Lot Sizing with Minimum Order Quantity

Young Woong Park and Diego Klabjan
Department of Industrial Engineering and Management Sciences
Northwestern University, Evanston, IL, USA.
ywpark@u.northwestern.edu, d-klabjan@northwestern.edu

Jan 19, 2013; revised June 9, 2014

Abstract

We consider single item lot sizing with minimum order quantity where each period has an additional constraint on minimum production quantity. We study special cases of the general problem from the algorithmic and mathematical programming perspective. We exhibit a polynomial case when capacity is constant and minimum order quantities are ordered in non-increasing order. Linear programming extended formulations are provided for various cases with constant minimum order quantity.

1 Introduction

The single item economic lot sizing problem is to find the production lot sizes of one item over several periods with the minimum cost. In this paper, we consider the single item lot sizing problem with minimum order quantity (MOQ). MOQ is an order requirement imposing that the amount of the production has to be at least a certain quantity when the period has a positive production. The MOQ requirement can be a hard constraint if it is due to business requirements such as the product required to be shipped in containers or pallets. However, it can be used as an alternative to fixed ordering cost or setup cost, as both of them prevent small orders that cause high per unit fixed or setup cost [1]. Hence, the MOQ requirement is an alternative way to achieve economies of scales in production and transportation [2]. Musalem and Dekker [1] and Zhao and Katehakis [2] provide real world cases where MOQ is used.

Given T periods, demands $d_1, \cdots, d_T$ must be satisfied by a sequence of production schedules, where the production level in period t must be at least l_t and no more than u_t, if it is positive, for all $t = 1, \cdots, T$. For notational convenience, let $d_{i,j} = \sum_{t=i}^{j} d_t$ be the summation of the demand from period i to j for $1 \leq i \leq j \leq T$. The feasibility set of the single item lot sizing problem with MOQ is a set $S(l_t, u_t) \subseteq \mathbb{R}^T$ defined as

$$S(l_t, u_t) = \left\{ \begin{array}{lll} x \in \mathbb{R}^T: & \sum_{t=1}^{j} x_t \geq d_{1,j}, & j = 1, \cdots, T \\ & x_t \in \{0\} \cup [l_t, u_t], & t = 1, \cdots, T \end{array} \right\},$$

where x_t is the production level in period t. When there is no MOQ, we denote the set as $S(\emptyset, u_t)$ and when the MOQ's are constant, we denote the set as $S(l, u_t)$. When there exists no upper bound, we denote the set as $S(l_t, \infty)$. Similarly, $S(l, u)$, $S(l, \infty)$, $S(\emptyset, u)$ are defined. In the literature, based on our convention, $S(\emptyset, u_t)$ and $S(\emptyset, u)$ is referred as capacitated lot sizing, and $S(\emptyset, \infty)$ is referred as uncapacitated lot sizing. Any feasibility set with a lower bound such as $S(l_t, u)$, $S(l, u)$, or $S(l_t, \infty)$ is referred as lot sizing with MOQ.

In our work, we consider time-dependent and constant MOQ with constant capacity on a finite time horizon. For the lot sizing problem with time dependent MOQ, we present a polynomial algorithm for the single item lot sizing problem with non-constant and non-increasing MOQ with non-increasing linear costs. The algorithm is based on the well-known dynamic programming algorithm of Florian and Klein [3], and it iterates over a set of tuples of parameters to find a solution to a production sequence. For each tuple, it preprocesses the demand and finds the production schedule greedily. For the lot sizing problem with constant MOQ, we present linear programming extended formulations for the lot sizing problem with constant MOQ and upper bound with non-increasing linear costs.

Since the seminal works of Manne [4] and Wagner and Whitin [5], the lot sizing problem has been extensively studied. In this review, we focus on the lot sizing problem such that demand is deterministic and order quantity is the only decision. To decrease the scope of the review further and to align it with the work presented herein, we consider

1. polynomial algorithms for the single item lot sizing problem with MOQ, and
2. polyhedral study and linear programming extended formulations for the lot sizing problem and related problems.

Since Anderson and Cheah [6] introduced the MOQ constraint for the multi-item lot sizing problem, there have been several studies on polynomial algorithms for the single item lot sizing problem with MOQ. Lee [7] provided the first polynomial algorithm for $S(l,\infty)$. Li *et al* [8] exhibit a polynomial algorithm for $S(l_t,\infty)$ with non-increasing cost and MOQ. Okhrin and Richter [9] also studied an algorithm for $S(l,\infty)$. Okhrin and Richter [10] provided a polynomial algorithm for $S(l,u)$ with constant holding cost. Hellion *et al* [11] also studied $S(l,u)$ with concave costs. Our polynomial case for $S(l_t,u)$ is different from the other polynomial cases as no previous study considered non-constant MOQ together with upper bounds. The polynomial cases for the lot sizing problem with MOQ are summarized in Table 1, considering only the different capacity and MOQ requirements.

	Constant capacity	Uncapacitated
Constant MOQ	$S(l,u)$: Hellion *et al* [11] Okhrin and Richter [10]	$S(l,\infty)$: Lee [7] Okhrin and Richter [9]
Non-constant MOQ	$S(l_t,u)$: Section 2 in this paper	$S(l_t,\infty)$: Li *et al* [8]

Table 1: Polynomial cases with MOQ

Polyhedra and LP extended formulations for the lot sizing problem also have been studied in the literature. Pochet [12] studied valid inequalities and facets of $S(\emptyset,u)$. Pochet and Wolsey [13] gave a tight and compact reformulation for $S(\emptyset,u)$ in the presence of the Wagner-Whitin cost. Constantino [14] studied the polyhedron of a relaxation of $S(l,u)$. Van Vyve [15] provided LP extended formulations for $S(\emptyset,u)$ with backlogging. Anily *et al* [16] provided an LP extended formulation for multi-item lot sizing where each item belongs to $S(\emptyset,u)$. Pochet and Wolsey [17] proposed a compact mixed integer programming reformulation whose linear programming relaxation solves $S(\emptyset,u_t)$ when capacities u_t's are non-decreasing over time. To the best of our knowledge, an LP extended formulation for the single item lot sizing problem with MOQ has not yet been studied. Recently, Angulo *et al* [18] studied the semi-continuous inflow set of a single node of the type $S(l_t,\infty)$. They provided an LP extended formulation for the semi-continuous inflow set. Our work is distinguished from the work in [18], since $S(l,u)$ is very different from the set they considered.

Our contribution can be summarized as follows.

1. In addition to the cases that are already proved to be polynomial, we identify that $S(l_t,u)$ with non-increasing linear costs and non-increasing l_t's can be solved in polynomial time, by providing a polynomial algorithm for the first time in the literature.
2. We provide various LP extended formulations for $S(l,u)$ and $S(l,\infty)$ with non-increasing costs. The proposed formulations are the first LP extended formulations for the single item lot sizing problem with the presence of MOQ.

The rest of the paper is organized as follows. Section 2 provides a polynomial time algorithm for $S(l_t,u)$ with additional assumptions on the orders of the lower bounds and objective cost coefficients. Section 3 develops the linear programming extended formulations for $S(l,u)$ and $S(l,\infty)$ with non-increasing production and fixed costs. In Section 4, we present the computational experiments for the proposed algorithm and formulations.

2 Polynomial Case

In this section, we show that $S(l_t,u)$ with a linear cost function can be solved in polynomial time if we additionally assume the following.

Non-increasing cost	$p_1 \geq p_2 \geq \cdots \geq p_T$
Non-increasing lower bound	$l_1 \geq l_2 \geq \cdots, \geq l_T$

We also assume that the demand satisfies $d_t \leq u$ for all t. The reader is referred to the book by Pochet and Wolsey [19] for the justification of this assumption after an appropriate preprocessing of d_t's. The key idea of the preprocessing is that, when $d_t > u_t$, the amount of demand $d_t - u_t$ exceeding u_t in time t must be fulfilled prior to t, and forwarding $d_t - u_t$ units of demand from t to $t-1$ does not affect the solution space and optimality. For our problem, we execute $d_{t-1} := d_{t-1} + \max\{d_t - u, 0\}$ and $d_t := \min\{u, d_t\}$ for $t = T$ down to 2. We refer to this procedure as *DemandForward*.

Demand assumption	$0 \leq d_t \leq u$ for all $t = 1, \cdots, T$

In the rest of the paper, we consider an optimal solution x^* and the corresponding inventory I^* defined as $I^* = \{I_1^*, I_2^*, \cdots, I_T^*\}$ with $I_t^* = \sum_{i=1}^{t} x_i^* - d_{1,t}$. We assume that x^* delays the production as much as possible. Hence, among all optimal solutions, we select the one that delays the production as much as possible.

Delayed optimal assumption	Optimal solution x^* delays production the most among all possible choices of optimal solutions

We stress that this is not really an assumption but we single it out here since we will often refer to it. In the lot sizing literature, many polynomial algorithms use the concept of a production sequence and the dynamic programming algorithm proposed by Florian and Klein [3]. We next briefly summarize the concepts and terms. Let period t with $I_t = 0$ be a *regeneration point* and let S_{ab} be a subset of a feasible production plan between two consecutive regeneration points a and b. We call S_{ab} a *production sequence*. Note that S_{ab} has $I_a = I_b = 0$ and $I_t > 0$ for $t = a+1, \cdots, b-1$. We assume that S_{ab} cannot be broken into two or more production sequences with equal or lower cost. Note also that any optimal production plan can be decomposed into a set of consecutive production sequences. Hence, a dynamic programming algorithm can be used to find an optimal set of production sequences.

However, for lot sizing with MOQ, we cannot assume zero ending inventory for every production sequence. The authors in [11] pointed out that the production sequence including last period T can have strictly positive ending inventory. They also proposed a modified network by adding nodes for production sequence S_{aT}, for $a \leq T$, with positive ending inventory.

The rest of this section is organized as follows. In Section 2.1, we present an algorithm that gives an optimal solution for production sequence S_{ab} with $I_b^* = 0$. In Section 2.2, we present an algorithm that gives an optimal solution for production sequence S_{aT} with $I_T^* > 0$. Finally, we summarize the overall algorithm in Section 2.3.

2.1 Case: S_{ab} with $I_b^* = 0$

Let us consider the case when S_{ab} ends with zero inventory. In this section, we show that an optimal solution has a certain structure. The overall algorithm enumerates all possible cases, and for each case the best solution is obtained. For constant MOQ and capacity, Okhrin and Richter [10] presented a similar structure.

We first describe some properties of a production sequence S_{ab} of an optimal solution x^*. Similar properties are presented for constant MOQ in [7, 8, 10], whereas the following two lemmas are for non-constant MOQ.

Lemma 1. For a period i such that $l_i < x_i^* \leq u$, we must have $x_t^* \in \{0, u\}$ for $t > i$.

Lemma 2. Let i be the first period such that $x_{\bar{t}}^* > l_{\bar{t}}$. Then, $x_t^* \in \{0, l_t\}$ for $t < i$.

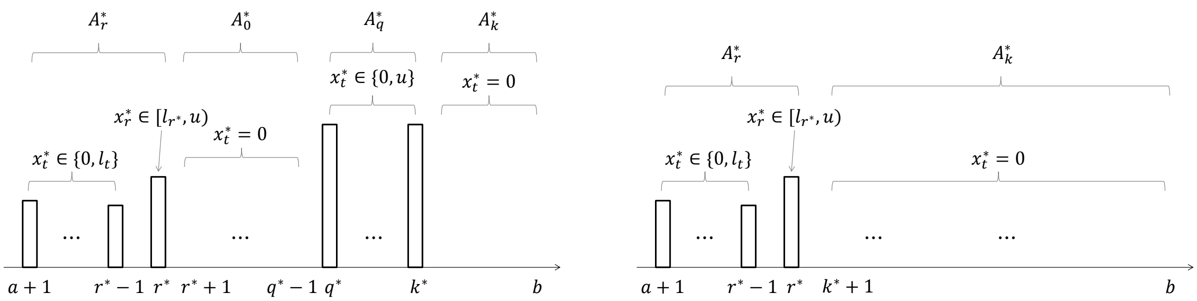
The proofs, which are given in Appendix A, are based on contradictions to the delayed optimal assumption. Based on Lemmas 1 and 2, we extend the result in [10] to non-constant and non-increasing MOQ. That is, S_{ab} can be decomposed into up to three phases based on the quantity produced by x^*:

1. periods producing either 0 or l_t,
2. a period with $l_t < x_t^* < u$, and
3. periods producing either 0 or u.

Let r^* be the last period with positive but strictly less than u production, q^* be the first period with production of u, and k^* be the last period with positive production. We also define n^* to be the number of periods such that $x_t^* = u$ for $a+1 \le t \le b$, or equivalently for $q^* \le t \le k^*$. We note that n^* is equivalent to K in [10] for lot sizing with constant MOQ. Based on r^*, q^*, and k^*, we decompose S_{ab} into four sub-sequences:

$$A_r^* = \{a+1, \cdots, r^*\}, \quad A_0^* = \{r^*+1, \cdots, q^*-1\}, \quad A_q^* = \{q^*, \cdots, k^*\}, \quad A_k^* = \{k^*+1, \cdots, b\}.$$

Figure (1a) illustrates these concepts. Observe that A_r^*, A_0^*, A_q^*, or A_k^* might not be defined depending on r^*, q^*, or k^*, as depicted in Figure (1b). Detailed cases for the existence of the sub-sequences and all possible combinations of (r^*, q^*, k^*) are available in Appendix C.



