A note on "The Economic Lot Sizing Problem with Inventory Bounds"

Mehmet Önal, Wilco van den Heuvel, Tieming Liu

Econometric Institute Report 2012-04

Abstract

In a recent paper, Liu (2008) considers the lot-sizing problem with lower and upper bounds on the inventory levels. He proposes an $\mathcal{O}(n^2)$ algorithm for the general problem, and an $\mathcal{O}(n)$ algorithm for the special case with non-speculative motives. We show that neither of the algorithms provides an optimal solution in general. Furthermore, we propose a fix for the former algorithm that maintains the $\mathcal{O}(n^2)$ complexity.

1 Introduction and problem description

In the economic lot-sizing problem with inventory bounds (ELSB), one seeks a minimal cost production plan, such that demands are satisfied and inventory levels are between certain bounds in each period of a discrete model horizon of length n. Recently, Liu (2008) developed an $\mathcal{O}(n^2)$ algorithm using the geometric approach of Wagelmans et al. (1992). For the special case of non-speculative motives, he proposed an $\mathcal{O}(n)$ time algorithm. We will show that neither of the algorithms provides an optimal solution in general. We propose a fix for the first algorithm that maintains the $\mathcal{O}(n^2)$ complexity. Love (1973) introduced the ELSB and solved it in $\mathcal{O}(n^3)$ time, while Toczylowski (1995) and Wolsey (2006) improved this to $\mathcal{O}(n^2)$ time by using approaches different from Liu (2008).

To formulate the ELSB, let d_i be the demand (with $d_{i,j} := \sum_{t=i}^{j} d_t$), p_i the unit production cost, h_i the unit inventory carrying cost, K_i the fixed setup cost, and I_i^L (I_i^U)

the lower (upper) bound on the inventory level in period $i = 1, \ldots, n$. Starting from a 'natural' formulation with variables y_i, the production quantity, and I_i, the inventory level in period i, Liu (2008) reformulates the ELSB in terms of the variables $x_i = d_{i+1,n} - I_i$ ($i = 1, \ldots, n$), called the *Net Cumulative Demand* (NCD). The NCD x_i level at the end of period i is the amount of replenishment required to satisfy demands in periods $i+1, \ldots, n$. This reformulation leads to lower (upper) bounds $L_i = d_{i+1,n} - I_i^U$ ($U_i = d_{i+1,n} - I_i^L$) on x_i. Furthermore, by redefining the marginal production costs as $c_i = p_i + \sum_{t=i}^{n} h_t$ ($i = 1, \ldots, n$), a valid formulation for the ELSB is as follows (Liu (2008, Formulation III)):

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} (c_i y_i + K_i \delta(y_i)) & \\
\text{s.t.} \quad & x_i = x_{i-1} - y_i & \text{for } i = 1, \ldots, n, \\
& L_i \leq x_i \leq U_i & \text{for } i = 1, \ldots, n, \\
& y_i \geq 0 & \text{for } i = 1, \ldots, n, \\
& x_0 = d_{1,n}, \; x_n = 0 &
\end{aligned}
$$

with $\delta(x) = 1$ if $x > 0$ and 0 otherwise. Without loss of generality, we may assume that the bounds L_i and U_i are effective, i.e., $L_i \leq L_{i-1}$ and $U_i \leq U_{i-1}$ for $i = 1, \ldots, n$.

2 The issue in the $\mathcal{O}(n^2)$ algorithm

2.1 Dynamic programming algorithm

We first describe the dynamic programming (DP) recursion proposed in Liu (2008). Let $C_m(x_i, x_j)$ be the cost of replenishing an amount of $x_i - x_j$ in period m:

$$
C_m(x_i, x_j) = \begin{cases} K_m + c_m(x_i - x_j) & \text{if } x_i > x_j \\ 0 & \text{if } x_i = x_j \\ \infty & \text{otherwise.} \end{cases}
$$

Let $e(i)$ be the latest period that can be completely satisfied by a replenishment in period i, i.e., $e(i) = \max\{j : U_j \geq L_i\} = \max\{j : d_{i+1,j} + I_j^L \leq I_i^U\}$. The DP relies on the characterization of extreme point solutions. In an extreme point solution, there is at most one replenishment period between two periods $i < j$ whose NCD levels are at their lower or upper bounds, i.e., $x_i \in \{U_i, L_i\}$ and $x_j \in \{U_j, L_j\}$.

