Single Item Lot-Sizing Problem with Minimum Order Quantity

Abstract

The traditional lot-sizing problem is to find the least cost production lot-sizes in several time periods. We consider the lot-sizing model with both capacity constraints and minimum order quantity requirements. We first show that the lot-sizing problem with linear cost functions, general capacities and minimum order quantities is NP hard. We then show that the problem is polynomially solvable in presence of constant capacities and minimum order quantities over a finite planning horizon. We also identify a polynomially solvable case with general minimum order quantities and infinite capacities. In the case of general capacities with the minimum order quantities, and in the presence of linear holding, backlogging, procurement costs, and a possible fixed component, we exhibit a fully polynomial time approximation scheme.

1 Introduction

The economic lot-sizing problem is a well-known problem in which a certain amount of a product is produced to satisfy known and deterministic demands over a finite planning horizon. The traditional lot-sizing problem is defined for a system with a single product, a fixed ordering cost, and a per item production and holding cost. The problem is to determine the production plan that satisfies all of the demand without backlogging while minimizing the total production and inventory holding costs. In the seminal work, Wagner and Whitin (1958) proposed a dynamic programming algorithm for solving the single-item uncapacitated lot-sizing problem.

Extensions to the Wagner and Whitin's lot-sizing problem have been studied since then. Zangwill (1966) made an early attempt to incorporate concave cost functions and to allow backlogging. He solved such a problem by formulating the problem as a network flow problem with concave arc costs. Love (1968) considered the problem with inventory constraints and developed a partial characterization of the structure of an optimal solution. The extension to allow capacity constraints is of significant importance as capacity is frequently encountered in practice. Florian and Klein (1971) studied the problem with constant capacities. They showed that it is polynomially solvable by exploiting the special structure of optimal production sequences. A dynamic programming algorithm was given, which runs in $O(T^4)$, where T is the number of time periods in the planning horizon. This complexity was later improved to $O(T^3)$ by van Hoesel and Wagelmans (1996). Pochet (1988) proposed a tight and compact linear extended formulation with $O(T^3)$ variables and constraints for this problem. When the capacities vary and are non-decreasing over time, the problem is also polynomially solvable under the conditions that the cost function is non-speculative and the set-up costs are non-decreasing overtime. A compact mixed integer programming reformulation whose linear relaxation solves this problem was proposed by Pochet and Wolsey (2007). The problem with general capacities is much more difficult. Such a problem with fixed set-up cost has been proven to be NP-hard by Florian et al. (1980).

Similar to the capacity requirement, the minimum order quantity (MOQ) restriction, which requires that the amount being produced has to be at least a certainty quantity, if produced at all, is also widely used in many industries. According to www.sticky-marketing.net, MOQ's are used in the presence of limitations, such as, to the production of a single item or where handling does not allow sales of very small or unitary number of items. MOQ also applies to production lines with limitations on machines. Moreover, MOQ's are often encountered in supply chains to enforce economies of scale due to high set-up costs associated with production or transportation processes. Snyder (1974) pointed out that when lot-sizes are small substantial savings are possible using MOQ's. Examples of MOQ's exist widely in everyday businesses in many industries, e.g., apparel, food, electric. In the case study by Fisher and Raman (1996), the company Sport Obermeyer requires a minimum ordering quantity per order of their garments. The manufacturer of gift items in Musalem and Dekker (2005) needs a specific chip to produce and the MOQ quantity for that chip is 10,000 items. The Logoers Company specifies on its website http://www.thelogoers.com MOQ's for their overseas products such as bags, clothing, and novelties. According to Zhao and Katehakis (2006), Home-Depot and Wal-Mart also have to honor the MOQ's specified by their suppliers.

We propose an interesting version of the lot-sizing problem with a very practical aspect. It considers time-dependent MOQs, which are required by many procurement contracts. We also identify two polynomially solvable cases. The approach in one case is unique to MOQ's. In addition, we develop an FPTAS for the MOQ problem with backlogging. MOQ's together with backlogs pose a significant challenge as existing approaches cannot be extended in a straightforward way.

There are several studies which explicitly consider MOQ. Anderson and Cheah (1993) proposed a multi-item capacitated lot-sizing problem with both setup times, and lower and upper bounds on the production level. They decomposed the problem into many single-item subproblems by applying Lagrangian relaxation on the capacity constraints. Constantino (1998) provided a polyhedral study of the multi-item MOQ. Mercé and Fontan (2003) considers MOQ with setups. They developed an MIP-based algorithm and solved it over a rolling horizon. All these works study mathematical programming based heuristics in the multi-item setting with MOQ. Our focus is on developing polynomially solvable special cases and an FPTAS. Lee (2004) addressed the uncapacitated version of MOQ with stepwise production costs and constant lower bounds on the production level. He analyzed the optimal properties of the solution policy and proposed a polynomial time algorithm to solve the problem. Our polynomially solvable costs differ from the setting in Lee (2004) since in one case, we consider constant upper bounds, and in the other case, we allow non-constant lower bounds, but no upper bounds.