(a) x^* with A_r^*, A_0^*, A_q^*, and A_k^*

(b) x^* only with A_r^* and A_k^*

Figure 1: Sub-sequences of S_{ab} defined by x^*

Note that we know the existence of (r^*, q^*, k^*, n^*) but we cannot derive them explicitly. To find this unknown tuple (r^*, q^*, k^*, n^*), we consider all possible choices of (r, q, k, n). Given a solution x, let A_r, A_0, A_q, and A_k be the partition of the periods in S_{ab} based on r, q, and k, defined similar to A_r^*, A_0^*, A_q^*, and A_k^*, respectively. That is, $A_r = \{a+1, \cdots, r\}$, $A_0 = \{r+1, \cdots, q-1\}$, $A_q = \{q, \cdots, k\}$, and $A_k = \{k+1, \cdots, b\}$, in which the existence of each set depends on r, q, and k. Let n be the number of periods such that $x_t = u$ for $t \in A_q$.

Let us assume that we are given a tuple (r, q, k, n). Note that A_r, A_0, A_q, and A_k require further structure on demand in addition to $d_t \le u$ for all t, which is assured by the execution of *DemandForward*. Hence, we modify the demand further by Algorithm 1. The main principle of Algorithm 1 hinges on the same arguments that let us assume $d_t \le u_t$. Using this principle with $l_t, 0, u$, and 0 as upper bounds for the demand in A_r, A_0, A_q, and A_k, respectively, we modify the demand while we also consider the fact that exactly nu units are produced in A_q. In summary, the modified demand $\bar{d}$ after Algorithm 1 satisfies

upper bound requirement: $\bar{d}_t \le l_t$ for $t \in A_r$, $\bar{d}_t = 0$ for $t \in A_0 \cup A_k$, $\bar{d}_t \le u$ for $t \in A_q$,
same total demand: $\sum_{t \in A_r} \bar{d}_t + \sum_{t \in A_q} \bar{d}_t = \bar{d}_{a+1,r} + \bar{d}_{q,k} = \bar{d}_{a+1,b} = d_{a+1,b}$,
total demand in A_q: $\sum_{t \in A_q} \bar{d}_t = \bar{d}_{q,k} = nu$,

while not affecting the solution space with respect to the original demand d given (r, q, k, n). In detail, in Step 0, we first copy d to $\bar{d}$ and check if the tuple (r, q, k, n) can give a feasible solution. The algorithm returns a null set if the tuple is not valid. In Step 1 (for A_k), we forward demand $\bar{d}_{k+1,b}$ to period k. In Step 2 (for A_q), the algorithm returns a null set since $\bar{d}_{qk} < nu$ implies positive inventory. Otherwise, we forward the extra demand $\bar{d}_{qk} - nu$ to period r. In Step 4 (for A_0), we forward $\bar{d}_{a+1,q-1}$ to period r. In Step 4 (for A_r), if the tuple is valid, we forward the demand to satisfy $\bar{d}_r \le u$ and $\bar{d}_t \le l_t$ for $t \in A_r \setminus \{r\}$.

Example. Let us illustrate Algorithm 1 with an example. Consider production sequence S_{ab} with $d = \{4, 2, 3, 4, 11, 12\}$, $r = 3, q = 5, k = 5$, and $n = 1$. For simplicity, let us assume constant lower and upper bounds $l = 7$ and $u = 12$. Note that we have $A_r = \{1, 2, 3\}$, $A_0 = \{4\}$, $A_q = \{5\}$, and $A_k = \{6\}$. See Figure 2 for the illustration of each step. In this illustration, we omit the calculation for returning a null set, as all of the conditions are not satisfied.

Step 1 We forward the demand in A_k to period k. Hence, $\bar{d}_5 := \bar{d}_5 + \bar{d}_6 = 11 + 12 = 23$ and $\bar{d}_6 := 0$. Since $\bar{d}_5 > u$, we apply *DemandForward* and obtain $\bar{d} = \{4, 2, 6, 12, 12, 0\}$.

Algorithm 1 PreGREEDY(d, r, q, k, n)

Input: d (original demand), (r, q, k, n) (defining A_r, A_0, A_q, and A_k)
Output: either (i) $\bar{d}$ (modified demand) or (ii) $\emptyset$

Step 0 $\bar{d} := d$, **if** $u + \sum_{t \in A_r \setminus \{r\}} l_t < d_{a+1,q-1}$ or $u(n+1) + \sum_{t \in A_r \setminus \{r\}} l_t < d_{a+1,b}$ **return** $\emptyset$

Step 1 $\bar{d}_k := \bar{d}_k + \bar{d}_{k+1,b}$, $\bar{d}_t := 0$ for $t \in A_k$, *DemandForward*$(\bar{d})$

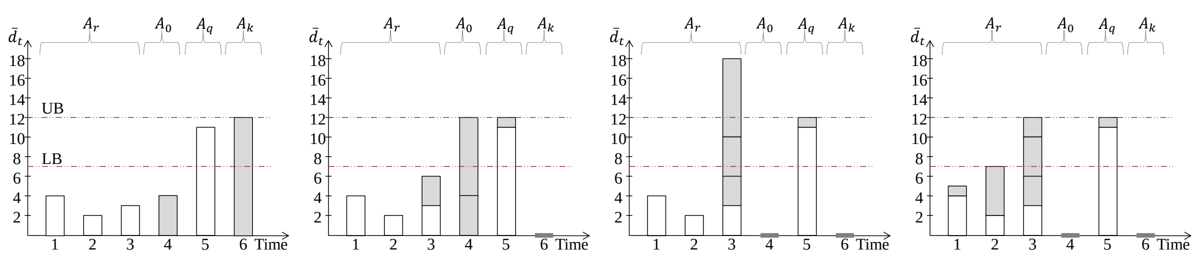
Step 2
  if $\bar{d}_{qk} < nu$ **return** $\emptyset$
  else if $\bar{d}_{qk} \geq nu$
    $\Delta := \bar{d}_{qk} - nu$, $t := q$
    while $\Delta > 0$
      $\delta := \min\{\Delta, \bar{d}_t\}$, $\bar{d}_r := \bar{d}_r + \delta$, $\bar{d}_t := \bar{d}_t - \delta$, $\Delta := \Delta - \delta$, $t := t + 1$
    end while
  end if

Step 3 $\bar{d}_r := \bar{d}_r + \bar{d}_{r+1,q-1}$, $\bar{d}_t := 0$ for $t \in A_0$.

Step 4
  if $\bar{d}_{a+1,r} > u + \sum_{t=a+1}^{r-1} l_t$ **return** $\emptyset$
  else
    $\bar{d}_{r-1} := \bar{d}_{r-1} + \max\{\bar{d}_r - u, 0\}$, $\bar{d}_r = \min\{u, \bar{d}_r\}$
    for $t = r-1, \cdots, a+2$, $\bar{d}_{t-1} := \bar{d}_{t-1} + \max\{\bar{d}_t - l_t, 0\}$, $\bar{d}_t = \min\{l_t, \bar{d}_t\}$
  end if
  if $\bar{d}_{a+1} > l_{a+1}$ **return** $\emptyset$

Step 2 We check if the total demand in A_q is equal to $nu = 12$. Since $\bar{d}_{qk} = \bar{d}_{5,5} = 12 \geq nu$, we set $\beta := 0$ and forward nothing. The modified demand remains $\bar{d} = \{4, 2, 6, 12, 12, 0\}$.
Step 3 We want to have $\bar{d}_t = 0$ for $t \in A_0$ and forward the demand in A_0 to period r. Hence, $\bar{d}_3 := \bar{d}_3 + \bar{d}_{4,4} = 6 + 12 = 18$ and $\bar{d}_4 := 0$. The modified demand is now $\bar{d} = \{4, 2, 18, 0, 12, 0\}$.
Step 4 We want $\bar{d}$ to satisfy $\bar{d}_r = \bar{d}_3 \leq u$ and $\bar{d}_t \leq l_t$ for all $t \in A_r \setminus \{r\} = \{1, 2\}$. Hence, we sequentially set (i) $\bar{d}_3 := \min\{\bar{d}_3, u\} = \min\{18, 12\} = 12$ and $\bar{d}_2 := \bar{d}_2 + (18 - 12) = 2 + 6 = 8$, (ii) $\bar{d}_2 := \min\{\bar{d}_2, l_2\} = \min\{8, 7\} = 7$ and $\bar{d}_1 := \bar{d}_1 + (8 - 7) = 4 + 1 = 5$, (iii) $\bar{d}_1 := \min\{\bar{d}_1, l_1\} = \min\{5, 7\} = 5$.

Hence, we obtain $\bar{d} = \{5, 7, 12, 0, 12, 0\}$ given $(r, q, k, n) = (3, 5, 5, 1)$. □



(a) Original demand (b) After Steps 1 and 2 (c) After Step 3 (d) After Step 4

Figure 2: Illustration of *PreGREEDY*

After obtaining the modified demand $\bar{d}$, we find the best solution for the given tuple (r, q, k, n), based on a greedy strategy. The algorithmic framework is presented in Algorithm 2. In Algorithm 2, Steps 1-7, 8, 9-12, and 13 tackle A_r, A_0, A_q, and A_k, respectively. In detail, for periods in A_0 and A_k (in Steps 8 and 13), the production is set to zero to satisfy the property of A_0 and A_k. For periods in A_r (in Steps 1-7), the algorithm is based on a greedy strategy. Starting from $t = a + 1$ up to r, the production x_t is set to l_t only when there is not enough inventory to cover the demand in period t. Observe that we check if $l_r \leq \bar{d}_r - \bar{I}_{r-1}$ in Step 6.

This is because given tuple (r, q, k, n) cannot produce a solution for S_{ab} if $l_r > \bar{d}_r - \bar{I}_{r-1}$. See Lemma 18 in Appendix B for the proof. For periods in A_q in Steps 9-12, the algorithm is based on a backward strategy. Starting from $t = k$ down to q, the production x_t is set to u only when the productions up to period k cannot cover $\bar{d}_{t,k}$.

Algorithm 2 GREEDY$(\bar{d}, r, q, k, n)$

Input: $\bar{d}$ (modified demand), (r, q, k, n) (defining A_r, A_0, A_q, and A_k)
Output: either (i) (x, z) (solution and objective value) or (ii) $\emptyset$

1: $x_{a+1} \leftarrow \max\{l_{a+1}, \bar{d}_{a+1}\}$, $\bar{I}_{a+1} \leftarrow x_{a+1} - \bar{d}_{a+1}$
2: **for** $t = a + 2$ **to** $r - 1$ **do**
3:    **if** $\bar{I}_{t-1} \geq \bar{d}_t$ **then** $x_t \leftarrow 0$, $\bar{I}_t \leftarrow \bar{I}_{t-1} - \bar{d}_t$
4:    **else** $x_t \leftarrow l_t$, $\bar{I}_t \leftarrow \bar{I}_{t-1} + l_t - \bar{d}_t$
5: **end for**
6: **if** $l_r \leq \bar{d}_r - \bar{I}_{r-1}$ **then** $x_r \leftarrow \bar{d}_r - \bar{I}_{r-1}$, $\bar{I}_r \leftarrow \bar{I}_{r-1} + x_r - \bar{d}_r$
7: **else return** $\emptyset$
8: $x_t \leftarrow 0$, $\bar{I}_t \leftarrow \bar{I}_r$ for $t = r + 1, \cdots, q - 1$
9: $m \leftarrow 0$ (counter)
10: **for** $t = k$ **down to** q **do**
11:    **if** $\bar{d}_{tk} > u \cdot m$ **then** $x_t \leftarrow u$, $m \leftarrow m + 1$
12: **end for**
13: $x_t \leftarrow 0$ for $t = k + 1, \cdots, b$, update $\bar{I}$

Example (continued). Let us illustrate *GREEDY* with the previous example. We are given $A_r = \{1, 2, 3\}$, $A_0 = \{4\}$, $A_q = \{5\}$, $A_k = \{6\}$, and the modified demand $\bar{d} = \{5, 7, 12, 0, 12, 0\}$.

1. For period 1 of A_r in Step 1, $x_1 := \max\{l_1, \bar{d}_1\} = 7$ and $\bar{I}_1 = x_1 - \bar{d}_1 = 2$.
2. For period 2 of A_r, we check $\bar{I}_1 = 2 < 7 = \bar{d}_2$ in Step 3. Hence, in Step 4, $x_2 := l_2 = 7$ and $\bar{I}_2 := \bar{I}_1 + x_2 - \bar{d}_2 = 2$.
3. For period 3 of A_r, we check $l_3 = 7 \leq \bar{d}_3 - \bar{I}_2 = 12 - 2 = 10$ in Step 6. Hence, $x_3 := \bar{d}_3 - \bar{I}_2 = 12 - 2 = 10$ and $\bar{I}_3 := \bar{I}_2 + x_3 - \bar{d}_3 = 2 + 10 - 12 = 0$.
4. For period 4 of A_0 in Step 8, we set $x_4 := 0$ and $\bar{I}_4 := \bar{I}_3 + x_4 - \bar{d}_4 = 0$.
5. For period 5 of A_q in Step 9, we set $x_5 := u = 12$ and $\bar{I}_5 := \bar{I}_4 + x_5 - \bar{d}_5 = 0 + 12 - 12 = 0$.
6. For period 6 of A_k in Step 13, we set $x_6 := 0$ and $\bar{I}_6 := \bar{I}_5 + x_6 - \bar{d}_6 = 0$.