Let $G_i(x_i)$, $i = 0, \ldots, n$, be the cost of the optimal production plan from period $i+1$ to n given a NCD level of x_i in period i , where $G_n(x) = 0$ for all x. Because an optimal solution is found among the extreme point solutions, we only need to consider $G_i(x)$ for $x \in \{U_i, L_i\}$. The aim is to compute $G_0(U_0) = G_0(d_{1,n})$ by the recursive equations

$$G_i(U_i) = \min\left\{C_{i+1}(U_i, L_{i+1}) + G_{i+1}(L_{i+1}), \min_{i+1\le j\le e(i+1)} \left(C_{i+1}(U_i, U_j) + G_j(U_j)\right)\right\} \tag{1}$$

$$G_i(L_i) = \min\left\{\min_{i+1\le k\le e(i)+1} \left(C_k(L_i, L_k) + G_k(L_k)\right), \min_{i+1\le m\le e(i)+1, e(i)+1\le j\le e(m)} \left(C_m(L_i, U_j) + G_j(U_j)\right)\right\}. \tag{2}$$

Clearly, a straightforward implementation of (1) and (2) leads to an $\mathcal{O}(n^3)$ algorithm.

2.2 The $\mathcal{O}(n^2)$ implementation proposed in Liu (2008)

To implement (1) and (2) more efficiently, Liu (2008) extends the geometric technique of Wagelmans et al. (1992). To illustrate this, consider the computation of $G_i(U_i)$ by (1) given the values $G_j(U_j)$ for $j = i+1, \ldots, n$. The second term is equivalent to

$$K_{i+1} + \min_{i+1\le j\le e(i+1)} \{c_{i+1}(U_i - U_j) + G_j(U_j)\}. \tag{3}$$

To evaluate (3) efficiently, Wagelmans et al. (1992) utilize the lower convex envelope of the points $(U_\ell, G_\ell(U_\ell))$ for $\ell = i+1, \ldots, e(i+1)$, denoted by $LE(i)$. The breakpoints and endpoints on the envelope are called *efficient points* and the corresponding periods are called *efficient periods*, denoted by $E(i)$. As illustrated in Figure 1, the minimum in (3) is obtained at the period corresponding to the efficient point tangent to the line with slope c_{i+1}. This point is denoted by $\tau(i+1)$ in Liu (2008) (so $\tau(i+1) = j$ in Figure 1) and can be determined in $\mathcal{O}(\log n)$ time by binary search. Therefore, (1) can be simplified to

$$G_i(U_i) = \min\left\{C_{i+1}(U_i, L_{i+1}) + G_{i+1}(L_{i+1}), C_{i+1}(U_i, U_{\tau(i+1)}) + G_{\tau(i+1)}(U_{\tau(i+1)})\right\}. \tag{4}$$

Given the $\tau(i)$ values, (4) takes constant time for fixed i. To compute $G_{i-1}(U_{i-1})$ in the next iteration, the point $(U_i, G_i(U_i))$ is included in the graph and the envelope is updated.

