maximum available warehouse space (c.f., Gallego et al. (1996)). In a labor intensive environment, e.g., retailer stores, labor could also be a stringent resource (c.f., Erhun and Tayur (2003), van Donselaar et al. (2006)). Second, the ICP problem is a very hard integer program inherent in a larger global optimization problem, which is of theoretical interests to researchers in this field (c.f., Gallego et al. (1992), Gallego et al. (1996), Hariga and Jackson (1996)).

In the following, we give a literature review in Section 2. In Section 3, we present a Law of Large Numbers for the ICP problem, which is based on the Bernstein concentration of measure inequality. In Section 4, we propose a random algorithm and an improved version based on local search. The numerical experiment results in Section 5 show that our algorithms are effective for many practical inventory systems. Finally, we conclude the paper and discuss future research directions in Section 6.

2 Literature review

The inventory replenishment policy for a multi-item inventory system is a fundamental topic in the inventory theory (c.f., Hadley and Whitin (1963), Johnson and Montgomery (1974), Naddor (1966), Tersine (1976), Zipkin (2000)). There are two types of decisions in inventory replenishment. One is the quantity of replenishment, which is determined by the cycle length; the other is the timing of replenishment, which is determined by the cycle offset (δ_k's). The ICP problem tries to answer the latter question, and is a sub problem in a larger global optimization problem.

When there exists a single global resource constraint in a multi-item inventory system, there are usually two approaches. One is the Lagrangian relaxation approach (c.f., Hadley and Whitin (1963), Parsons (1966), Johnson and Montgomery (1974)), where Lagrangian multipliers are used and the global optimization problem is decomposed into single item problems. The other is the equal cycle length approach (c.f., Krone (1964), Parsons (1965), Homer (1966), Page and Paul (1976), Zoller (1977), Goyal (1978)), where all the items share the same cycle, and the decisions are to choose the equal cycle length and the replenishment times for items.

In the 1990s, the advantages of joint replenishment receive intensive study (c.f., Zipkin (2000)). When multiple items are replenished simultaneously, pooling effects will bring significant cost reduction. In joint replenishment policy, usually a base cycle is determined, and the cycle of any item is an integer multiple of the base cycle. So this approach is also called the base cycle approach (c.f., Goyal (1973), Silver (1976), Goyal and Belton (1979), Kapsi and Rosenblatt (1983)). It can be shown when the base cycle length is given, the problem of determining the integer multiples can be regarded as a partitioning problem and solved efficiently in most practical cases (c.f., Chakravarty et al. (1982), Chakravarty et al. (1985)). In Gallego et al. (1996), a variant of the base cycle approach is proposed, where the cycle lengths of items are set to be power of two times of the base cycle length. In Hariga and Jackson (1996), a similar method is discussed.

However, in most base cycle approach papers, there is very little discussion on the cycle offsetting problem; it is usually assumed that there exists times along the horizon, such that the overall resource occupied is just the sum of the maximum resource occupations of individual items. This may be due to the difficulty of solving the cycle offsetting problem as a subproblem. In Gallego et al. (1992), it is shown that the cycle offsetting problem is strongly NP-complete. In Goyal (1978), a heuristic is proposed to incorporate the cycle offsetting within a fixed cycle approach. Then in Hall (1988), the author considers the separate replenishment policy for two items, where the cycle length of one item is defined as the basic cycle, and the other item's cycle length is an integer multiple of the basic cycle. The optimal offsetting solution for this special case is derived. The two item case is also studied in Hartley and Thomas (1982), Thomas and Hartley (1983). In Shaw (1990), the case of no more than three replenishment in each cycle is studied. It is shown that for this special case, the ICP problem can be transformed to the knapsack problem, and a pseudo polynomial algorithm can be used. Later in Murphy et al. (2003), the general two item cycle offsetting problem is solved. By using modular arithmetic, a closed form optimal solution is obtained. Furthermore, the authors show that cycle offsetting can increase resource utilization by as much as fifty percent, which implies that it is very necessary to consider the cycle offsetting effect when the resource is limited.

To conclude, in previous research on the inventory cycle offsetting problem, the two item case was solved, while for the case of larger number of items, only a few heuristics exist.

3 Law of Large Numbers for ICP

Our first observation of ICP is the following theoretical lower bound, whose proof can be found in Croot and Huang (2007).

Theorem 1

$$M_L(q_1, ..., q_K; d_1, ..., d_K) \ \geq \ B,$$

where

$$B := \frac{d_1(q_1+1) + \cdots + d_K(q_K+1)}{2},$$

and L is the least common multiple (LCM) of the q_k's.