Hence, we obtain $x = \{7, 7, 10, 0, 12, 0\}$ and $\bar{I} = \{2, 2, 0, 0, 0, 0\}$ from *GREEDY* with $\bar{d} = \{5, 7, 12, 0, 12, 0\}$, $r = 3, q = 5, k = 5$, and $n = 1$. Observe that the inventory based on the original demand d is $I = \{3, 8, 15, 11, 12, 0\}$. □

In the following two lemmas, we establish feasibility of the solution produced by *GREEDY* and then show that *GREEDY* with optimal parameters (r^*, q^*, k^*, n^*) produces an optimal solution, where the proofs are available in Appendix A.

Lemma 3. Suppose that $\bar{d}$ is from Algorithm 1 and $l_r \leq \bar{d}_r - \bar{I}_{r-1}$ is ensured in Algorithm 2. Then, given (r, q, k, n) and $\bar{d}$, Algorithm 2 produces a feasible solution.

Lemma 4. Let x^* and $\bar{I}^*$ be an optimal solution and the corresponding inventory with underlying values $\bar{d}$ and (r^*, q^*, k^*, n^*). Then, $GREEDY(\bar{d}, r^*, q^*, k^*, n^*)$ produces an optimal solution.

Example (continued). Let us consider *GREEDY* with optimal parameters r^*, q^*, k^*, n^*. The optimal solution that satisfies the delayed optimal assumption is $x^* = (7, 0, 7, 0, 10, 12)$. Hence, we obtain $r^* = 5$, $q^* = 6$, $k^* = 6$, and $n^* = 1$. Let us consider $r = 5, q = 6, k = 6$, and $n = 1$. Note that the parameters fit Case 7 of (16) in Appendix C and we are given $A_r = \{1, 2, 3, 4, 5\}$, $A_0 = \emptyset$, $A_q = \{6\}$, and $A_k = \emptyset$. Note that $\bar{d} = (4, 2, 3, 4, 11, 12)$ is obtained by *PreGREEDY* with these parameters and is different from the previous example since before we used $r = 3, q = 5, k = 5$ and $n = 1$. By executing *GREEDY* with $(r, q, k, n) = (5, 6, 6, 1)$, we obtain $x = \{7, 0, 7, 0, 10, 12\}$, which is equivalent to x^*. ☐

The overall algorithm is presented in Algorithm 3, where we simply iterate through all possible combinations of (r, q, k, n). For each (r, q, k, n), we obtain a feasible solution x with objective function value z by *GREEDY* and update the best solution and objective function value during the algorithm. Observe that, by Algorithm 3, we consider all possible cases of (16) in Appendix C. The 'for' loop in Algorithm 3 is constructed in such a way that combinations of r, q, k, n not fitting into one of the eight cases are automatically discarded by the loop itself.

Algorithm 3 EnumerateMOQ

Input: $(d_a \cdots, d_b)$ (demand), S_{ab}
Output: x_{best} (best solution), z_{best} (best objective function value)

$x_{best} \leftarrow \emptyset$, $z_{best} \leftarrow \infty$
for $n \in [0, k-q+1]$, $q \in [r+1, k]$, $k \in [r, b]$, $r \in [a+1, b]$ **do**
  $\bar{d} \leftarrow PreGREEDY(d, r, q, k, n)$
  if $\bar{d}$ is returned successfully **then** $(x, z) \leftarrow GREEDY(\bar{d}, r, q, k, n)$
  if $x \neq \emptyset$ and $z < z_{best}$ **then** update x_{best} and z_{best}
end for

We are ready to show the optimality of the solution produced by Algorithm 3.

Lemma 5. For a production sequence S_{ab}, either (i) the optimal solution is $x_{a+1} = d_{a+1}$ if $d_{a+1} \in \{0\} \cup [l_{a+1}, u]$ and $a + 1 = b$; or (ii) the solution x_{best} from Algorithm 3 is an optimal solution.

Proof. The statement is trivial if $b = a + 1$ since there is only one period in this production sequence. Hence, for the rest of the proof, let us assume that $b > a+1$. Let x^* be an optimal solution. Observe that we can pick r, q, k and n such that $A_r = A_r^*$, $A_0 = A_0^*$, $A_q = A_q^*$, $A_k = A_k^*$, and $n = n^*$, since we enumerate all possible choices of r, q, k and n in Algorithm 3. Then, by Lemma 4, x_{best} is an optimal solution. ☐

2.2 Case: $I_T^* > 0$ for S_{aT}

Let us consider production sequence S_{aT} with $I_T^* > 0$. We start with a property of x^*, which is similar to the property described by Hellion *et al* [11] for constant MOQ.

Lemma 6. For production sequence S_{aT} with $I_T^* > 0$, we have $x_t^* \in \{0, l_t\}$ for $t \in S_{aT}$.

The proof is omitted and is similar to the proof of Lemma 2. Let $\bar{d}$ the modified demand obtained by executing *DemandForward* where l_t's are the upper bounds. Based on Lemma 6, we present a greedy algorithm that returns an optimal solution for production sequence S_{aT} with $I_T^* > 0$.

Lemma 7. Let us consider the following greedy algorithm: for $t = a+1, \cdots, T$, (i) $x_t = l_t$ if $\bar{I}_{t-1} < \bar{d}_t$, (ii) $x_t = 0$ otherwise. This algorithm gives an optimal solution to S_{aT} with $I_T > 0$.

Proof. By Lemma 6, we must have $x_t^* \in \{0, l_t\}$ for $t \in S_{aT}$. Observe that the structure is the same as the structure of A_r^* in Section 2.1. The same proof technique of Lemma 4 can be applied to show that the greedy algorithm produces an optimal solution. ☐

2.3 Summary of Overall Algorithm

In Sections 2.1 and 2.2, we developed the algorithms that find an optimal solution for S_{ab} with $I_b^* = 0$ and S_{aT} with $I_T^* > 0$. Now, the dynamic programming algorithm of Hellion *et al* [11] can be used to solve the overall problem with Algorithm 3 and Lemma 7 as subroutines. The modification from the conventional DP of Florian and Klein [3] is that for each S_{aT}, we have two types of nodes: one with $I_T^* = 0$ and the other one with $I_T^* > 0$. Using this approach, we can solve the overall problem optimally.

We end this section by deriving the run time analysis of the overall algorithm. For Algorithm 3, the complexity of the for loop for r, k, q, n is $O(T^4)$. Since the run times of *GREEDY* and *PreGREEDY* are linear in T, we conclude $O(T^5)$ for S_{ab} with $I_b^* = 0$. For the greedy algorithm in Lemma 7, it is easy to show that we have $O(T)$ steps. For the overall problem based on the DP, since we have $O(T^2)$ production sequences, the time complexity is $O(T^7)$. We summarize all of the findings of this section in the following theorem.

Theorem 1. Algorithm 3 and Lemma 7 provide a polynomial algorithm that finds an optimal solution for the capacitated lot sizing problem with non-increasing linear costs and non-increasing MOQ in $O(T^7)$ steps.

3 Linear Programming Extended Formulation

In this section we present LP extended formulations for the single item lot sizing problem with MOQ in presence of constant lower and upper bounds l and u. We also extend the result to the case when fixed cost is present. The reader is referred to the works in [13, 20] for LP extended formulations of other cases without MOQ. Our LP extended formulations for $S(l, u)$ and $S(l, \infty)$ are different from all known results as the previous works study $S(\emptyset, u)$.

We again employ the non-increasing cost structure, the demand assumption, and the delayed optimal assumption from Section 2 for an optimal solution. Let us define the quotient and the remainder of the division of u by l:

$k = \left\lfloor \frac{u}{l} \right\rfloor$ and $\varepsilon = u - kl$.

We study four special cases:

Case 1: $S(l, u)$ with $u = l$.
Case 2: $S(l, u)$ with $u = kl$, where $k \geq 2$.
Case 3: $S(l, u)$ with $u = kl + \varepsilon$, where $k \geq 2$ and $0 \leq \varepsilon < l$.
Case 4: $S(l, \infty)$

For each case, the complete formulation is derived as follows. Recall that lot sizing with MOQ can have an optimal solution with positive ending inventory. For S_{aT} with $I_T^* > 0$, we can use an explicit formula, proposed by Hellion *et al* [11], to calculate the optimal objective function value. For S_{ab} with $I_b^* = 0$, we formulate an LP. Then, using the structured shortest path techniques from [20], the LP can be extended into a larger LP that solves the entire problem. Since all four cases can be extended using the shortest path based formulation and S_{aT} with $I_T^* > 0$ case can be solved easily, in this section we focus on the LP formulation for production sequence S_{ab} with $I_b^* = 0$ for the four cases.

We start with Case 1. Suppose that $l = u$. In this case, any positive production is equal to l. Note that, if $d_{a+1,b}$ is not a multiple of l, we cannot have zero ending inventory at b. Hence, we assume $d_{a+1,b} \bmod l = 0$. For $t \in S_{ab}$, let y_t be a decision variable defined by $y_t = 1$ if production is positive in period t. Note that we have $\sum_{t=a+1}^{j} l y_t \geq d_{a+1,j}$ for period j in order to have non-negative inventory. Since l is constant and y_t's are binary, the constraint can be strengthened to $\sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil$. Hence, we formulate the following integer program for production sequence S_{ab}.

$$\beta = \min \quad l\Big(\sum_{t \in S_{ab}} p_t y_t\Big) \tag{1a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil, \qquad j \in S_{ab}, \tag{1b}$$

$$y_t \in \{0, 1\}, \qquad t \in S_{ab} \tag{1c}$$

It can be easily shown that the LP relaxation of (1) is integral, since the matrix of (1) is a lower triangle matrix and the right hand side of (1b) is integer. A rigorous argument is provided later in Lemma 25 for a more general case.

The rest of this section is organized as follows. In Section 3.1, we consider the case $u = kl$ which is then extended to $u = kl + \varepsilon$ in Section 3.2. In Section 3.3, we derive the LP extended formulation for $S(l, \infty)$. Finally, in Section 3.4, we consider fixed cost for all the LP extended formulations derived in this section.

3.1 Case: $S(l, u)$ with $u = kl$

Let us assume that u is a multiple of l. Suppose that we know the last period i such that the production is positive but strictly less than u. Then, we can decompose production sequence S_{ab} into two sub-sequences: $A_l = \{a+1, \cdots, i\}$ and $A_u = \{i+1, \cdots, b\}$. Note that $x_t \in \{0, l\}$ for $t \in A_l \setminus \{i\}$, $l \le x_i < u$, and $x_t \in \{0, u\}$ for $t \in A_u$. We first formulate an LP based on the assumption that we know period i. Later, we extend the LP with fixed i to a larger LP for production sequence S_{ab} by letting the model choose i.

Let us first derive the LP based on the assumption that we know period i. The main idea of the derivation of the LP given fixed i is as follows. First, we derive the amount of the demand that must be forwarded from A_u to A_l, by considering the fact that $x_t \in \{0, u\}$ for $t \in A_u$. Next, we calculate the fractional amount of the demand that cannot be covered by multiple l's in A_l, by considering the fact that $x_i \in \{0, l\}$ for $t \in A_l \setminus \{i\}$. Finally, the exact fractional production in period i is derived. Let us define

$$\bar{\delta}^i = d_{i+1,b} \bmod u. \tag{2}$$

If $\bar{\delta}^i = 0$, we need to produce u units prior to $i+1$ in order to make positive inventory at period i. If $\bar{\delta}^i > 0$, by delaying the production as late as possible, we need to have $\bar{\delta}^i$ units of inventory in period i. We show that x^* satisfies certain property based on the value of $\bar{\delta}^i$.

Lemma 8. An optimal solution x^* must satisfy

$$\textstyle\sum_{t \in A_u} x_t^* = \begin{cases} d_{i+1,b} - \bar{\delta}^i = \left\lfloor \frac{d_{i+1,b}}{u} \right\rfloor u & \text{if } \bar{\delta}^i > 0, \\ d_{i+1,b} - u = (\frac{d_{i+1,b}}{u} - 1)u & \text{if } \bar{\delta}^i = 0. \end{cases}$$

The proof is available in Appendix A. By Lemma 8, we forward $\bar{\delta}^i$ from A_u to A_l. Hence, $\bar{\delta}^i$ must be covered in A_l and $\bar{\delta}^i$ can be also interpreted as the amount of forwarded demand from A_u to A_l.

Next, let us consider the production up to period $i-1$. Recall that the production is either 0 or l up to period $i-1$. Hence, any feasible solution must produce at least $\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$ periods producing l up until period $i-1$. Let us define

$$\varphi^i = \left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil l - d_{a+1,i-1} = l - (d_{a+1,i-1} \bmod l) \tag{3}$$

to be the minimum inventory possible in period $i-1$ for $i > a+1$. If $i = a+1$, then $\varphi^i = 0$. Lemma 24 in Appendix A shows that φ^i is the optimal inventory in period $i-1$ if period i has infinite capacity.

For now, let us assume that period i has infinite capacity. We ensure the upper bound requirement for period i by a constraint later. Considering the definition of φ^i and Lemma 24, we can interpret $d_i + \bar{\delta}^i - \varphi^i$ as the exact amount to be produced in period i, if we forward $\bar{\delta}^i$. However, there are cases $\bar{\delta}^i$ in (2) may not be the exact quantity to be forwarded. Hence, we define, if $i < b$,

$$\delta^i = \begin{cases} \bar{\delta}^i & \text{if } \bar{\delta}^i > 0 \text{ and } d_i + \bar{\delta}^i - \varphi^i \ge l, \\ \bar{\delta}^i + u & \text{if } \bar{\delta}^i > 0 \text{ and } d_i + \bar{\delta}^i - \varphi^i < l, \\ u & \text{if } \bar{\delta}^i = 0, \end{cases} \tag{4}$$

to be the forwarded demand from A_u to A_l. If $i = b$, we set $\delta^i = 0$. The derivation of δ^i in (4) is available in Appendix D.