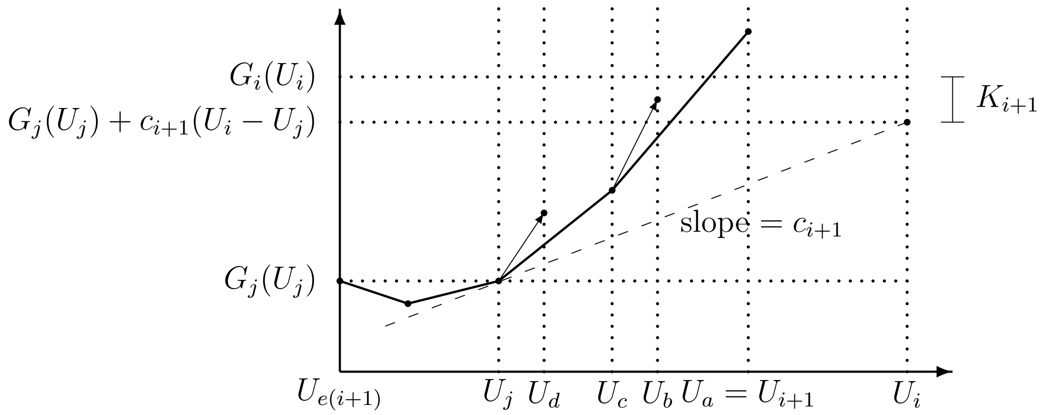


Figure 1: Illustration of the convex lower envelope and the dominance relations

Now consider the computation of $G_i(L_i)$ in (2). The bottle neck is the second term, i.e., finding the optimal replenishment period m ($i+1 \leq m \leq e(i)+1$) that reduces the NCD level down to U_j for some j with $e(i)+1 \leq j \leq e(m)$. As in Liu (2008), we call i the *origination period*, m the *replenishment period*, and j the *destination period*. Liu (2008) claims that the destination period corresponding to each replenishment period m ($i+1 \leq m \leq e(i)+1$) is equal to $\tau(m)$ and hence independent of the origination period. Since the values $\tau(m)$ have already been found in the computation of $G_\ell(U_\ell)$ ($\ell = i+1, \ldots, n$), they can be directly used in the computation of $G_i(L_i)$. Therefore, Liu (2008) simplifies (2) to

$$G_i(L_i) = \min \left\{ \min_{i+1 \leq j \leq e(i)+1} \left(C_j(L_i, L_j) + G_j(L_j) \right), \min_{i+1 \leq m \leq e(i)+1} \left(C_m(L_i, U_{\tau(m)}) + G_{\tau(m)}(U_{\tau(m)}) \right) \right\}. \tag{5}$$

Given the values $\tau(m)$ ($i+1 \leq m \leq e(i)+1$), (5) can be evaluated in $\mathcal{O}(n)$ for a fixed index i. Liu (2008) claims that the lower envelope can be updated in $\mathcal{O}(n)$ overall time. Furthermore, it takes $\mathcal{O}(n \log n)$ time to compute the $\tau(i)$ values for $i = 1, \ldots, n$. Therefore, the values $G_i(L_i)$ for $i = 1, \ldots, n$ can be computed in $\mathcal{O}(n^2)$ time. As a result, the overall time complexity of the algorithm of Liu (2008) becomes $\mathcal{O}(n^2)$.

2.3 The issue in the implementation

As explained in the previous section, Liu (2008) is able to reduce the computational complexity to $\mathcal{O}(n^2)$ by claiming that "given the marginal cost c_m in the replenishment period, the optimal destination period can be determined independent of the origination period". Recall that for a replenishment period m, $\tau(m)$ is found using the lower envelope $LE(m-1)$ of points $(U_\ell, G_\ell(U_\ell))$ for $\ell = m, \ldots, e(m)$. However, as seen in (2), if the origination period is i, the destination period for m should be within the periods $\{e(i)+1, \ldots, e(m)\}$, which is a subset of $\{m, \ldots, e(m)\}$. Therefore, the destination period $\tau(m)$ found in the computation of $G_{m-1}(U_{m-1})$ may be an infeasible destination period in the computation of $G_i(L_i)$. As a result, the recursive equations (4) and (5) do not lead to an optimal solution in general. We illustrate the issue in Liu (2008) in Example 1.

Example 1. Consider the 4 period problem instance with $d_i = 2$, $I_i^L = 0$ $(i = 1, \ldots, 4)$ and the other parameters as in Table 1. Table 2 shows the values of the recursion variables

i	0	1	2	3	4
I_i^U		3	3	1	0
K_i		0	0	3	0
c_i		0	2	0	0
L_i	8	3	1	1	0
U_i	8	6	4	2	0
$e(i)$	0	2	3	3	4

Table 1: Problem instance to illustrate the mistake in Liu (2008)

$G_i(L_i)$ and $G_i(U_i)$, when we either apply (1) and (2), or (4) and (5). When applying the correct recursive equations (1) and (2), we find the optimal solution $(y_1, y_2, y_3, y_4) = (5, 1, 0, 2)$ with total cost equal to 2. However, applying (4) and (5) gives the solution $(y_1, y_2, y_3, y_4) = (5, 0, 2, 1)$ with total cost equal to 3.