On the other hand, it is trivial to see that the upper bound of the ICP problem is:

$$d_1 q_1 + \cdots + d_K q_K.$$

In this section we consider a large number of items, in which case the Law of Large Numbers will come into play. We can show that the existence of a "good" solution can be guaranteed when the distributions of the cycles (the q_k's) and unit volumes (the d_k's) of items satisfy certain conditions, where a "good" solution is a solution providing a total capacity requirement close enough to the theoretical lower bound.

3.1 Statistics

We take the view that the parameters q_k's and d_k's are taken from random distributions. Therefore, the following equations can be regarded as definitions of a series of statistics. The theorems in this section will be stated using these statistics.

$$\begin{aligned}
L &:= \mathrm{LCM}(q_1, ..., q_K) \\
U &:= \operatorname*{Max}_{1 \leq k \leq K} d_k(q_k - 1) \\
\mu_1 &:= \frac{d_1(q_1+1) + \cdots + d_K(q_K+1)}{K} \\
\sigma_1^2 &:= \frac{\sum_{k=1}^{K}[d_k(q_k+1) - \mu_1]^2}{K} \\
\lambda_1 &:= \frac{\sigma_1}{\mu_1} \\
\lambda_2 &:= \frac{U}{\mu_1}. \\
\sigma^2 &:= \frac{1}{12}\sum_{k=1}^{K} d_k^2(q_k^2 - 1).
\end{aligned}$$

Notice that now B can be written as $B = \frac{K\mu_1}{2}$.

3.2 Applications of Bernstein's inequality

In this subsection we will use Bernstein's inequality (Theorem 2) to show that in the case of large number of items, if we pick $\delta_1, ..., \delta_K$ at random, then $S(t; \delta_1, ..., \delta_K)$ is not too large for lots of times t. We first list the Bernstein's inequality without proof:

Theorem 2 (Bernstein's Inequality) *Let $X_1, ..., X_n$ be independent random variables with $\mathbb{E}X_i = 0$, $\mathbb{E}(X_i^2) = \sigma_i^2$, and suppose that $|X_i| \leq c$. Write*

$$\sigma^2 \ = \ \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2,$$

then for any $t \geq 0$ we have the inequality:

$$P(X_1 + \cdots + X_n > nt) \ \leq \ \exp\left(-\frac{n^2t^2}{2\sigma^2 + 2ct/3}\right).$$

Now we consider the following: Suppose we fix a time t, and choose $\delta_1, ..., \delta_K$ independently at random, where

$$\delta_k \ \in \ \{0, 1, ..., q_k - 1\},$$

with the uniform distribution. How close to B would we expect $S(t; \delta_1, ..., \delta_K)$ to come? To answer this question, first let

$$X_k \ = \ d_k f_k(t + \delta_k) - d_k \mathbb{E}(f_k(t + \delta_k)) \ = \ d_k f_k(t + \delta_k) - \frac{d_k(q_k + 1)}{2}, \quad (2)$$

which satisfies

$$\mathbb{E}(X_k) \ = \ 0, \text{ and } V(X_k) \ = \ d_k^2 V(f_k(t + \delta_k)) \ = \ \frac{d_k^2(q_k^2 - 1)}{12}.$$

Then let

$$S \ := \ X_1 + \cdots + X_K.$$

We have $\mathbb{E}(S) = 0$. Since all X_k's are independently chosen,

$$\sigma^2 \ = \ V(S) \ = \ \sum_{k=1}^{K} V(X_k) \ = \ \frac{1}{12} \sum_{k=1}^{K} d_k^2(q_k^2 - 1).$$

Now, we can take for the parameter c in the statement of Bernstein's inequality,

$$c = \frac{U}{2} = \operatorname*{Max}_k \frac{d_k(q_k - 1)}{2};$$

and so, we obtain from Bernstein's inequality

$$P(S > Kt) \leq \exp\left(-\frac{K^2t^2}{2\sigma^2 + Ut/3}\right). \tag{3}$$

With this inequality, we can derive the following theorem:

Theorem 3 *For any fixed $t \geq 0$, if we choose δ_k's independently at random from uniform distributions, then we have*

$$P\left[S(t;\delta_1,\cdots,\delta_K) > (1+\alpha)B\right] \leq \exp\left(-\frac{3K}{2}\frac{\alpha^2\mu_1^2}{\mu_1^2+\sigma_1^2+\alpha\mu_1 U/K}\right). \tag{4}$$