Next, let ρ^i be the fractional amount of demand that cannot be covered by multiple l's in A_l. Recall that δ^i units of demand has been forwarded to A_l. We define

$$\rho^i = (d_{a+1,i} + \delta^i) \bmod l. \tag{5}$$

For $t \in S_{ab}$, let y_t be a decision variable defined by

$$y_t = \begin{cases} 1 & \text{if there is positive production in } t \\ 0 & \text{otherwise.} \end{cases}$$

We derive the following constraints to satisfy the demand.

1. For $j \in A_l \setminus \{i\}$, we produce either 0 or l while total production by period j is greater than or equal to $d_{a+1,j}$. This can be written as $\sum_{t=a+1}^{j} l y_t \geq d_{a+1,j}$. Since l is constant and y_t's are binary, the constraint can be strengthened to $\sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil$.

2. In period i, we must satisfy $\sum_{t=a+1}^{i} x_t = d_{a+1,i} + \delta^i$. Recall that ρ^i is the amount that cannot be covered by multiple of l's. Hence, ρ^i must be covered in period i. Before we derive the constraint, we first strengthen the upper bound of x_i.

   (a) The upper bound of x_i can be strengthened from $u = kl$ to $(k-1)l + \rho^i$.

   (b) In period i, we must satisfy $\sum_{t=a+1}^{i} x_t = d_{a+1,i} + \delta^i$. Recall that ρ^i is the amount that cannot be covered by multiple of l's. Hence, ρ^i must be covered in period i. This implies $x_i \leq (k-1)l + \rho^i$. Then from $(\sum_{t=a+1}^{i-1} x_t) + x_i = d_{a+1,i} + \delta^i$, we derive $\sum_{t=a+1}^{i-1} x_t = d_{a+1,i} + \delta^i - x_i \geq d_{a+1,i} + \delta^i - \rho^i - (k-1)l$. Dividing by l, we obtain $\sum_{t=a+1}^{i-1} \frac{x_t}{l} \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - (k-1)$. Hence, we have $\sum_{t=a+1}^{i-1} y_t = \sum_{t=a+1}^{i-1} \frac{x_t}{l} \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - (k-1)$. Since we assume $y_i = 1$, we obtain $\sum_{t=a+1}^{i} y_t \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 2$.

3. For $j \in A_u$, we can derive $I_j = I_i + \sum_{t=i+1}^{j} x_t - d_{i+1,j} = \delta^i + \sum_{t=i+1}^{j} x_t - d_{i+1,j} \geq 0$ to ensure feasible inventory, since $I_i = \delta^i$. This gives $\sum_{t=i+1}^{j} x_t \geq d_{i+1,j} - \delta^i$. Dividing by u, we obtain $\sum_{t=i+1}^{j} y_t = \sum_{t=i+1}^{j} \frac{x_t}{u} \geq \frac{d_{i+1,j}-\delta^i}{u}$. Since u is constant and y_t's are binary, the constraint can be strengthened to $\sum_{t=i+1}^{j} y_t \geq \left\lceil \frac{d_{i+1,j}-\delta^i}{u} \right\rceil$.

Finally, we define

$$\lambda^i = d_{a+1,i} + \delta^i - \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i} + \delta^i - \rho^i}{l} - k + 1\}l \tag{6}$$

to be the amount of production in period i, where $\lambda^i \geq l$. It is worth to note that λ^i is correctly defined only for a valid choice of i.

Example. Let us consider production sequence $\{1, 2\}$ with $d = \{6, 3\}$, $l = 5$, and $u = 10$. If $i = 1$, then $\rho^1 = 4$, $\delta^1 = 3$, and $\lambda^1 = 9$. However, if $i = 2$, then $\rho^2 = 4$, $\delta^2 = 0$, and $\lambda^2 = -1$. Based on the definition of i, setting $i = 2$ implies that $x_1 \in \{0, 5\}$, which is infeasible since $d_1 = 6$. □

Observe that $d_{a+1,i}$ is the demand we must satisfy by the end of period i, since δ^i is forwarded from A_u. The maximum operator is the minimum inventory we need by the end of period $i-1$. In the maximum operator, the first term ensures the fulfillment of demand up until period $i-1$, while the second term defines the minimum quantity to satisfy the upper bound constraint in period i. Note also that the second term has $\frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1$ instead of $\frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 2$ since we are assuming $x_i \geq l$. Hence, λ^i is the amount we need to produce in period i.

With all the parameters and constraints derived, we obtain the following integer program.

$$\alpha^i = \min \quad l(\sum_{t \in A_l} p_t y_t) + p_i(\lambda^i - l) + u(\sum_{t \in A_u} p_t y_t) \tag{7a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil, \qquad j \in \{a+1, \cdots, i-1\}, \tag{7b}$$

$$\sum_{t=a+1}^{i} y_t \geq \frac{d_{a+1,i} + \delta^i - \rho^i}{l} - k + 2, \tag{7c}$$

$$\sum_{t=i+1}^{j} y_t \geq \left\lceil \frac{d_{i+1,j} - \delta^i}{u} \right\rceil, \qquad j \in \{i+1, \cdots, b\}, \tag{7d}$$

$$y_i = 1, \tag{7e}$$

$$y_t \in \{0,1\}, \qquad t \in S_{ab} \tag{7f}$$

Observe that period i appears twice in the objective function. The first term captures lp_i with $y_i = 1$ from (7e), while the second term calculates the cost of the additional production beyond l in period i.

In the following two lemmas, we first present that a feasible solution to (7) has a matching feasible solution to the original problem, where the proof is given in Appendix A. Then we show integrality of (7).

Lemma 9. Let y be a feasible solution to (7) with given i. Then, there exists a corresponding feasible solution x with the same objective function value. Further, for an optimal solution y^* to (7), the corresponding solution $\bar{x}^*$ and $\bar{I}^*$ satisfy $\bar{I}_i^* = \delta^i$ and $\bar{I}_b^* = 0$.

Lemma 10. The LP relaxation of (7) is integral.

Proof. Note that the RHS of (7b) and (7d) are integer. Also, by the definition of ρ^i in (5), the RHS of (7c) is integer. Finally, it can be shown that the matrix of (7) is totally unimodular. See Lemma 25 in Appendix B. Therefore, the LP relaxation of (7) is integral. □

We can extend the formulation using the shortest path network to solve the entire S_{ab} without the assumption that we know i. Let us redefine variables and sets. For $i \in S_{ab}$, let

$$z^i = \begin{cases} 1 & \text{if period } i \text{ is fractional,} \\ 0 & \text{otherwise,} \end{cases}$$

and, for $t \in S_{ab}, i \in S_{ab}$, let

$$y_t^i = \begin{cases} 1 & \text{if period } i \text{ is fractional and period } t \text{ has a positive production,} \\ 0 & \text{otherwise.} \end{cases}$$

In addition we define

$$A_l^i = \{a+1, \cdots, i\}, \text{ for } i \in S_{ab}, \text{ and } A_u^i = \{i+1, \cdots, b\}, \text{ for } i \in S_{ab}.$$

We extend the IP formulation (7) by adding constraint to select only one z^i and forcing y_t^i's to be zero for z^i's equal to zero. The integer program for production sequence S_{ab} is defined as follows.

$$\beta = \min \quad \sum_{i=a+1}^{b} [l(\sum_{t \in A_l^i} p_t y_t^i) + p_i(\lambda^i - l)z^i + u(\sum_{t \in A_u^i} p_t y_t^i)] \tag{8a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t^i \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil z^i, \qquad j \in \{a+1, \cdots, i-1\}, i \in S_{ab}, \tag{8b}$$

$$\sum_{t=a+1}^{i} y_t^i \geq \left(\frac{d_{a+1,i} + \delta^i - \rho^i}{l} - k + 2 \right) z^i, \qquad i \in S_{ab}, \tag{8c}$$

$$\sum_{t=i+1}^{j} y_t^i \geq \left\lceil \frac{d_{i+1,j} - \delta^i}{u} \right\rceil z^i, \qquad j \in \{i+1, \cdots, b\}, i \in S_{ab}, \tag{8d}$$

$$y_i^i = z^i, \qquad i \in S_{ab}, \tag{8e}$$

$$y_t^i \leq z^i, \qquad t \in S_{ab}, i \in S_{ab}, \tag{8f}$$

$$\sum_{i=a+1}^{b} z^i = 1, \tag{8g}$$

$$y_t^i \in \{0,1\}, \qquad t \in S_{ab}, i \in S_{ab}, \tag{8h}$$

$$z^i \in \{0,1\}, \qquad i \in S_{ab}, \tag{8i}$$

where the costs assiciated with period i with $\lambda_i < l$ is replaced with ∞.

Lemma 11. The LP relaxation of (8) solves production sequence S_{ab}.

The proof is given in Appendix A. As mentioned earlier in this section, we use the shortest path formulation given in [20] for the entire time horizon. With Lemma 11, it is easy to apply their model and extend the formulation for the entire time horizon $t = 1, \cdots, T$.

3.2 Case: $S(l, u)$ with $u = kl + \varepsilon$

In this section, we study the case $u = kl+\varepsilon$, where $\varepsilon > 0$ is allowed. The formulation and the other settings are almost identical to the model in Section 3.1, except for (7c) and (6) the constraint for period i and parameter λ^i, respectively. This is because the derivations of (7b) and (7d) are independent of whether u is multiple of l. For this reason, we describe only the changes from the model in Section 3.1.

We start with the constraint for period i. Note that we must satisfy $\sum_{t=a+1}^{i} x_t = d_{a+1,i} + \delta^i$. Recall that ρ^i is the amount that cannot be covered by multiple of l's and ρ^i must be covered in period i. Before we derive the constraint, we first strengthen the upper bound on x_i.

1. If $\varepsilon < \rho^i$, then the upper bound on x_i can be strengthened from $u = kl + \varepsilon$ to $(k-1)l + \rho^i$.
2. If $\varepsilon \geq \rho^i$, then the upper bound on x_i can be strengthened from $u = kl + \varepsilon$ to $kl + \rho^i$.

Hence, we consider the two cases $\varepsilon < \rho^i$ and $\varepsilon \geq \rho^i$ to derive the constraint for period i.

1. Case: $\varepsilon < \rho^i$
   From $(\sum_{t=a+1}^{i-1} x_t) + x_i = d_{a+1,i} + \delta^i$, we derive

   $$\textstyle\sum_{t=a+1}^{i-1} x_t = d_{a+1,i} + \delta^i - x_i \geq d_{a+1,i} + \delta^i - \rho^i - (k-1)l,$$

   where the inequality holds since $x_i \leq (k-1)l + \rho^i$. Dividing by l, we obtain $\sum_{t=a+1}^{i-1} \frac{x_t}{l} \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - (k-1)$. Hence, we have $\sum_{t=a+1}^{i-1} y_t = \sum_{t=a+1}^{i-1} \frac{x_t}{l} \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - (k-1)$. Since we assume $y_i = 1$, we obtain $\sum_{t=a+1}^{i} y_t \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 2$.

2. Case: $\varepsilon \geq \rho^i$
   We obtain $\sum_{t=a+1}^{i} y_t \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1$ by using the same approach as in the previous case except by using inequality $x_i \leq kl + \rho^i$ instead of $x_i \leq (k-1)l + \rho^i$.

Finally, we define

$$\lambda^i = \begin{cases} d_{a+1,i} + \delta^i - \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\}l & \text{if } \varepsilon < \rho^i, \\ d_{a+1,i} + \delta^i - \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k\}l & \text{if } \varepsilon \geq \rho^i, \end{cases} \tag{9}$$

to be the amount of production in period i, where $\lambda^i \geq l$. The main principle of this definition is the same as the one in Section 3.1. Hence, we obtain the following mathematical program.

$$\alpha^i = \min \quad l(\sum_{t \in A_l} p_t y_t) + p_i(\lambda^i - l) + u(\sum_{t \in A_u} p_t y_t) \tag{10a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil, \qquad j \in \{a+1, \cdots, i-1\}, \tag{10b}$$

$$\sum_{t=a+1}^{i} y_t \geq \frac{d_{a+1,i} + \delta^i - \rho^i}{l} - k + 1 + 1_{\{\varepsilon < \rho^i\}}, \tag{10c}$$

$$\sum_{t=i+1}^{j} y_t \geq \left\lceil \frac{d_{i+1,j} - \delta^i}{u} \right\rceil, \qquad j \in \{i+1, \cdots, b\}, \tag{10d}$$

$$y_i = 1, \tag{10e}$$

$$y_t \in \{0, 1\}, \qquad t \in S_{ab}, \tag{10f}$$

where indicator function $1_{\{\varepsilon<\rho^i\}}$ is defined to distinguish the two cases. For the feasibility of a solution to (10), we present the following analogous lemma to Lemma 9.

Lemma 12. Let y be a feasible solution to (10) with given i. Then, there exists a corresponding feasible solution x with the same objective function value. Further, for an optimal solution y^* to (10), the corresponding solution $\bar{x}^*$ and $\bar{I}^*$ satisfy $\bar{I}^*_i = \delta^i$ and $\bar{I}^*_b = 0$.

The proof is omitted and is similar to the proof of Lemma 9. Observe that (7) and (10) have exactly the same structure except (i) the right hand side of (7c) and (10c) and (ii) λ^i in the objective functions. Hence, using the results in Section 3.1, (10) can be extended to formulate an integer program for S_{ab} in the same way as (7) is extended to (8). Further, the LP relaxation of the new integer program solves production sequence S_{ab} by Lemma 11.