	use of (1) and (2)					use of (4) and (5)				
i	0	1	2	3	4	0	1	2	3	4
$G_i(U_i)$	2	7	3	0	0	3	7	3	0	0
$G_i(L_i)$	2	2	0	0	0	3	3	0	0	0

Table 2: Application of the recursive equations

The issue of Liu (2008) is illustrated in the computation of $G_1(L_1)$. The true minimum is attained for $m = 2$ and $j = 3$ in (2): $G_1(3) = C_2(3,2) + G_3(2) = 2 \cdot (3-2) + 0 = 2$. In fact, $j = 3$ is the only feasible destination period corresponding to origination period $i = 1$ and replenishment period $m = 2$. In order to apply (5), we first need to find $\tau(2)$ by computing the lower convex envelope of the points $(U_i, G_i(U_i))$ for $i = 2, 3$ (since $e(2) = 3$). This results in the set of efficient points $\{(4,3), (2,0)\}$. The line with slope $c_2 = 2$ is tangent to the point $(U_2, G_2(U_2)) = (4,3)$ and hence $\tau(2) = 2$ turns out to be the optimal destination period in the algorithm of Liu (2008). Substituting this in the second term of (5) gives the cost term $C_2(3,4)+G_2(4)$, which corresponds to an infeasible solution since it implies a production quantity of $y_2 = -1$ (note that $C_2(3,4) = \infty$). Since origination period $i = 1$ and replenishment period $m = 2$ are part of the optimal solution, it will not be found when applying (4) and (5). Instead, the (wrong) minimum of $G_1(L_1)$ is attained in the first term of (5) for $j = 3$: $G_1(L_1) = C_3(L_1, L_3) + G_3(L_3) = 3 + 0 = 3$.

2.4 Fix of the issue

The $\mathcal{O}(n^2)$ algorithm in Liu (2008) does not guarantee an optimal solution because the destination period does not only depend on the replenishment period but also on the origination period. We fix the issue by computing the optimal destination periods for every pair (i, m) of origination and replenishment periods. Let $\tau(i, m)$ be the optimal destination period of the replenishment period m when the origination period is i. We can then rewrite (2) as

$$G_i(L_i) = \min \left\{ \min_{i+1 \le j \le e(i)+1} \left(C_j(L_i, L_j) + G_j(L_j) \right), \min_{i+1 \le m \le e(i)+1} \left(C_m(L_i, U_{\tau(i,m)}) + G_{\tau(i,m)}(U_{\tau(i,m)}) \right) \right\}. \tag{6}$$

Given $\tau(i, m)$ for $m = i+1, \ldots, e(i)+1$, and $G_j(L_j)$ and $G_j(U_j)$ for $j = i+1, \ldots, n$, computing (6) takes $\mathcal{O}(n)$ time for fixed i. Therefore, to maintain an overall time complexity of $\mathcal{O}(n^2)$, it is sufficient to show that all $\tau(i, m)$ values can be determined in $\mathcal{O}(n^2)$ time.

To describe our approach, we need some additional notation. Let $b(j)$ be the earliest feasible origination period for period j, i.e., $b(j) = \min\limits_{1 \le i \le n} \{i : e(i) + 1 \ge j\}$. As shown in Section 2.3, to determine $\tau(i, m)$, we need the lower envelope of the points $(U_\ell, G_\ell(U_\ell))$ for $\ell = e(i)+1, \ldots, e(m)$, which we denote by $LE(i, m)$ ($i = 1, \ldots, n$, $m = i+1, \ldots, e(i)+1$).

For convenience, let $LE(i,i)$ be the lower envelope of the single point $(U_{e(i)}, G_{e(i)}(U_{e(i)}))$ and let $\tau(m,m) = e(m)$. To determine all the $\tau(i,m)$ values ($i = b(m), \ldots, m-1$, $m = 1, \ldots, n$) using the geometric approach efficiently, we exploit the following property.