Proof: We have from (3) that

$$P\left(S(t;\delta_1,...,\delta_K) > (1+\alpha)B\right) = P(S > \alpha B) = P[S > K(\alpha B/K)].$$

Letting $t = \frac{\alpha B}{K}$, we deduce from (3), along with the easy-to-verify inequality

$$\sigma^2 \leq \frac{K(\mu_1^2+\sigma_1^2)}{12},$$

that

$$\begin{aligned} P[S(t;\delta_1,...,\delta_K) > (1+\alpha)B] &\leq \exp\left(-\frac{\alpha^2B^2}{2\sigma^2+\alpha UB/3K}\right) \\ &\leq \exp\left(-\frac{3K\alpha^2\mu_1^2}{2\mu_1^2+2\sigma_1^2+2\alpha\mu_1 U/K}\right). \end{aligned}$$

■

Remark. The first observation is that for any fixed t, if we choose appropriate U, μ_1, σ_1, K and α, and if the following inequality holds:

$$\exp\left(-\frac{3K}{2}\frac{\alpha^2\mu_1^2}{\mu_1^2+\sigma_1^2+\alpha\mu_1 U/K}\right) < 1,$$

then

$$P\left[S(t;\delta_1,\cdots,\delta_K) \leq (1+\alpha)B\right] \;\geq\; \epsilon_1, \text{ for some } \epsilon_1 > 0.$$

In other words, there exists a positive probability of finding a value for S that is within $(1+\alpha)$ times the lower bound B for the fixed t. Furthermore, if we regard U, μ_1, σ_1, T and α as fixed parameters, then the right hand side of (4) is an exponentially decreasing function for any t with respect to K. Therefore, as long as K is large enough, we can always guarantee that there is a "good" solution; more precisely,

$$P\left(B \;\lesssim\; \operatorname*{Max}_{0\leq t<T} S(t;\delta_1,\cdots,\delta_K) \;\leq\; (1+\alpha)B\right) \;>\; 0.$$

The reason for the $\lesssim$ here is that we only have $\leq$ here when $T = L$ (i.e., when $T < L$, B may not be a lower bound); however, a result in Croot and Huang (2007) shows that when T is "sufficiently large", then B is still essentially a lower bound for $\text{Max}_{0\leq t<T}\, S(t;\delta_1,...,\delta_K)$.

From the remark, we have the following immediate corollary of Theorem 3:

Corollary 1 *Let*

$$\lambda_1 \;=\; \frac{\sigma_1}{\mu_1}, \text{ and } \lambda_2 \;=\; \frac{U}{\mu_1}.$$

Suppose that

$$\frac{3K}{2}\frac{\alpha^2}{1+\lambda_1^2+\alpha\lambda_2/K} \;>\; \log T. \tag{5}$$

Then, if we choose δ_k's randomly from uniform distributions, we have a strictly positive probability for the event

$$\operatorname*{Max}_{0\leq t<T} S(t;\delta_1,\cdots,\delta_K) \;\leq\; (1+\alpha)B, \tag{6}$$

which implies that

$$M_T(q_1,...,q_K;d_1,...,d_K) \;\leq\; (1+\alpha)B.$$

Proof: Let event A_t be the event

$$S(t;\delta_1,\cdots,\delta_K) \;\leq\; (1+\alpha)B,$$

then the event

$$\underset{0\le t<T}{\text{Max}}\, S(t;\delta_1,\cdots,\delta_K) \;\le\; (1+\alpha)B$$

is the same as event $\cap_{t=0}^{T-1} A_t$. Using Theorem 3 and inequality (5), we have

$$\begin{aligned} P\left(\cap_{t=0}^{T-1} A_t\right) &\ge 1-\sum_{t=0}^{T-1} P(\overline{A_t}) \\ &\ge 1 - T\exp\left(-\frac{3K}{2}\frac{\alpha^2\mu_1^2}{\mu_1^2+\sigma_1^2+\alpha\mu_1 U/K}\right). \end{aligned}$$

The conclusion (6) clearly holds if this last quantity is positive, which is equivalent to (5). ■

Remark. One can make even stronger deductions from Theorem 3, but for the purpose of this paper, Corollary 1 is good enough, as it gives a rough idea of what is possible to prove. Corollary 1 shows that when we have a large number of items, the Law of Large Numbers becomes effective, in which case it is guaranteed that there exists "good" solutions as long as the ratios λ_1 and λ_2 are not too large.

4 Random algorithms

In this section, we present three random algorithms to find "good" solutions in certain practical instances. Furthermore, we present an algorithm that produces better lower bounds than just B for the resource requirement.