3.3 Case: $S(l, \infty)$

In this section, we present the LP extended formulation for $S(l, \infty)$, where $x_t \in \{0\} \cup [l, \infty)$. Recall that we only consider production sequence S_{ab} with $I^*_b = 0$. We start this section by describing some properties of an optimal solution x^* that are similar to Lemmas 1 and 2 in Section 2, where the proofs are omitted and are similar to the proofs of Lemmas 1 and 2.

Lemma 13. For a period i such that $x^*_i > l$, we must have $x^*_t \in \{0, l\}$ for $t < i$.

Lemma 14. For a period i such that $x^*_i > l$, we must have $x^*_t = 0$ for $t > i$.

From Lemmas 13 and 14, we observe that S_{ab} can be decomposed up to three phases:

1. periods producing either 0 or l,
2. one period with $x^*_t > l$, and
3. periods with no production.

Suppose now that we know the period such that $x^*_t > l$, and let i be such a period. If there is no such period, then we define i to be the last period with positive production. Hence, for x^*, we have either of the following cases:

1. $x^*_t \in \{0, l\}$ for $t < i$, $x^*_i > l$, $x^*_t = 0$ for $t > i$, or
2. $x^*_t \in \{0, l\}$ for $t < i$, $x^*_i = l$, $x^*_t = 0$ for $t > i$

We show two properties of x^* in the following two lemmas, where the proofs are given in Appendix A.

Lemma 15. Given period i such that $x^*_i > l$, we must have $\sum_{t=a+1}^{i-1} x^*_t = l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$. Further, $I^*_{i-1} = l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil - d_{a+1,i-1}$.

Lemma 16. Given period i such that $x^*_i > l$, we must have $x^*_i = d_{a+1,b} - l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$.

Formulations		Number of variables		Number of constraints	
Case	Reference	S_{ab}	Overall	S_{ab}	Overall
$S(l,u)$ with $l=u$	(1)	$O(T)$	$O(T^3)$	$O(T)$	$O(T^3)$
$S(l,u)$ with $l>u$	(8)	$O(T^2)$	$O(T^4)$	$O(T^2)$	$O(T^4)$
$S(l,\infty)$	(12)	$O(T^2)$	$O(T^4)$	$O(T^2)$	$O(T^4)$

Table 2: Size of the LP extended formulations

Let us define

$$\lambda^i = d_{a+1,b} - l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil \tag{11}$$

to be the amount of production in period i. Let y_t, for $t<i$, be a decision variable defined by

$$y_t = \begin{cases} 1 & \text{if there is positive production in } t, \\ 0 & \text{otherwise.} \end{cases}$$

For the constraints, we use the same principle from the previous sections for each period $j<i$. Hence, we obtain the following integer program.

$$\alpha^i = \min \quad l\left(\sum_{t=a+1}^{i} p_t y_t\right) + p_i \lambda^i \tag{12a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil, \qquad j \in \{a+1,\cdots,i-1\}, \tag{12b}$$

$$y_t \in \{0,1\}, \qquad t \in \{a+1,\cdots,i-1\} \tag{12c}$$

We present the feasibility of a solution to (12).

Lemma 17. Let y be a feasible solution to (12) with given i. Then, there exists a corresponding feasible solution x with the same objective function value.

The proof is omitted and is similar to the proof of Lemma 9. Observe that it can be easily shown that the LP relaxation of (12) is integral. Identical techniques to those in Section 3.1 can be used to obtain an LP formulation over the entire time horizon.

We end this section with a comparison of (12) and the previous formulations. Recall that (7) of $S(l,u)$ with $u=kl$, (10) of $S(l,u)$ with $u=kl+\varepsilon$, and (12) of $S(l,\infty)$ are comparable since all of them are assuming that we know period i. Observe that (7b), (10b), and (12b) have the same form. This is because all of them are related to the periods producing 0 or l. Let us compare (12) and (7) in detail. Recall that we assume $x_t=0$ for $t>i$ for $S(l,\infty)$ and $x_t\in\{0,u\}$ for $t>i$ for $S(l,u)$. For this reason, λ^i, the production quantity in period i, in (6) and (11) are defined differently. Hence, it is not trivial to obtain (12) from (7) by setting $u=\infty$ and by dropping (7c) $-$ (7e).

3.4 Fixed cost

We consider fixed cost for all of the previous results in this section. Let f_t be the fixed cost in period t for $t=1,\cdots,T$. If production is positive in period t, then cost f_t occurs regardless of and in addition to the quantity produced in the period. Recall that all the models presented in the previous sections rely on the fact that it is best to delay the production, due to the non-increasing cost. Hence, in order to use the same principle, we employ the following assumption on the fixed cost.

Non-increasing fixed cost $\quad f_1 \geq f_2 \geq \cdots \geq f_T$

Observe that, for all the models in this section, the parameters and constraints do not rely on the production cost $p_1,\cdots,p_T$ and the binary variables y_t's represent whether positive production occurs in each period.

Hence, we can easily include the fixed cost in the objective functions without modifying the parameters and constraints. The new objective functions for each case are as follows:

$$\begin{array}{lll} \min & \sum_{t\in S_{ab}}(f_t+lp_t)y_t & \text{for (1a),} \\ \min & \sum_{t\in A_l}(f_t+lp_t)y_t+[f_i+p_i(\lambda^i-l)]+\sum_{t\in A_u}(f_t+up_t)y_t & \text{for (7a) and (10a), and} \\ \min & \sum_{t=a+1}^{i}(f_t+lp_t)y_t+(f_i+p_i\lambda^i) & \text{for (12a).} \end{array}$$

4 Computational Experiment

In this section, we present a computational study of Algorithm 3 for $S(l_t,u)$ and the LP extended formulation for $S(l,u)$ with $k=2$. All experiments were performed on Intel Xeon X5660 2.80 GHz dual core server with 32 GB RAM, running Windows Server 2008 64 bit. All algorithms are implemented in C#, where the LP extended formation is solved by CPLEX.

To test the performance of the algorithms, we randomly generate 10 instances for each $T\in\{50,60,70\}$ for $S(l_t,u)$ and 10 instances for each $T\in\{20,30,40,50,60,70\}$ for $S(l,u)$. Hence, we have 30 instances for $S(l_t,u)$ and 60 instances for $S(l,u)$. The instance generation procedure is similar to those in [10, 11]. Given mean and standard deviation of the demand $\mu_d=100$ and $\sigma_d=60$, respectively, we generate $d_t\sim N(\mu_d,\sigma_d)$ for $t=1,\cdots,T$, while we make sure $d_t\geq 0$. In order to generate non-increasing cost, we set $p_t=\text{Round}\Big(5(10-\frac{t}{T})\Big)/5$ for $t=1,\cdots,T$. For MOQ and capacity of $S(l_t,u)$, we set $u=\mu_d+\sigma_d$ and $l_t=\mu_d-\text{Round}\Big(\frac{t\cdot 0.2\cdot\sigma_d}{T}\Big)$ for $t=1,\cdots,T$. For MOQ and capacity of $S(l,u)$ with $k=2$, we set $l=\mu_d$ and $u=2l$.

Due to the large size of the LP extended formulation, for $S(l,u)$ with $T\geq 50$, we only consider variables and constraints associated with production sequences S_{ab} with $|S_{ab}|\leq 10$. In Table 3, we present the execution times (in seconds) of (i) Algorithm 3 for $S(l_t,u)$ and (ii) Algorithm 3 and the LP extended formulation for $S(l,u)$ with $k=2$. For each instance class and each algorithm, we report the minimum, average, maximum execution times over 10 instances.

	$S(l_t,u)$			$S(l,u)$ with $u=2l$								
	Algorithm 3			Algorithm 3			LP			LP with S_{ab}, $\lvert S_{ab}\rvert\leq 10$		
T	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
20				0.1	0.1	0.2	1.1	1.4	4.2			
30				0.7	1.0	1.4	6.9	7.7	8.2			
40				3.5	4.7	6.7	39.9	43.3	48.9			
50	7.5	9.0	11.8	8.8	11.4	16.1				3.1	3.5	4.3
60	23.4	26.6	34.2	26.9	32.9	46.3				4.8	6.1	8.1
70	62.0	68.4	85.4	71.0	82.7	114.2				8.5	9.8	10.3

Table 3: Performances of Algorithm 3 and the LP extended formulation

The execution times of Algorithm 3 grow slower than its theoretical bound $O(T^7)$ for both $S(l_t,u)$ and $S(l,u)$ instances, where the execution times for $S(l_t,u)$ instances are smaller than those for $S(l_t,u)$ instances. This is because the production bounds $[l_t,u]$ are tighter for the generated $S(l_t,u)$ instances and this enables the algorithm to skip the calculation for several shortest path nodes. By comparing the execution times of Algorithm 3 and the LP extended formulation for $S(l,u)$, we observe that Algorithm 3 outperforms. However, if we solve the LP extended formulation only with production sequences S_{ab} with $|S_{ab}|\leq 10$, the execution times of the LP extended formulation reduce substantially. Even though we consider only production sequences with less than 10 time periods, the obtained solution was always optimal.

5 Conclusions

We identified the first polynomial case for the lot sizing problem with time dependent MOQ. On the practical side, the non-increasing cost and non-increasing MOQ often occurs in procurement contracts. Note that these non-increasing assumptions on cost and MOQ play important role in our proofs. However, a future research could consider to relax one of the assumptions.

We also proposed the first LP extended formulations for the lot sizing problem with the presence of MOQ requirement. The proposed formulations only work with constant MOQ and capacity and non-increasing cost assumptions. Hence, LP extended formulations with time dependent setting or non-ordered costs could be a future research direction.

6 Acknowledgment

We appreciate the helpful comments of the referees and an anonymous referee for finding a flaw in a proof.

References

[1] E. Musalem, R. Dekker, Controlling inventories in a supply chain: A case study, International Journal of Production Economics 93–94 (2005) 179–188.

[2] Y. Zhao, M. Katehakis, Controlling inventories in a supply chain: A case study, Probability in the Engineering and Informational Sciences 20 (2006) 257–270.

[3] M. Florian, M. Klein, Deterministic production planning with concave costs and capacity constraints, Management Science 18 (1971) 12–20.

[4] A. Manne, Programming of economic lot sizes, Management Science 4 (1958) 115–135.

[5] H. Wagner, T. Whitin, Dynamic version of economic lot sizing model, Management Science 5 (1958) 89–96.

[6] E. Anderson, B. Cheah, Capacitated lot-sizing with minimum batch sizes and setup times, Physica 30–31 (1993) 137–152.

[7] C.-Y. Lee, Inventory replenishment model: lot sizing versus just-in-time delivery, Operations Research Letters 32 (2004) 581–590.

[8] L. Li, C. Pun, D. Klabjan, Single item lot-sizing problem with minimum order quantity, Available from http://www.klabjan.dynresmanagement.com.

[9] I. Okhrin, K. Richter, The linear dynamic lot size problem with minimum order quantity, International Journal of Production Economics 133 (2011) 688–693.

[10] I. Okhrin, K. Richter, An $O(t^3)$ algorithm for the capacitated lot sizing problem with minimum order quantities, European Journal of Operational Research 211 (2011) 507–514.

[11] B. Hellion, F. Mangione, B. Penz, A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs, European Journal of Operational Research 222 (2012) 10–16.

[12] Y. Pochet, Valid inequalities and separation for capacitated economic lot-sizing, Operations Research Letters 7 (1988) 109–116.

[13] Y. Pochet, L. Wolsey, Polyhedra for lot-sizing with wagner-whitin costs, Mathematical Programming 18 (1994) 767–785.

[14] M. Constantino, Lower bounds in lot-sizing models: A polyhedral study, Mathmatics of Operations Research 23 (1998) 101–118.

[15] M. V. Vyve, Linear-programming extended formulations for the single-item lot-sizing problem with backlogging and constant capacity, Mathematical Programming 108 (2006) 53–77.

[16] S. Anily, N. Tzur, L. Wolsey, Multi-item lot-sizing with joint set-up costs, Mathematical Programming 119 (2009) 79–94.

[17] Y. Pochet, L. Wolsey, Single item lot-sizing with non-decreasing capacities, Mathematical Programming 42 (2010) 142–150.

[18] G. Angulo, S. Ahmed, S. Dey, Semi-continuous network flow problems, Mathematical Programming 145 (1-2) (2014) 565–599.

[19] Y. Pochet, L. Wolsey, Production planning by mixed integer programming, Springer, 2006.

[20] Y. Pochet, L. Wolsey, Lot-sizing with constant batches: Formulation and valid inequalities, Mathematics of Operations Research 67 (1993) 297–323.

APPENDIX

A Proof of Lemmas

Proof of Lemma 1

For a contradiction, let us assume that there exists a period j such that $l_j \leq x_j^* < u$ and $i < j$. Let us define $\delta = \min\{x_i^* - l_i, I_i^*, \cdots, I_{j-1}^*, u - x_j^*\} > 0$. Observe that (i) $x_i^* - \delta \geq l_i$, (ii) $x_j^* + \delta \leq u$, and (iii) $I_t^* - \delta \geq 0$ for $t = i, \cdots, j$. Hence, we can postpone δ units of production from period i to period j. This contradicts the delayed optimal assumption. □

Proof of Lemma 2

We have two cases.