Theorem 1. *It holds $\tau(i,m) \leq \tau(i+1,m)$ for $i = b(m), \ldots, m-1$ and $m = 1, \ldots, n$.*

Proof. For a given pair (i,m), the destination period $\tau(i,m)$ is the period where the line with slope c_m is tangent to $LE(i,m)$. Now consider $\tau(i+1,m)$ and the corresponding lower envelope $LE(i+1,m)$, i.e., the lower envelope of the points $(U_\ell, G_\ell(U_\ell))$ for $\ell = e(i+1)+1, \ldots, e(m)$. If $\tau(i,m) \geq e(i+1)+1$, then the line with slope c_m is still tangent to $LE(i+1,m)$ at period $\tau(i,m)$ (since the slopes to the right of $\tau(i,m)$ have not become smaller) and we have $\tau(i+1,m) = \tau(i,m)$. On the other hand, if $\tau(i,m) < e(i+1)+1$, then it is immediate that $\tau(i,m) < e(i+1)+1 \leq \tau(i+1,m)$. $\square$

Our approach is summarized in Algorithm 1. Instead of updating a single lower envelope as in Liu (2008), we construct a lower envelope from scratch in each iteration. At the iteration for origination period i, we start by creating the lower envelope $LE(i+1,i+1) = \{(U_{e(i+1)}, G_{e(i+1)}(U_{e(i+1)}))\}$. We then determine the destination period of $i+1$ for all possible origination periods h, $b(i+1) \leq h \leq i$, in decreasing order of h (see lines 4–7). First, we obtain $LE(h,i+1)$ by adding the points $(U_\ell, G_\ell(U_\ell))$, $\ell = e(h+1), \ldots, e(h+2)-1$, to $LE(h+1,i+1)$ using the update procedure described in Wagelmans et al. (1992). Since $\tau(h,i+1) \leq \tau(h+1,i+1)$ by Theorem 1, we can determine $\tau(h,i+1)$ by a monotonic search that starts from period $\tau(h+1,i+1)$. In this way, we obtain all relevant $\tau(i,m)$ values needed in the computation of $G_i(L_i)$.

In contrast to Liu (2008), points are only added to the envelope, while no deletions are needed in Algorithm 1. Using the update procedure in Wagelmans et al. (1992), it takes $\mathcal{O}(1)$ amortized time to add a point to the envelope. Therefore, for a fixed index i, it takes $\mathcal{O}(n)$ overall time to build the lower envelopes between lines 4 and 7. The monotonic search evaluates every efficient point on the envelope at most once. Hence, it takes an additional $\mathcal{O}(n)$ time to determine $\tau(h,i+1)$ for $h = i, \ldots, b(i+1)$. Therefore, the operations in the inner loop take $\mathcal{O}(n)$ time. When we reach line 8, we have all the required destination periods, and we can compute $G_i(L_i)$ in $\mathcal{O}(n)$ time. Since the

Algorithm 1 The proposed $\mathcal{O}(n^2)$ algorithm

```
1: Set G_n(L_n) = G_n(U_n) = 0
2: for i = n − 1 to 0 do
3:     Create LE(i + 1, i + 1) and Set τ(i + 1, i + 1) := e(i + 1)
4:     for h = i to b(i + 1) do
5:         Obtain LE(h, i + 1) from LE(h + 1, i + 1)
6:         Determine τ(h, i + 1) by a monotonic search starting from τ(h + 1, i + 1)
7:     end for
8:     Compute G_i(L_i) by (6)
9:     Compute G_i(U_i) by (1)
10: end for
```

computation of $G_i(U_i)$ by (1) takes $\mathcal{O}(n)$ as well, $G_i(L_i)$ and $G_i(U_i)$ are calculated in $\mathcal{O}(n)$ time for a fixed index i. Hence, the overall time complexity of the algorithm is $\mathcal{O}(n^2)$.

3 The issue in the $\mathcal{O}(n)$ algorithm

Liu (2008) also considers the special ELSB where the marginal production costs satisfy $c_i \geq c_{i+1}$ ($i = 1, \ldots, n-1$), also known as non-speculative motives in the literature. In this case there exists an optimal solution such that between any two consecutive production periods, there is a period for which the NCD (inventory) level is equal to the upper (lower) bound. (If additionally $I_i^L = 0$ ($i = 1, \ldots, n$), then this property reduces to the zero-inventory ordering property.) Due to this property, (1) and (2) simplify to

$$G_i(U_i) = \min_{i+1 \leq j \leq e(i+1)} \{C_{i+1}(U_i, U_j) + G_j(U_j)\} \tag{7}$$