4.1 A simple random algorithm

Corollary 1 implies a simple randomized procedure to find a "good" solution when K is large, where a "good" solution is one such that $S(t;\delta_1,\cdots,\delta_K) \le (1+\alpha)B$ for all $0 \le t < T$, given an α that is small enough. Assume inequality (5) holds, we just need to generate a random sample of $\delta_1,\cdots,\delta_K$ from uniform distributions, then we test whether or not (6) holds. If this is true, we claim that $(\delta_1,\cdots,\delta_K)$ is a good solution, otherwise we generate another identically independently distributed sample. We present this randomized procedure in Algorithm 1.

Algorithm 1 A simple random algorithm

1: Given $q_1, ..., q_K$ and $d_1, ..., d_K$, select $0 \le \delta_k \le q_k - 1$ at random using uniform measures.
2: Test the solution $\delta_1, \cdots, \delta_K$. If it is true that $S(t; \delta_1, \cdots, \delta_K) \le (1+\alpha)B$ for all $0 \le t < T$, then stop and return the solution. Otherwise go to Step 1.

Remark. In each iteration of Algorithm 1, the complexity is $\mathcal{O}(T)$.

Note that the effectiveness of Algorithm 1 is guaranteed by the selection of α according to inequality (5). The managerial intuition behind (5) is that for similar values of U, μ_1 and σ_1, the total inventory capacity is directly related to the item number K and time horizon T. When T is fixed, the larger the item number K is, the smaller the value α can be. When K is fixed, the smaller the time horizon T is, the smaller the value α can be. In other words, when we have given U, μ_1 and σ_1 and α, the effectiveness of the randomized procedure depends on the ratio between K and T. When T is larger, we need a larger K for Corollary 1 to work; when T is smaller, we only need a smaller K.

4.2 An improved random algorithm: Basic local search

We can easily improve the performance of the previous algorithm by performing a series of "local searches". One round of local search simply applies the following procedure for a given index $1 \le i \le K$:

Local Search (LS):

1. We assume that we have some values for $\delta_1, ..., \delta_K$ to begin with.
2. Reset δ_i as follows

$$\delta_i \ \leftarrow \ \text{argmin}_{0 \le \delta \le q_i - 1} \operatorname*{Max}_{0 \le t < T} S(t; \delta_1, ..., \delta_{i-1}, \delta, \delta_{i+1}, ..., \delta_K).$$

Note: There may be multiple δ that minimize the maximum S. We choose here any δ producing such a minimum maximum.

This now leads us to an improved random algorithm, given as follows:

Algorithm 2 An improved random algorithm with basic local search

1: Given $q_1, ..., q_K$ and $d_1, ..., d_K$, select $0 \leq \delta_k \leq q_k - 1$ at random using uniform measures.
2: For j from 1 to N do steps 3 and 4.
3: Select i randomly from among the integers $\{1, ..., K\}$ using uniform measures.
4: Perform local search for this chosen value of i.

Remark. In each iteration of Algorithm 2, the complexity is $\mathcal{O}(NQT)$, where Q is the largest q_k value, i.e., $1 \leq q_k \leq Q$ for all k. Note N is a parameter to control the computation effort of the algorithm. In the following experiments, we set $N = 500$.

4.3 An improvement to basic local search

Before we explain how to further improve local search, let us begin with the following observation: Suppose we have two initial sets of values of the δ_k for which we wish to apply local search. The first set of values of the δ_k's produces many times t in the window $0 \leq t < T$ where S is "large", while the second set produces very few times t where S is "large". Which set of values of δ_k's would probably be a better candidate to which to apply local search?

It would seem that the second set would be better, and the reason is as follows: Suppose one applies local search to the first set of δ_k's, say changing the value of δ_i so as to reduce the size of $\text{Max}_{0 \leq t \leq T-1} S(t; \delta_1, ..., \delta_K)$. Since this set of δ_k's leads to lots of times t where S is large, by changing δ_i by a little bit it is true that one can perhaps get S to be a little smaller at one of the peak times, but one is just as likely to make S larger at another peak time. So, one is more likely to get stuck in a "local minimum" where local search does not much help. However, if there were fewer peaks of the function S to begin with (as in the case of the second set of δ_k's), then it is more likely that changing δ_i to reduce the size of one peak value of S at one particular time t, will not much increase the size of S at one of the other peak value times. So, there is a greater chance of not getting stuck in a local minimum.