1. Case: $l_i < x_i^* < u$
   For a contradiction, let us assume that there exists a period j such that $l_j < x_j^* = u$ and $j < i$. Let us define $\delta = \min\{x_j^* - l_j, I_j^*, \cdots, I_{i-1}^*, u - x_i^*\} > 0$. Observe that (i) $x_j^* - \delta \geq l_j$, (ii) $x_i^* + \delta \leq u$, and (iii) $I_t^* - \delta \geq 0$ for $t = j, \cdots, i-1$. Hence, we can postpone δ units of production from period j to period i. This contradicts the delayed optimal assumption.
2. Case: $x_i^* = u$
   For a contradiction, let us assume that there exists a period j before i such that $l_j < x_j \leq u$. This contradicts the selection of period i.

Therefore, we have $x_t^* \in \{0, l_t\}$ for $t < i$. □

Proof of Lemma 3

We need to show $x_t \in \{0\} \cup [l_t, u]$ and $\bar{I}_t \geq 0$ for all t. Note that we are given $\bar{I}_a \geq 0$. We use induction to show $\bar{I}_t \geq 0$ based on $\bar{I}_{t-1} \geq 0$.

1. For $t \in A_r \setminus \{r\}$, if $\bar{I}_{t-1} \geq \bar{d}_t$, then $\bar{I}_t = \bar{I}_{t-1} - \bar{d}_t \geq 0$. If $\bar{I}_{t-1} < \bar{d}_t$, then $\bar{I}_t = \bar{I}_{t-1} + l_t - \bar{d}_t \geq \bar{I}_{t-1}$ since $\bar{d}_t \leq l_t$. Hence, for both cases, $\bar{I}_t \geq 0$ for $t \in A_r \setminus \{r\}$. Also, clearly $x_t \in \{0, l_t\}$ for $t \in A_r \setminus \{r\}$.
2. For period r, we only consider the case $l_r \leq \bar{d}_r - \bar{I}_{r-1}$. It follows $x_r = \bar{d}_r - \bar{I}_{r-1} \geq l_r$, $x_r = \bar{d}_r - \bar{I}_{r-1} \leq \bar{d}_r \leq u$, and $\bar{I}_r = \bar{I}_{r-1} + x_r - \bar{d}_r = \bar{I}_{r-1} + (\bar{d}_r - \bar{I}_{r-1}) - \bar{d}_r = 0$.
3. For $t \in A_0$, we have $\bar{d}_t = 0$. Also, $\bar{I}_r = 0$. Hence, we have $\bar{I}_t = \bar{I}_{t-1} = 0$ and $x_t = 0$.
4. For $t \in A_q$, we have $\bar{d}_{t+1,k} \leq \sum_{i=t+1}^{k} x_i$ by Algorithm 2. Also, $\bar{I}_{q-1} \geq 0$. Then, for $t \in A_q$, we have $\bar{I}_t = \sum_{i=q}^{t} x_i - \bar{d}_{qt} = (nu - \bar{d}_{qk}) + \sum_{i=q}^{t} x_i - \bar{d}_{qt} = \sum_{i=t+1}^{k} x_i - \bar{d}_{t+1,k} \geq 0$. Also, $x_t \in \{0, u\}$ for $t \in A_q$.
5. For $t \in A_k$, we have $\bar{d}_t = 0$. Also, $\bar{I}_k = 0$, and we have $\bar{I}_t = \bar{I}_{t-1} = 0$ and $x_t = 0$.

We showed that $\bar{I}_t \geq 0$ for all t. Note that Algorithm 2 produces x satisfying $x_t \in \{0\} \cup [l_t, u]$ for all t. Hence, Algorithm 2 produces a feasible solution. □

Proof of Lemma 4

Let x and $\bar{I}$ be the solution and the corresponding inventory from Algorithm 2 with input parameters $\bar{d}, r^*, q^*, k^*$, and n^*. Note that we must have $x_t = x_t^*$ for $t \in A_0^* \cup A_k^*$ by the definition of A_0^*, A_k^*. Hence, it suffices to check the periods in A_r^* and A_q^*.

Let us first consider A_q^*. Observe that Steps 9-12 of Algorithm 2 postpone the production as late as possible while ensuring the inventories are non-negative. Observe also that x produces exactly n times in A_q^*. Hence, Algorithm 2 produces an optimal solution in A_q^*.

Let us next consider A_r^*. Let h be the number of periods such that x and x^* are different, and let $H = \{i_1, i_2, \cdots, i_h\}$ be the set of periods such that x and x^* are different. We consider several cases.

1. If $x_{i_1} = l_{i_1}$ and $x_{i_1}^* = 0$, then we have $\bar{I}_{i_1-1} < \bar{d}_{i_1}$ from Algorithm 2. Then, $\bar{I}_{i_1}^* = \bar{I}_{i_1-1}^* - \bar{d}_{i_1} = \bar{I}_{i_1-1} - \bar{d}_{i_1} < 0$ implies x^* is infeasible.
2. If $x_{i_1} = 0$ and $x_{i_1}^* = l_{i_1}$, we consider the case when $x_{i_2} = l_{i_2}$ and $x_{i_2}^* = 0$. We generate a new solution $\hat{x}$ such that (i) $\hat{x}_t = x_t^*$ for $t \neq i_1, i_2$, (ii) $\hat{x}_{i_1} = 0$, (iii) $\hat{x}_{i_2} = l_{i_1}$. Solution $\hat{x}$ is the same as x^* except that the production of x^* in period i_1 is postponed to period i_2. We have $\hat{x}_t = x_t$ for $t \leq i_2 - 1$, which implies $\hat{I}_t \geq 0$ for $t \leq i_2 - 1$. Further, since $\hat{I}_{i_2} = \bar{I}_{i_2}^*$ and $\hat{x}_t = x_t^*$ for $t > i_2$, we also have $\hat{I}_t = \bar{I}_t^*$ for $t \geq i_2$. Observe that $\hat{x}_{i_2} = l_{i_1} \in [l_{i_2}, u]$ since $l_{i_1} \geq l_{i_2}$. Therefore, $\hat{x}$ is a feasible solution and x^* is postponed. This contradicts the delayed optimal assumption. This case shows that $x_{i_2} = 0$ and $x_{i_2}^* = l_{i_2}$.

We next consider period i_3 with the given remaining case $x_{i_1} = x_{i_2} = 0$, $x_{i_1}^* = l_{i_1}$, and $x_{i_2}^* = l_{i_2}$. Consider the following procedure started with $c = 3$ (corresponding to i_c).

Step 1 Let $x_t = 0$ and $x_t^* = l_t$ for $t \in \{i_j | j = 1, \cdots, c-1\}$. If $c = h$, we terminate the procedure. Otherwise go to Step 2.
Step 2 If $x_{i_c} = l_{i_c}$ and $x_{i_c}^* = 0$, then we generate $\hat{x}$ by postponing the production of x^* in period i_{c-1} to period i_c.
Step 3 If $x_{i_c} = 0$ and $x_{i_c}^* = l_{i_c}$, we increase c by 1 and go to Step 1.

Observe that the above procedure asserts that $x_t = 0$ and $x_t^* = l_t$ for $t \in H \setminus \{i_h\}$. Next, we have the following sub-cases.

1. Let $i_h < r^*$.
   (a) Let $x_{i_h} = l_{i_h}$ and $x_{i_h}^* = 0$. In this case, we can postpone the production in period i_{h-1} to period i_h using the same technique as in the previous case when $x_{i1} = 0, x_{i1}^* = l_{i1}, x_{i1} = l_{i2}$, and $x_{i1}^* = 0$. This contradicts the delayed optimal assumption.
   (b) The remaining case is $x_{i_h} = 0$ and $x_{i_h}^* = l_{i_h}$. Observe that $x_t = 0$ and $x_t^* = l_t$ for $t \in H$, $x_t = x_t^*$ for $t \in A_r^* \setminus H$. By Lemma 19 in Appendix B, we must have $\bar{I}_r^* = 0$ and $\bar{I}_r = 0$ since now we are executing *GREEDY* with optimal parameters. However, since $x_t < x_t^*$ for $t \in A_r^* \setminus H$, we cannot have $\bar{I}_r = \bar{I}_r^*$. This contradicts Lemma 19.
2. Let $i_h = r^*$. Again, note that $x_t = 0$ and $x_t^* = l_t$ for $t \in H \setminus \{i_h\}$. In order to have the same inventory $\bar{I}_r^* = \bar{I}_r$, we must have $x_r > x_r^*$. Observe that $x_t \leq x_t^*$ for $t \leq r^* - 1$, and $x_r > x_r^*$. By Lemma 20 in Appendix B, x and x^* have the same cost in A_r^*.

Therefore, Algorithm 2 produces an optimal solution for A_r^*, and x has as low cost as x^*, and thus x is an optimal solution. □

Proof of Lemma 8

Note that $\sum_{t \in A_u} x_t^*$ must be a multiple of u by the definition of A_u. Hence, we can write $\sum_{t \in A_u} x_t^* = d_{i+1,b} - \bar{\delta}^i - qu$, where q is an integer. We have two cases.

1. Let us first consider the case $\bar{\delta}^i > 0$.
   (a) If $q = 0$, we have nothing to prove.

(b) If $q < 0$, then $\sum_{t\in A_u} x_t^* = d_{i+1,b} - \bar{\delta}^i + |qu| > d_{i+1,b}$ implies $I_b^* > 0$, which contradicts the assumption of a production sequence.

(c) If $q \geq 1$, then we have $I_i^* > qu = qkl$. Since $d_t \leq u$ for every t, we derive $\sum_{t\in A_u} x_t^* = d_{i+1,b} - \bar{\delta}^i - qu \leq (b-i-q)u - \bar{\delta}^i < (b-i)u$. This implies that x^* must have at least one period in A_u such that $x^* = 0$. Then, by Lemma 23 in Appendix B, x^* can be postponed. This contradicts the delayed optimal assumption.

2. Let us now consider the case $\bar{\delta}^i = 0$. Recall that $\sum_{t\in A_u} x_t^* = d_{i+1,b} - \bar{\delta}^i - qu = d_{i+1,b} - qu$ for this case.

   (a) If $q = 1$, we have nothing to prove.

   (b) If $q = 0$, then $\sum_{t\in A_u} x_t^* = d_{i+1,b}$ and $I_b^* = 0$ imply $I_i^* = 0$. Hence, S_{ab} can be broken into two sequences. This contradicts the definition of a production sequence.

   (c) If $q < 0$, then $\sum_{t\in A_u} x_t^* = d_{i+1,b} - qu = d_{i+1,b} + |qu| > d_{i+1,b}$ implies $I_b^* > 0$. This is a contradiction since S_{ab} would not be a production sequence.

   (d) If $q \geq 2$, since $d_t \leq u$ for every t, we can derive $\sum_{t\in A_u} x_t^* = d_{i+1,b} - qu \leq (b-i-q)u < (b-i)u$. This implies that x^* must have at least one period r in A_u such that $x_r^* = 0$. Then, by Lemma 23 in Appendix B, x^* can be postponed. This contradicts the delay assumption.

The above two cases end the proof. □

Proof of Lemma 9

Let us define x by

$$x_t = \begin{cases} ly_t & \text{for } t \in A_l \setminus \{i\} \\ \lambda^i & \text{for } t = i \\ uy_t & \text{for } t \in A_u. \end{cases}$$

The proof of the first statement consists of two parts: (i) upper and lower bound constraints of x and (ii) non-negative inventory over all periods.

1. For the upper and lower bound constraints, observe that x_t satisfies the constraint for all t, $t \neq i$. For x_i, we have to distinguish four cases based on (6).

   (a) Case: $\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil \leq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1$
   The upper bound of x_i is derived from

   $$x_i = d_{a+1,i} + \delta^i - l(\tfrac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1) = \rho^i + kl - l \leq l + kl - l = kl = u,$$

   where the inequality holds since $\rho^i \leq l$. The lower bound of x_i follows from

   $$x_i = d_{a+1,i} + \delta^i - l(\tfrac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1) = \rho^i + kl - l \geq kl - l = (k-1)l \geq l,$$

   where the two inequalities hold since $\rho^i \leq l$ and $k \geq 2$, respectively.

   (b) Case: $\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil > \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1$
   The upper bound of x_i is established by

   $$x_i = d_{a+1,i} + \delta^i - \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l < d_{a+1,i} + \delta^i - l(\tfrac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1) = (\rho^i - l) + kl \leq kl = u,$$

   where the last inequality holds since $\rho^i \leq l$. For the lower bound of x_i, we have two cases.

   i. If (i) $\bar{\delta}^i = 0$ or (ii) $\bar{\delta}^i > 0$ and $d_i + \bar{\delta}^i - \varphi^i < l$, then we have $\delta^i \geq u$. We also have

   $$x_i = d_{a+1,i} + \delta^i - \left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil l \geq d_{a+1,i-1} + \delta^i - \left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil l > -l + \delta^i \geq u - l = (k-1)l \geq l,$$

   where the first strict inequality holds by the property of the ceiling function.

ii. If $\bar{\delta}^i > 0$ and $d_i + \bar{\delta}^i - \varphi^i \geq l$, then $d_i + \delta^i - \varphi^i \geq l$. We derive

$$\begin{aligned} x_i &= d_{a+1,i} + \delta^i - \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l = d_{a+1,i-1} + d_i + \delta^i - \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l \\ &\geq d_{a+1,i-1} + \varphi^i + l - \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l \\ &= d_{a+1,i-1} + \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l - d_{a+1,i-1} + l - \left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil l = l, \end{aligned}$$

where the third equality (third line) holds by the definition of φ^i in (3).

Hence, we showed $l \leq x_i \leq u$ for all cases.

2. To establish non-negativity of inventory, we have three parts.

(a) For $j \in A_l \setminus \{i\}$, we have $\sum_{t=a+1}^{j} x_t = \sum_{t=a+1}^{j} l y_t \geq d_{a+1,j}$, where the inequality holds by (7b).