or, since period j is the optimal destination period for replenishment period $i+1$, to

$$G_i(U_i) = C_{i+1}(U_i, U_j) + G_{\tau(i+1)}(U_{\tau(i+1)}). \tag{8}$$

To determine $\tau(i+1)$ using the geometric technique, the lower convex envelope $LE(i)$ of the points $(U_\ell, G_\ell(U_\ell))$ ($\ell = i+1, \ldots, e(i+1)$) should be available. Since $c_i \geq c_{i+1}$ ($i = 1, \ldots, n-1$), it follows that $\tau(i+1) \geq \tau(i)$. Hence, it is sufficient to investigate efficient periods to the right of $\tau(i+1)$ to determine $\tau(i)$. In turn, this means that the binary search to find the destination period can be replaced by a monotonic search.

To determine $G_{i-1}(U_{i-1})$, Liu (2008) computes $LE(i-1)$ by updating the left and right borders of $LE(i)$. The right border is updated by including $(U_i, G_i(U_i))$ in $LE(i)$ as in Wagelmans et al. (1992). In the updating process, some originally efficient points may leave the lower envelope. Using the terminology of Liu (2008), such a point is said to be *dominated* by the newly included point $(U_i, G_i(U_i))$ and *right dominated* by the originally efficient point on its immediate left. The set of periods that period i right dominates is denoted by $R(i)$. Before adding the point $(U_i, G_i(U_i))$ to $LE(i-1)$ in Figure 1, $R(c) = \{b\}$ and $R(j) = \{d\}$. After adding the point, periods $a = i+1$ and c leave the lower envelope. Period a becomes right dominated by period c and period c by period j, as indicated by the arrows. Hence, $R(c) = \{a, b\}$ and $R(j) = \{d, c\}$.

To obtain $LE(i-1)$ from $LE(i)$, the left border of $LE(i)$ needs to be updated as well if $e(i+1) > e(i)$. Liu (2008) proposes to add the periods $t \in R(l)$ $(e(i) < l \leq e(i+1))$ with $t \leq e(i)$ straight to the set $E(i)$. It takes $\mathcal{O}(n)$ amortized time to update the right and left borders of $LE(i-1)$ from $LE(i)$ for $i = 1, \ldots, n$. Hence, the overall time complexity becomes $\mathcal{O}(n)$. However, Liu's algorithm does not update the lower envelope correctly (the convexity is lost), which is crucial to apply the monotonic search. Hence, the $\mathcal{O}(n)$ algorithm does not lead to an optimal solution in general, as shown in Example 2.

Example 2. Consider the 5 period problem instance with $d_i = 1$, $I_i^L = 0$ $(i = 1, \ldots, 5)$ and the other data as in Table 3. The left part of Table 4 shows the summary of the

i	0	1	2	3	4	5
I_i^U		3	3	2	1	0
K_i		0	3	1	5	3
c_i		1	0	0	0	0
L_i	5	1	0	0	0	0
U_i	5	4	3	2	1	0
$e(i)$		4	5	5	5	5

Table 3: Problem instance illustrating the mistake in the $\mathcal{O}(n)$ algorithm

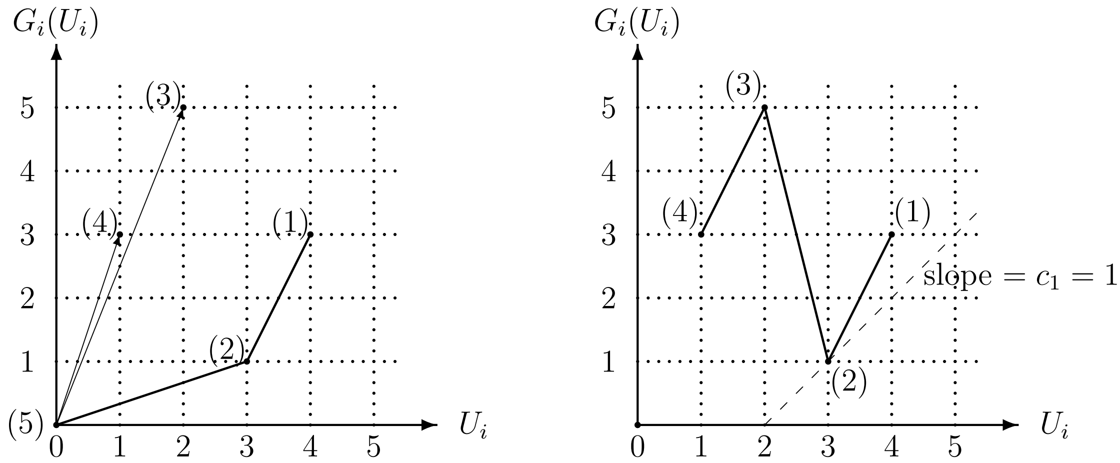
computations if the left border of the envelope is updated as described in Liu (2008), while the right part shows the correct computations.