How might we take advantage of this intuition? One way is to devise and use a type of local search that first adjusts the δ_k's to try to reduce the number of peak values of the function S, and then to apply the local search procedure from subsection 4.2 to the resulting set of δ_k's.

One way to measure whether a choice of δ_k's leads to many times t where S is large (i.e., "many peaks") is just to compute a certain L^c norm: For $c \geq 1$ we let

$$F_c(\delta_1, ..., \delta_k) \;:=\; \sum_{0 \leq t < T} \left(\sum_{1 \leq k \leq K} d_k f_k(t + \delta_k) \right)^c .$$

The larger the value of F_c, the more peaks that the function S enjoys.

This now leads us to the following:

L^c local search (LcLS):

1. Assume that q_k, δ_k, d_k, $k = 1, ..., K$ are given, let i be an index where for which to improve the value of δ_i, and finally let $c \geq 1$ be a choice of exponent.

2. Let $0 \leq \delta \leq q_i - 1$ be chosen so as to minimize $F_c(\delta_1, ..., \delta_{i-1}, \delta, \delta_{i+1}, ...\delta_K)$.

3. Reset $\delta_i \leftarrow \delta$.

Numerical evidence suggests that using $c = 4$ gives good results. Let us now state our modified local search algorithm.

Algorithm 3 An improved random algorithm with modified local search

---

1: Given $q_1, ..., q_K$ and $d_1, ..., d_K$, select all the $\delta_1, ..., \delta_K$ at random using uniform measures.
2: For j from 1 to N_1 do steps 3 and 4.
3: Select i randomly from among the integers $\{1, ..., K\}$ using uniform measures.
4: Perform L^4 local search (L4LS) for this chosen value of i.
5: For j from 1 to N_2 do steps 6 and 7.
6: Select i randomly from among the integers $\{1, ..., K\}$ using uniform measures.
7: Perform regular local serach (LS) for this chosen value of i.

Remark. In each iteration of Algorithm 3, the complexity is $\mathcal{O}(NQT)$, where $N = N_1 + N_2$, and N_1, N_2 are parameters to control the computation effort of the algorithm. In the following experiments, we set $N_1 = 200$ and $N_2 = 300$. Like in Algorithm 2 in subsection 4.2, we run for $N = 500$ iterations in total; however, it turns out that this new algorithm usually has better performance, as numerical studies in Section 5 will demonstrate.

4.3.1 An algorithm for better lower bound on the objective

Note that B is valid when $T = L$. However, when $T \neq L$, B could be an approximate lower bound. Especially when $T \ll L$, B can be very inaccurate. Therefore, there is a need to develop some better lower bound.

We begin by defining the quantity

$$X \;:=\; \operatorname*{Min}_{\substack{0 \le \delta_k \le q_k - 1 \\ k=1,\ldots,K}} \sum_{0 \le t < T} \left(\sum_{k=1}^{K} d_k f_k(t + \delta_k) \right)^2 .$$

Although we don't know the value of this quantity, we can at least efficiently compute a lower bound for it. Note that if we were to drop the "min", and expand the square, we would get

$$\sum_{1 \le i,j \le K} d_i d_j \sum_{0 \le t < T} f_i(t + \delta_i) f_j(t + \delta_j).$$

Regardless of what values of δ_k we choose, this is always bounded from below by

$$\sum_{1 \le i,j \le K} d_i d_j \operatorname*{Min}_{\substack{0 \le \delta_i \le q_i - 1 \\ 0 \le \delta_j \le q_j - 1}} \sum_{0 \le t < T} f_i(t + \delta_i) f_j(t + \delta_j).$$

A slightly more efficient way of computing this is to write it as

$$A_0 + A_1,$$

where

$$A_0 \;:=\; \sum_{1 \le i \le K} d_i^2 \operatorname*{Min}_{0 \le \delta_i \le q_i - 1} \sum_{0 \le t < T} f_i(t + \delta_i)^2,$$

and

$$A_1 \;:=\; 2 \sum_{1 \le i < j \le K} d_i d_j \operatorname*{Min}_{\substack{0 \le \delta_i \le q_i - 1 \\ 0 \le \delta_j \le q_j - 1}} \sum_{0 \le t < T} f_i(t + \delta_i) f_j(t + \delta_j).$$

So, we have that

$$X \;\ge\; A_0 + A_1.$$

Now we arrive at the following, almost trivial inequality that underlies our algorithm for a better lower bound:

$$M_T(q_1, ..., q_K; d_1, ..., d_K) \;\geq\; \sqrt{(A_0 + A_1)/T}.$$

Our algorithm for a lower bound for M_T is now given as follows:

Algorithm 4 An algorithm for better low bound on the objective

---

1: We are given $q_1, ..., q_K, d_1, ..., d_K$ and T.
2: Set $A_0 := 0.0$ and $A_1 := 0.0$.
3: For i from 1 to K do steps 4 through 6.
4: Set $A_0 \leftarrow A_0 + d_i^2 \,\mathrm{Min}_{0\leq\delta_i\leq q_i-1} \sum_{0\leq t<T} f_i(t+\delta_i)^2$.
5: For j from 1 to $i-1$ do step 7.
6: Set $A_1 \leftarrow A_1 + 2 * d_i d_j \,\mathrm{Min}_{\substack{0\leq\delta_i\leq q_i-1\\ 0\leq\delta_j\leq q_j-1}} \sum_{0\leq t<T} f_i(t+\delta_i) f_j(t+\delta_j)$.
7: Return $\sqrt{(A_0 + A_1)/T}$.

Remark. The complexity to compute the better lower bound is $\mathcal{O}(K^2Q^2T)$. In subsection 5.4 we present some data showing how well Algorithm 3 performs, using the better lower bound produced by Algorithm 4.

5 Numerical Experiments

In this section, we first present the performances of Algorithm 1-3 separately. Then we further compare the three algorithms. The results show that Algorithm 3 has the best performance. We also use the better lower bound to further verify the performance of Algorithm 3. Note that all the numerical experiments are implemented in ANSI C.

5.1 Performance of the simple random algorithm

We first present some data on the effectiveness of the simple random algorithm. For each value of K, Q and T, we perform the following procedure 10 times, and we record the average value of the ratio $\mathrm{Max}_{0\leq t<T}\, S(t; \delta_1, ..., \delta_K)/B$ for these 10 runs: We select K random integers $1 \leq q_1, ..., q_K \leq Q$ using the

uniform measure, then select unit volumes $d_1, ..., d_K \in (0, 1]$ also using the uniform measure (actually, the d_i are rational numbers of the form $x/1000$, where natural number x satisfies $1 \leq x \leq 1000$), and finally select offsets $\delta_1, ..., \delta_K$ randomly as in Algorithm 1. The results are presented in Table 1.

From Table 1, we can see the law of large numbers does come into play when the number of items K becomes large. When $K \geq 300$, Algorithm 1 is able to generate offsets whose resource requirement is within 12 percent of the minimum requirement. Furthermore, the impact of T is obvious. Generally speaking, the solution quality becomes better when T becomes smaller.

Table 1: Performance of Algorithm 1

K	Q	T	Average $\text{Max}_{0 \leq t < T} S(t; \delta_1, ..., \delta_K)/B$
300	500	1000	1.10526
100	500	1000	1.19750
300	500	2000	1.11819
100	500	2000	1.20675
50	100	1000	1.31076
50	100	2000	1.33550

5.2 Performance of the basic local search

Next, we present some data on the effectiveness of "basic local search" using $N = 500$ iterations. Basically, for each choice of K, Q and T we perform the 500 iterations of basic local search on 10 random data sets, by selecting the q_i's, d_i's and δ_i's at random using uniform measures. Then we compute the average value of $\text{Max}_{0 \leq t < T} S(t; \delta_1, ..., \delta_K)/B$ for those 10 data sets. The results are presented in Table 2.

From Table 2, we can see that Algorithm 2 performs much better than Algorithm 1. And it is surprising how well the random algorithms can perform. When $K \geq 300$, Algorithm 2 is able to generate offsets whose resource requirement is within 4 percent of the minimum requirement. This means that for a practical inventory system with many items, these random algorithms can find near-optimal solutions.

Table 2: Performance of Algorithm 2

K	Q	T	Average $\text{Max}_{0\leq t<T} S(t;\delta_1,...,\delta_K)/B$
300	500	1000	1.02044
100	500	1000	1.04624
300	500	2000	1.03702
100	500	2000	1.07070
50	100	1000	1.16366
50	100	2000	1.19088

5.3 Performance of the modified local search

We perform the same experiment as in the previous subsection, except that we use "modified local search" in place of "basic local search". The results are presented in Table 3. Note that in every single set of the experiment, for each choice of K, Q and T, Algorithm 3 beats Algorithm 2 and Algorithm 1.