(b) For period i, let us consider (7b) for period $i-1$ and (7c). By plugging (7e) into (7c), we have

$$\textstyle\sum_{t=a+1}^{i-1} y_t \geq \left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil \quad \text{and} \quad \sum_{t=a+1}^{i-1} y_t \geq \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1.$$

Note that a feasible solution y must satisfy

$$\textstyle\sum_{t=a+1}^{i-1} y_t \geq \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\}.$$

By multiplying by l, we derive

$$\textstyle\sum_{t=a+1}^{i-1} x_t = \sum_{t=a+1}^{i-1} l y_t \geq \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\} l.$$

Hence, we obtain

$$\textstyle I_{i-1} = \sum_{t=a+1}^{i-1} x_t - d_{a+1,i-1} \geq \max\{\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil, \frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\} l - d_{a+1,i-1}.$$

Now, we derive

$$\begin{aligned} I_i &= I_{i-1} + x_i - d_i \\ &\geq \left[\max\{\left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil, \tfrac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\} l - d_{a+1,i-1}\right] \\ &\quad + \left[d_{a+1,i} + \delta^i - \max\{\left\lceil \tfrac{d_{a+1,i-1}}{l} \right\rceil, \tfrac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 1\} l\right] - d_i \\ &= d_{a+1,i} - d_{a+1,i-1} + \delta^i - d_i = \delta^i \geq 0. \end{aligned}$$

Hence, $I_i \geq \delta^i \geq 0$.

(c) Let us consider $j \in A_u$. Since we know $I_i \geq \delta^i$, we obtain $I_i + \sum_{t=i+1}^{j} x_t \geq \delta^i + \sum_{t=i+1}^{j} x_t = \delta^i + \sum_{t=i+1}^{j} u y_t \geq d_{a+1,j}$, where the last inequality holds by (7d).

Hence, any feasible y has a corresponding feasible solution x.

Let us next consider the second statement. From case 2(b) of this proof, we know $\bar{I}_i^* \geq \delta^i$. Since we want to show $\bar{I}_i^* = \delta^i$, let us assume $\bar{I}_i^* > \delta^i$ for a proof by contradiction. We derive

$$\begin{aligned} \bar{I}_b^* &= \textstyle\sum_{t \in A_l} \bar{x}_t^* + \sum_{t \in A_u} \bar{x}_t^* - d_{a+1,b} \\ &= \textstyle(\sum_{t \in A_l} \bar{x}_t^* - d_{a+1,i}) + (\sum_{t \in A_u} \bar{x}_t^* - d_{i+1,b}) \\ &= \textstyle\bar{I}_i^* + (\sum_{t \in A_u} \bar{x}_t^* - d_{i+1,b}) \\ &> \textstyle\delta^i + (\sum_{t \in A_u} \bar{x}_t^* - d_{i+1,b}) \\ &= 0. \end{aligned}$$

Note that $\bar{I}_b^* > 0$ contradicts the property of a production sequence. Also, we consider S_{aT} with $I_T^* = 0$. Hence, we must have $\bar{I}_i^* = \delta^i$. We can also show $\bar{I}_b^* = 0$ by changing the inequality to equality in the above derivation. □

Proof of Lemma 11

Observe that (8) can be decomposed into $b-a$ sub-problems given z^i's satisfying (8g). Let us define linear program $LP(\bar{z}^i)$ for given period i and $\bar{z}^i \in [0,1]$, and let $LP^*(\bar{z}^i)$ be the optimal objective function value of $LP(\bar{z}^i)$.

$$LP^*(\bar{z}^i) = \min \quad l(\sum_{t\in A_l^i} p_t y_t^i) + p_i(\lambda^i - l)\bar{z}^i + u(\sum_{t\in A_u^i} p_t y_t^i) \tag{13a}$$

$$s.t. \quad \sum_{t=a+1}^{j} y_t^i \geq \left\lceil \frac{d_{a+1,j}}{l} \right\rceil \bar{z}^i, \qquad j \in \{a+1,\cdots,i-1\}, \tag{13b}$$

$$\sum_{t=a+1}^{i} y_t^i \geq \left(\frac{d_{a+1,i}+\delta^i-\rho^i}{l} - k + 2 \right) \bar{z}^i, \tag{13c}$$

$$\sum_{t=i+1}^{j} y_t^i \geq \left\lceil \frac{d_{i+1,j}-\delta^i}{u} \right\rceil \bar{z}^i, \qquad j \in \{i+1,\cdots,b\}, \tag{13d}$$

$$y_i^i = \bar{z}^i, \qquad i \in S_{ab}, \tag{13e}$$

$$y_t^i \leq \bar{z}^i, \qquad t \in S_{ab}, i \in S_{ab}, \tag{13f}$$

$$0 \leq y_t^i \leq 1, \qquad t \in S_{ab}, i \in S_{ab} \tag{13g}$$

We first show that, given i and $\bar{z}_i \in [0,1]$, we have $LP^*(\bar{z}^i) = \alpha^i \bar{z}^i$. Let us consider $LP(\bar{z}^i)$ based on three different values of $\bar{z}^i$.

1. Case: $\bar{z}^i = 1$
   Observe that $LP(1)$ is equivalent to (7). Hence, $LP^*(1) = \alpha^i = \alpha^i \bar{z}^i$.
2. Case: $\bar{z}^i = 0$
   Due to (13f), we have $LP^*(0) = 0 = \alpha^i \bar{z}^i$.
3. Case: $0 < \bar{z}^i < 1$
   Let y^{*i} be an optimal solution to $LP(1)$. We claim that $\bar{y}^i = \alpha^i y^{*i}$ is an optimal solution to $LP(\bar{z}^i)$. For a contradiction, suppose that $\bar{y}^i$ is not an optimal solution and, instead, $\tilde{y}^i$ is an optimal solution to $LP(\bar{z}^i)$ with $\bar{y}^i \neq \tilde{y}^i$. Let us define $\hat{y}^i = \frac{\tilde{y}^i}{\alpha^i}$. By plugging $\tilde{y}^i = \alpha^i \hat{y}^i$ into (13), we obtain a problem equivalent to (7), and thus $\hat{y}^i$ is an optimal solution to $LP(1)$. Note that we have $y^{*i} = \frac{\bar{y}^i}{\alpha^i} \neq \frac{\tilde{y}^i}{\alpha^i} = \hat{y}^i$.
   (a) If $\tilde{y}^i$ has strictly greater optimal objective function value than $\bar{y}^i$, then $\hat{y}^i$ also has strictly greater optimal objective function value than y^{*i}. Hence, y^{*i} is not an optimal solution to $LP(1)$. This is a contradiction.
   (b) If $\tilde{y}^i$ and $\bar{y}^i$ have the same optimal objective function value, then both $\hat{y}^i$ and y^{*i} are optimal to $LP(1)$. Hence, $\bar{y}^i$ is an optimal solution to $LP(\bar{z}^i)$.

   We obtain $LP^*(\bar{z}^i) = \bar{z}^i LP^*(1) = \alpha^i \bar{z}^i$ for $0 < \bar{z}^i < 1$.

Hence, we conclude that $LP^*(\bar{z}^i) = \alpha^i \bar{z}^i$ for given i and any $\bar{z}^i \in [0,1]$.

Next, observe that the LP relaxation of (8) can be rewritten as

$$\min\{\sum_{i=a+1}^{b} LP^*(z^i)z^i : (8g), 0 \leq z^i \leq 1\}. \tag{14}$$

Further, since we have $LP^*(\bar{z}^i) = \alpha^i \bar{z}^i$, (14) is equivalent to

$$\min\{\sum_{i=a+1}^{b} \alpha^i z^i : \sum_{i=a+1}^{b} z^i = 1, 0 \leq z^i \leq 1\}, \tag{15}$$

which is clearly integral. Hence, the LP relaxation of (8) gives an optimal solution for production sequence S_{ab}. □

Proof of Lemma 15

Let us consider the first statement. For a contradiction, suppose $\sum_{t=a+1}^{i-1} x_t^* > l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$, or equivalently, $\sum_{t=a+1}^{i-1} x_t^* \geq l\left(1 + \left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil\right) = l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil + l$. Let r be the last period with positive production before i. Then, we know that $\sum_{t=a+1}^{i-1} x_t^* = \sum_{t=a+1}^{s} x_t^* \geq l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil + l$ for any period s such that $r \leq s \leq i-1$. Let $\bar{x}$ be the same solution as x^* except that $\bar{x}$ postpones l unites of production of x^* from period r to i. Then, for period s such that $r \leq s \leq i-1$,

$$\sum_{t=a+1}^{s} \bar{x}_t = -l + \sum_{t=a+1}^{s} x_t^* \geq l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil \geq l\left\lceil \frac{d_{a+1,s}}{l} \right\rceil$$

holds, which proves feasibility of $\bar{x}$. Hence, x^* is postponed and this contradicts the delayed optimal assumption.

The second statement can be derived from $I_{i-1}^* = \sum_{t=a+1}^{i-1} x_t^* - d_{a+1,i-1}$. $\square$

Proof of Lemma 16

From Lemma 15, we know that $\sum_{t=a+1}^{i-1} x_t^* = l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$. From Lemma 14, we know that $x_t^* = 0$ for $t > i$. Hence, we must have $x_i^* = d_{i,b} - I_{i-1}^* = d_{i,b} - \left(l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil - d_{a+1,i-1}\right) = d_{a+1,b} - l\left\lceil \frac{d_{a+1,i-1}}{l} \right\rceil$, where I_{i-1}^* is substituted from Lemma 15. $\square$

B Additional Lemmas

Lemma 18. If $l_r > \bar{d}_r - \bar{I}_{r-1}$ in Algorithm 2, then *GREEDY* cannot produce a solution for production sequence S_{ab}.

Proof. Note that $l_r > \bar{d}_r - \bar{I}_{r-1}$ implies that we must produce at least l_r in order to prevent negative inventory. Hence, we have $\bar{I}_r = \bar{I}_{r-1} + l_r - \bar{d}_r > 0$. However, since $\bar{d}$ from Algorithm 1 satisfies $\sum_{t \in A_q} \bar{d}_t = nu$ and $\bar{d}_t = 0$ for $t \in A_0 \cup A_k$, we know that $\bar{I}_r > 0$ implies $\bar{I}_b > 0$. Since the ending inventory $\bar{I}_b$ is positive, the algorithm fails to produce a solution for production sequence S_{ab}. $\square$

Lemma 19. Let x^* be an optimal solution with parameters $\bar{d}, r^*, q^*, k^*$, and n^*, and let $\bar{I}^*$ the corresponding inventory with the modified demand $\bar{d}$. Then, $\bar{I}_r^* = 0$, $\bar{I}_k^* = 0$, $\bar{I}_t^* = 0$ for $t \in A_0^* \cup A_k^*$.

Proof. Observe that we have $\sum_{t=a+1}^{b} d_t = \sum_{t=a+1}^{b} \bar{d}_t = \sum_{t=a+1}^{k} \bar{d}_t$ since there is no demand after period k for $\bar{d}$. Hence, we must have $\bar{I}_k^* = 0$. Since $x_t^* = 0$ for $t \in A_k^*$, $\bar{I}_k^* = 0$ implies $\bar{I}_k^* = \bar{I}_t^* = 0$ for $t \in A_k^*$. Similarly, since $\bar{d}_{qk} = n^* \cdot u = \sum_{t \in A_q^*} x_t^*$, we must have $\bar{I}_{q-1}^* = 0$. This implies $\bar{I}_r^*$ and $\bar{I}_t^* = 0$ for $t \in A_0^*$ since $x_t^* = 0$ for $t \in A_0^*$. Therefore, given $\bar{d}, r^*, q^*, k^*$, and n^*, we must have $\bar{I}_r^* = 0$, $\bar{I}_k^* = 0$, $\bar{I}_t^* = 0$ for $t \in A_0^* \cup A_k^*$.

Note that Lemma 19 does not contradict the requirement that inventory is positive in a production sequence for an optimal solution. This is because Lemma 19 is based on $\bar{d}$ while the production sequence requirement is with respect to d. $\square$

Lemma 20. Let $\tilde{x}$ and x be feasible solutions. If there exist two periods i and j such that (i) $\tilde{x}_t \geq x_t$ for $t \leq i$, (ii) $\tilde{x}_t = x_t$ for $i < t < j$, and (iii) $\tilde{x}_t \leq x_t$ for $t \geq j$, then $\tilde{x}$ cannot have lower cost than x.

Proof. Let $A = \{t | \tilde{x}_t > x_t, t \leq i\}$ and $B = \{t | \tilde{x}_t < x_t, t \geq j\}$. Let also $\varepsilon_t = \tilde{x}_t - x_t > 0$ for $t \in A$ and $\delta_t = x_t - \tilde{x}_t > 0$ for $t \in B$. Observe that, since we consider a production sequence, $\tilde{x}_t$ and x have same ending inventories of 0, which implies $\sum_{t \in A} \varepsilon_t = \sum_{t \in B} \delta_t$. Then, we derive

$$\begin{aligned}
\sum_{i \in S_{ab}} p_i \tilde{x}_i - \sum_{i \in S_{ab}} p_i x_i \quad &= \sum_{t \in A} p_t \varepsilon_t - \sum_{t \in B} p_t \delta_t \\
&\geq p_i \sum_{t \in A} \varepsilon_t - p_j \sum_{t \in B} \delta_t \quad (p_t \geq p_i \text{ for } t \leq i \text{ and } p_t \leq p_j \text{ for } t \geq j) \\
&= p_i \sum_{t \in A} \varepsilon_t - p_j \sum_{t \in A} \varepsilon_t \quad (\text{since } \textstyle\sum_{t \in A} \varepsilon_t = \sum_{t \in B} \delta_t) \\
&= (p_i - p_j) \sum_{t \in A} \varepsilon_t \\
&\geq 0.
\end{aligned}$$

Hence, $\tilde{x}$ cannot have lower cost than x. □

Lemma 21. Let h be a period in A_l such that $I^*_h > l$ and $x^*_h = 0$. Let $r < h$ be the latest period such that $x^*_r = l$ and $x^*_t = 0$ for $t = r+1, \cdots, h$. Let $\bar{x}$ be the same solution as x^* except $\bar{x}_r = 0$ and let $\bar{I}$ be the corresponding inventory. Then, $\bar{I}_t > 0$ for $t = a+1, \cdots, h$.