Consider the computation of $\tau(1)$ in $G_0(U_0)$. The envelope at the start of the iteration

	computations according to Liu (2008)						correct computations					
i	5	4	3	2	1	0	5	4	3	2	1	0
$E(i)$	$\emptyset$	{5}	{5,4}	{5,3}	{5,2}	{4,3,2,1}	$\emptyset$	{5}	{5,4}	{5,3}	{5,2}	{4,2,1}
$\tau(i)$	5	5	5	5	4		5	5	5	5	2	
$G_i(U_i)$		3	5	1	3	7		3	5	1	3	3

Table 4: Computations associated with Example 2

is illustrated in Figure 2(a). We see that $R(5) = \{4, 3\}$. To find $\tau(1)$, the left border of the envelope has to be updated since $e(1) = 4 < 5 = e(2)$. Following the update process described in Liu (2008), we remove the point $(U_5, G_5(U_5))$ from the envelope and add the points $(U_4, G_4(U_4))$ and $(U_3, G_3(U_3))$ to obtain the lower envelope as in Figure 2(b). Clearly, this envelope is not convex anymore. Nevertheless, starting at period 4, we find $\tau(1) = 4$ since the first line segment with a slope higher than c_1 is the first one in the envelope. The solution found by Liu's algorithm is $(y_1, y_2, y_3, y_4, y_5) = (4, 0, 0, 0, 1)$ with a total cost of 7. However, as shown in Figure 2(b) the true value of $\tau(1) = 2$, resulting in the optimal solution $(y_1, y_2, y_3, y_4, y_5) = (2, 0, 3, 0, 0)$ with a total cost of 3.



(a) Envelope at the start of computing $G_1(U_1)$ (b) $E(0)$ after updating the left border

Figure 2: Computation of $G_1(U_1)$ by the convex envelopes

To the best of our knowledge, there is no fix based on the geometric approach of Wagelmans et al. (1992) and maintains the $\mathcal{O}(n)$ running time. Brodal and Jacob (2002) present a data structure to maintain a convex hull of n points in the plane under insertion and deletion of points in amortized $\mathcal{O}(\log n)$ time per operation. A straightforward application of this result leads to an $\mathcal{O}(n \log n)$ algorithm. Recently, Hwang and van den Heuvel (2012) proposed an $\mathcal{O}(n)$ time algorithm by using another type of geometric technique, which maintains a lower envelope of line segments. In their approach, line segments only need to be added to an existing envelope and no deletions are required. This property together with the non-speculative motives cost structure allows for an $\mathcal{O}(n)$ implementation.

References

Brodal, G. S., Jacob, R., 2002. Dynamic planar convex hulls. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science. pp. 617–626.

Hwang, H.-C., van den Heuvel, W., 2012. Improved algorithms for a lot-sizing problem with inventory bounds and backlogging. Naval Research Logistics (forthcoming).

Liu, T., 2008. Economic lot sizing with inventory bounds. EJOR 185, 204–215.

Love, F., 1973. Bounded production and inventory models with piecewise concave costs. Management Science 20, 313–318.

Toczylowski, E., 1995. An $O(T^2)$ algorithm for the lot-sizing problem with limited inventory levels. In: INRIA/IEEE Symposium on Emerging Technologies and Factory Automation. Vol. 3. pp. 77–85.

Wagelmans, A. P. M., van Hoesel, C. P. M., Kolen, A., 1992. Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. OR 40, S145–S156.

Wolsey, L. A., 2006. Lot-sizing with product and delivery time windows. MP 107, 471–489.