Table 3: Performance of Algorithm 3

K	Q	T	Average $\text{Max}_{0\leq t<T} S(t;\delta_1,...,\delta_K)/B$
300	500	1000	1.01914
100	500	1000	1.03917
300	500	2000	1.03435
100	500	2000	1.05393
50	100	1000	1.11588
50	100	2000	1.15931

One way that we can further improve the random algorithms is to run them multiple times on the same set of $q_1, ..., q_K$ and $\delta_1, ..., \delta_K$, and then take the best choice of $\delta_1, ..., \delta_K$ among all the different runs. This can be rather computationally intensive, so we only carried out the computations for the case $K = 50$, $Q = 100$ and $T = 2000$. The exact procedure we use is given as follows:

1. For i from 1 to 5 do steps 2 through 9
2. Generate a random values $1 \leq q_1, ..., q_{50} \leq 100$, $0 < d_1, ..., d_{50} \leq 1$.

3. For j from 1 to 10 do steps 4 through 7

4. Generate $\delta_1, ..., \delta_{50}$ at random.

5. Run "basic local search" on this set of values of δ_k's, and record the performance.

6. Generate new $\delta_1, ..., \delta_{50}$ at random.

7. Run "modified local search" on this set of values of δ_k's, and record the performance.

8. Among all the runs of 10 runs of "basic local search", record the one leading to the smallest resource requirement. Do the same for "modified local search".

9. Report the results.

The results for $K = 50$, $Q = 100$, and $T = 2000$ are presented in Table 4. Note that in every single instance, the "modified local search" beats the "basic local search".

Table 4: Comparisons of best $\text{Max}_{0 \leq t < T} S(t; \delta_1, ..., \delta_K)/B$

i	Algorithm 2	Algorithm 3
1	1.17667	1.15581
2	1.17401	1.14841
3	1.18944	1.17055
4	1.16541	1.14330
5	1.17849	1.16759

5.4 Modified local search compared against the better lower bound

"Modified local search" performs better than was indicated in the last subsections, because our baseline of comparison was the lower bound B on the resource requirement. So we use the better lower bound as the baseline of comparison, using $K = 50, Q = 100$ and $T = 2000$. In generating our data below, we execute the following procedure five times:

1. Generate random $1 \leq q_1, ..., q_{50} \leq 100$ and $0 < d_1, ..., d_{50} \leq 1$.

2. Set $b := \infty$.

2. For i from 1 to 10 do steps 3 through 5.

3. Generate random $\delta_1, ..., \delta_{50}$, where $0 \leq \delta_k \leq q_k - 1$.

4. Apply "modified local search" using the q_k's, d_k's and δ_k's.

5. If the capacity c produced by "modified local search" is smaller than b, then set $b \leftarrow c$.

6. Run Algorithm 4 using $q_1, ..., q_{50}$ and $d_1, ..., d_{50}$, and let B' be the lower bound on the resource requirement it produces.

7. Compute $B := \sum_k d_k(q_k + 1)/2$.

8. Report the ratios b/B and b/B'.

Notice that we run "modified local search" ten times, and take the best of those ten, as we did in subsection 5.3. The data that our procedure above generates after five runs is presented in Table 5.

Table 5: Performance of Algorithm 3 based on better lower bound

i	b/B	b/B'
1	1.136444	1.117237
2	1.148475	1.134835
3	1.158067	1.140820
4	1.127392	1.114115
5	1.146271	1.132386

The average of these ratios b/B' is roughly 1.12788. So, "modified local search" for $K = 50, Q = 100, T = 2000$ is typically no worse than about 13% above the theoretical minimum when using the best of ten runs. It may be the case that the performance of "modified local search" is in fact better than this, as could be demonstrated with a better lower bound than B' for the theoretical minimum.

6 Conclusions

In this paper we studied the inventory cycle offsetting problem using probability theory and large deviation inequalities. We showed that when there are a large number of items, "good" solutions exist and can be obtained by a random algorithm, and even better solutions can be obtained by applying "basic local search" and "modified local search". The numerical experiment results were especially interesting, as they indicated that "modified local search" can obtain solutions that come rather close to the theoretically best-possible.

There are several future research directions that we will explore:

- First, the analysis in Section 3 shows the simple random algorithm is asymptotically optimal. It would be good to be able to carry out a similar analysis for "basic local search" and "modified local search", and to show that they can give much better solutions than the simple random algorithm.

- Second, it would be good to have a better theoretical lower bound on the best-possible resource requirement than that produced in Algorithm 4. Perhaps there are better algorithms for this that require only modest computing resources (certainly one can do "cubic" and "quadratic" versions of our algorithm, but these require enormously more computing resources than our algorithm).

- Third, the choice of parameters N, N_1, N_2 in "basic local search" and "modified local search", was rather ad hoc. It would be interesting to work out, experimentally, what the best combination of uses of LcLS and LS in "modified local search" are so as to typically produce the smallest resource requirement.