Proof. Since $h \in A_l$, we know that $I^*_h > l$ implies x^* has at least two periods with positive production up until period h. Hence, we have periods with positive production that can be reduced to zero.

1. For period $t \in \{a+1, \cdots, r-1\}$, observe that $\bar{I}_t = I^*_t$ since x^* and $\bar{x}$ are the same up until period $r-1$.
2. For period $t \in \{r, \cdots, h\}$, we know $I^*_r \geq I^*_{r+1} \geq \cdots \geq I^*_{h-1} \geq I^*_h > l$ since $I^*_h > l$ and $x^*_t = 0$. Then, we derive $\bar{I}_t = I^*_t - l \geq I^*_h - l > 0$ for $t = r, \cdots, h$.

Hence, $\bar{x}$ has positive inventories up to period h. □

Lemma 22. For $q \geq 1$, if $I^*_i > qkl$, then we can reduce qkl units of replenishments of x^* in A_l while maintaining non-negative inventory up to period i.

Proof. Let $\bar{x}$ be a solution, which initially is set to x^*. We alter $\bar{x}$ iteratively. We can first reduce the production of $\bar{x}$ in period i by setting $\bar{x}_i = l + (x^*_i \bmod l)$. Observe that, in order to reduce qkl units total, the remaining amount to be reduced is $qkl - (x^*_i - \bar{x}_i) \geq (q-1)kl + 2l$, where the inequality holds since $x^*_i - \bar{x}_i \leq (k-2)l$. Observe also that $\bar{I}_i > qkl - (x^*_i - \bar{x}_i) \geq (q-1)kl + 2l > l$. Hence, we satisfy the condition of Lemma 21 with $h = i$ and additional restriction $r < i$, and we can reduce the production by l at a period in $\{a+1, \cdots, i-1\}$. By iteratively applying Lemma 21 and continuously updating $\bar{x}$, we can reduce the production by l at each iteration until we have $0 \leq \bar{I}_i < l$. This implies that we can reduce qkl units of x^* in A_l with non-negative inventories up to period i. □

Lemma 23. For $q \geq 1$, suppose $I^*_i > qkl$. Then there cannot exist a period in A_u with no production.

Proof. Let us assume that there is a period in A_u with no production. As in the proof of Lemma 22, let $\bar{x}$ be a solution initially set to x^*. By Lemma 22, we can reduce up to qkl units of x^* in A_l while maintaining feasible inventories up to period i. Let us reduce kl of $\bar{x}$ in A_l and let B be the periods that are reduced, i.e., $B = \{t | 0 = \bar{x}_t < x^*_t = l, t \in A_l\}$. Let r be the earliest period that has zero production in A_u. Let us set $\bar{x}_r = kl$.

1. We have $\bar{I}_t \geq 0$ for $t = a+1, \cdots, i$ by Lemma 22.
2. Note that $x^*_t = u$ for $t = i+1, \cdots, r-1$ by the definition of r and A_u. Note also that, by the assumption, $d_t \leq u$. Hence, we have $0 \leq \bar{I}_i \leq \bar{I}_{i+1} \leq \bar{I}_{i+2} \leq \cdots \leq \bar{I}_{r-1}$ since $\bar{I}_i \geq 0$.
3. Observe that $\sum_{t=a+1}^{r} \bar{x}_t = \sum_{t=a+1}^{r} x^*_t$ since we postpone kl in B to period r. This implies $\bar{I}_t = I^*_t > 0$ for $t = r, \cdots, b$.

Therefore, $\bar{x}$ is feasible and x^* is postponed. This contradicts the delayed optimal assumption of x^*. □

Lemma 24. If period i has infinite capacity, and thus $d_i + \bar{\delta}^i$ can be covered in period i, then $I^*_{i-1} = \varphi^i$.

Proof. Suppose $I^*_{i-1} \neq \varphi^i$ for a contradiction. Since I^*_{i-1} must be at least φ^i, $I^*_{i-1} < \varphi^i$ implies that x^* is infeasible. Hence, let us assume $I^*_{i-1} > \varphi^i$. Since all productions in $A_l \setminus \{i\}$ are l, inequality $I^*_{i-1} > \varphi^i$ implies $I^*_{i-1} \geq \varphi^i + l$. Let r be the earliest period before $i-1$ such that $x^*_r > 0$ and $I^*_t \geq l$ for $t = r, \cdots, i-1$.

1. If such an r exists, we can postpone l from period r to $i-1$. Let $\bar{x}$ and $\bar{I}$ be the postponed new solution and the corresponding inventory. Then, it is easy to see that (i) $\bar{I}_t = I^*_t - l \geq 0$ for $t = r, \cdots, i-1$, and (ii) $\bar{x}_r = x^*_r + l = l$ or $2l$ since $x^*_r \in \{0, l\}$ for $r \in A_l \setminus \{i\}$, satisfying upper bound constraints. Hence, x^* can be postponed and this contradicts the delay assumption.
2. If r does not exist, we consider two cases.
   (a) Case: $x^*_{i-1} = l$
   We move l units from period $i-1$ to i. Since we assume period i has infinite capacity, x^* can be postponed. This contradicts the delayed optimal assumption.

(b) Case: $x^*_{i-1} = 0$
Let s be the last period before $i-1$ such that $x^*_s > 0$. Note that $I^*_s \geq I^*_{s+1} \geq \cdots \geq I^*_{i-1} \geq \varphi^i + l$ since $x^*_t = 0$ for $t = s+1, \cdots, i-1$ and $d_t \leq l$ for $t = s+1, \cdots, i-1$ by the demand assumption. Let us generate $\bar{x}$ by postponing l units from period s to $i-1$. Then, it is easy to see that $\bar{I}_t = I^*_t - l \geq 0$ for $t = s, \cdots, i-2$ and $\bar{I}_{i-1} = I^*_{i-1} \geq 0$. Hence, x^* can be postponed and this contradicts the delay assumption.

Hence, we must have $I^*_{i-1} = \varphi^i$. □

Lemma 25. The matrix of (7) is totally unimodular.

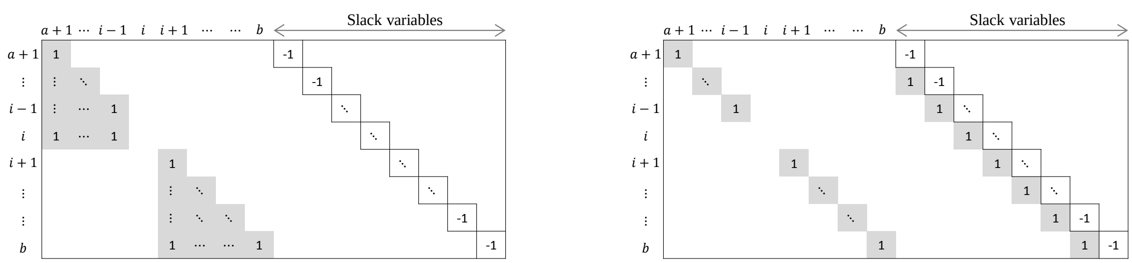
Proof. Let us add slack variables $s_{a+1}, \cdots, s_b$ to (7). By (7e), we have $y_i = 1$. Plugging this into (7c), we obtain an equation system with $(b-a)$ rows and $2(b-a)$ columns. Let A be the corresponding $(b-a)$ by $2(b-a)$ matrix. In other words, A is the augmented form of the matrix of (7) with slack variables. A has the structure depicted in Figure 3a. The gray, lined, and the empty cells represent the elements with $1, -1$, and 0, respectively. Observe that each row of A represents a period. Let us generate matrix B by the following elementary row operations on A:

1. new j^{th} row := j^{th} row - $(j-1)^{th}$ row, for $j = a+2, \cdots, i$
2. new j^{th} row := j^{th} row - $(j-1)^{th}$ row, for $j = i+2, \cdots, b$.

This yields matrix B with the structure presented in Figure 3b. To generate A from B, we execute

1. new j^{th} row := j^{th} row + $(j-1)^{th}$ row, for $j = a+2, \cdots, b-1$ such that $j \neq i+1$.

Since A and B can be generated from each other only by elementary row operations, matrices A and B are row equivalent. By a well-known result, B is totally unimodular since it contains no more than one 1 and no more than one -1 in each column. Therefore, A is totally unimodular.



(a) Augmented matrix of (7)

(b) Row equivalent matrix of (7)

Figure 3: Structure of the matrices

□

C Possible Cases of A^*_r, A^*_0, A^*_q, and A^*_k

In this section, we present all possible cases of A^*_r, A^*_0, A^*_q, and A^*_k based on r^*, q^*, k^*. Observe that we have $r^* \leq q^* \leq k^*$ based on the definitions of r^*, q^*, k^*. Hence, we have

1. if $q^* = a+1$, then r^* is not defined and $A^*_r = A^*_0 = \emptyset$,
2. if $r^* + 1 = q^*$, then $A^*_0 = \emptyset$,
3. if $r^* = k^*$, then q^* is not defined and $A^*_0 = A^*_q = \emptyset$, and
4. if $k^* = b$, then $A^*_k = \emptyset$.

Considering the above cases together, we have the following all possible combination of (r^*, q^*, k^*) and the existence of A_r^*, A_0^*, A_q^*, and A_k^*.

$$\begin{array}{l} \text{Case 1. If } r^* = k^* < b \text{, then } A_r^* \text{ and } A_k^* \text{ are naturally defined.} \\ \text{Case 2. If } r^* = k^* = b \text{, then } A_r^* \text{ is defined.} \\ \text{Case 3. If } r^* < k^* < b,\ a+1 < q^* \text{, and } r^*+1 < q^* \text{, then } A_r^*, A_0^*, A_q^* \text{, and } A_k^* \text{ are defined.} \\ \text{Case 4. If } r^* < k^* < b,\ a+1 < q^* \text{, and } q^* = r^*+1 \text{ then } A_r^*, A_q^* \text{, and } A_k^* \text{ are defined.} \\ \text{Case 5. If } r^* < k^* < b \text{ and } a+1 = q^* \text{, then } A_q^* \text{, and } A_k^* \text{ are defined.} \\ \text{Case 6. If } k^* = b,\ a+1 < q^* \text{, and } r^*+1 < q^* \text{, then } A_r^*, A_0^* \text{, and } A_q^* \text{ are defined.} \\ \text{Case 7. If } k^* = b,\ a+1 < q^* \text{, and } q^* = r^*+1 \text{, then } A_r^* \text{ and } A_q^* \text{ are defined.} \\ \text{Case 8. If } k^* = b \text{ and } a+1 = q^* \text{, then } A_q^* \text{ is defined.} \end{array} \tag{16}$$

D Derivation of δ^i

In this section, we derive δ^i in (4). We have the following three cases.

1. If $\bar{\delta}^i > 0$ and $d_i + \bar{\delta}^i - \varphi^i \geq l$, then the actual amount needed in period i is greater than or equal to the lower bound l. Hence, we can satisfy the lower bound requirement. The upper bound requirement is assured by a constraint later. Therefore, in this case, we do not need to adjust the forwarded demand.
2. If $\bar{\delta}^i > 0$ and $d_i + \bar{\delta}^i - \varphi^i < l$, then positive production in period i implies $I_i^* = I_{i-1}^* + x_i^* - d_i \geq \varphi^i + x_i^* - d_i \geq \varphi^i + l - d_i > \bar{\delta}^i$. Note that $I_i^* > \bar{\delta}^i \geq 0$ implies $I_b^* = I_i^* + \sum_{t \in A_u} x_t - d_{i+1,b} = I_i^* - \bar{\delta}^i > 0$. Hence, we have positive ending inventory and S_{ab} is not a production sequence. To prevent this, we need to forward more demand from A_u to A_l. Since the productions of A_u are a multiple of u, we can only forward a multiple of u. Hence, we forward the minimum amount u to A_l. This increases the total amount of forwarded demand from $\bar{\delta}^i$ to $\bar{\delta}^i + u$.
3. If $\bar{\delta}^i = 0$, then we have $I_i^* = 0$ and this contradicts the definition of a production sequence. To prevent this, we forward u to A_l. Observe that we must have $d_i + u - \varphi^i \geq d_i + u - l \geq d_i + (k-1)l \geq (k-1)l \geq l$. Hence, forwarding u instead of $\bar{\delta}^i = 0$ satisfies the lower bound constraint.

Based on the arguments above, we define δ^i in (4). Note that, if $\bar{\delta}^i = 0$, then we have $d_i + \delta^i - \varphi^i = d_i + u - \varphi^i \geq l$ regardless of the relationship of $d_i + \bar{\delta}^i - \varphi^i$ with respect to l. For this reason, we do not have to have two cases for $\bar{\delta}^i = 0$.