- Fourth, it would be interesting to see whether one can devise random algorithms as good as "modified local search", that allow the demand to vary with time (i.e., stochastic demand).

References

A.K. Chakravarty, J.B. Orlin, and Rothblum U.G. A partition problem with additive objective with an application to optimal inventory groupings for joint replenishment. *Operations Research*, 30(5):1018–1022, 1982.

A.K. Chakravarty, J.B. Orlin, and Rothblum U.G. Consecutive optimizers for a partitioning problem with application to optimal inventory groupings for join replenishment. *Operations Research*, 33(4):820–834, 1985.

E. Croot and K. Huang. Threshold results for the inventory cycle offsetting problem. *Submitted paper*, 2007.

F. Erhun and S. Tayur. Enterprise-wide optimization of total landed cost at a grocery retailer. *Operations Research*, 51(3), 2003.

G. Gallego, M. Queyranne, and D. Simchi-Levi. Single resource multi-item inventory systems. *Operations Research*, 40:580–595, 1996.

G. Gallego, D. Shaw, and D. Simchi-Levi. The complexity of the staggering problem and other classical inventory problems. *Operations Research Letters*, 12:47–52, 1992.

S.K. Goyal. Determination of economic packaging frequency for items jointly replenished. *Management Science*, 20:232–235, 1973.

S.K. Goyal. A note on multi-product inventory situations with one restriction. *The Journal of the Operational Research Society*, 29(3):269–271, 1978.

S.K. Goyal and A.S. Belton. On a simple method of determining order quantities in joint replenishments under deterministic demand. *Management Science*, 25:604, 1979.

J. Hadley and T.M. Whitin. *Analysis of Inventory Systems*. Prentice Hall, Englewood Cliffs, N.J., 1963.

N.G. Hall. Separate vs. joint replenishement policies with maximum storage requirement costs. *European Journal of Operational Research*, 36:180–185, 1988.

M.A. Hariga and P.L. Jackson. The warehouse scheduling problem: formulations and algorithms. *IIE Transactions*, 28:115–127, 1996.

R. Hartley and L.C. Thomas. The deterministic, two-product inventory system with capacity constraint. *Journal of Operational Research Society*, 33: 1013–1020, 1982.

E.D. Homer. Space-limited aggregate inventory with phased deliveries. *Journal of Industrial Engineering*, 17:327–333, 1966.

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

M. Kapsi and M.J. Rosenblatt. A improvement of silver's algorithm for the joint replenishment problem. *IIE Transactions*, 15:264–267, 1983.

L.H. Krone. A note on econocmic lot sizes for multi-purpose equipment. *Management Science*, 10:461, 1964.

N.N. Murphy, W.C. Benton, and P.A. Rubin. Offsetting inventory cycles of items sharing storage. *European Journal of Operational Research*, 150: 304–319, 2003.

E. Naddor. *Inventory Systems*. John Wiley, New York, 1966.

E. Page and R.J. Paul. Multi-product inventory situations with one restriction. *Operational Research Quarterly*, 27(4):815–834, 1976.

J.A. Parsons. A note on krone's econocmic lot size formulas. *Management Science*, 12:314, 1965.

J.A. Parsons. Multi-product lot size determination when certain constraints are active. *Journal of Industrial Engineering*, 17:360–365, 1966.

M.J. Rosenblatt. Multi-item inventory system with budgetary constraint: a comparison between the lagrangian and the fixed cycle approach. *International Journal of Proudction Research*, 19(4):331–339, 1981.

D. Shaw. Complexity and heuristics of a special case of warehouse capacity restricted lot-scheduling problem. Working paper, Department of Industrial Engineering and Operations Research, Columbia University, New York, 1990.

E.A. Silver. A simple method of determining order quantities in joint replenishments under deterministic demand. *Management Science*, 22:1351–1361, 1976.

R.J. Tersine. *Materials Management and Inventory Systems*. North-Holland, New York, 1976.

L.C. Thomas and R. Hartley. An algorithm for limited capacity inventory problem with staggering. *Journal of Operational Research Society*, 34: 81–85, 1983.

K.H. van Donselaar, V. Gaur, T. van Woensel, R.A.C.M. Broekmeulen, and F.C. Fransoo. An empirical study of ordering behavior of retail stores. *Submitted manuscript*, 2006.

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

K. Zoller. Deterministic multi-item inventory systems with limited capacity. *Management Science*, 24:451–455, 1977.