The rest of this paper is organized as follows. Section 2 presents the general mathematical formulation of the single-item multi-period lot-sizing problem with both MOQ requirements and capacity constraints. It also shows that this general MOQ lot-sizing problem is NP-hard. Section 3 studies two polynomially solvable cases. The first case deals with constant capacities and MOQ's. The other case considers general MOQ's and no capacities with a certain assumption on the cost and MOQ values. In both cases, we study the structure of optimal solutions and then develop polynomial algorithms based on the dynamic programming framework. Section 4 develops a fully polynomial approximation scheme for the general single-item capacitated lot-sizing problem with minimum order quantities and backlogging and with linear cost functions and a possible fixed procurement cost.

2 Problem Description and Complexity

In this section, we first formally describe the lot-sizing problem with MOQ. Then, we show that the problem is NP-hard even in the absence of fixed costs.

We assume that demand and production happen at the beginning of each period. For $i \in \{1, \ldots, T\}$, where T is the number of time periods, we let: $d_i \geq 0$ represent the known demand for the product in period i; $d_{i,j} := \sum_{t=i}^{j} d_t$ be the total demand from period i to period j; x_i be the amount produced in period i; I_i be the inventory that is carried over period i; $p_i()$ be the production cost function for period i; and $h_i()$ be the holding cost from period i to period $i+1$. Furthermore, the amount of product produced in period i must be either 0 or between l_i and u_i, i.e., $x_i \in [l_i, u_i] \cup \{0\}$. Clearly we can also choose not to produce at all. We also make the following typical assumptions. There is no initial inventory, $I_0 = 0$; the demand in each period must be satisfied in that period, i.e., backlogging or lost sales are not allowed, which is equivalent to $I_t = \sum_{i=1}^{t} x_i - d_{1,t} \geq 0$; $\sum_{i=1}^{t} u_i \geq d_{1,t}$ for $t = 1, \ldots, T$, which ensures the existence of a feasible solution; at last, the functions $p_i()$ and $h_i()$ are nondecreasing concave and $p_i(0) = h_i(0) = 0$. It is important to notice that with MOQ's requiring the inventory level at the end of the time horizon to be zero might yield an infeasible problem.

The problem is to minimize the total production and storage costs. The model reads

$$
\begin{aligned}
&\min \sum_{i=1}^{T} p_i(x_i) + \sum_{i=1}^{T} h_i(I_i). \\
&I_i = I_{i-1} + x_i - d_i && i = 1, \ldots, T \\
&I_i \geq 0 && i = 1, \ldots, T \\
&I_0 = 0 \\
&x_i \in [l_i, u_i] \cup \{0\} && i = 1, \ldots, T.
\end{aligned}
$$

Since we do not require $I_T = 0$, $h_T(I_T)$ can be viewed as a penalty cost of the last period if $I_T > 0$. The following NP completeness result can be shown by a rather standard reduction from the knapsack problem by setting $l_i = u_i$ for every i.

Proposition 1. *The lot-sizing problem with MOQ's is NP-hard even if p_i and h_i are linear for every i.*

The following section studies two special cases, which are shown to be polynomially solvable.

3 Two Polynomially Solvable Cases

We first discuss the constant MOQ and capacity case. This is followed by a different special case with general MOQ's and no capacities.

3.1 Constant MOQ's and Capacities

We call period t a regeneration point if $I_t = 0$. Let S_{uv}, called the production sequence, represent a subset of a feasible production plan between two consecutive regeneration points u and v. Thus a production sequence S_{uv} has $I_u = I_v = 0$ and $I_{u+1} > 0, I_{u+2} > 0, \ldots, I_{v-1} > 0$. Since I_T may not be equal to 0, for the last sequence S_{uT}, we require $I_T \geq 0$ (instead of $I_T = 0$). For notational convenience, this last sequence is still called a production sequence even if $I_T > 0$.

It is easy to see that any optimal production plan can be decomposed into a set of consecutive production sequences. Such an observation is the basis of the dynamic programming recursion in our algorithm. Let $F(t), t = 1, \ldots, T$ be the cost associated with an optimal production plan over periods $\{0, \ldots, t\}$. Given that $I_0 = 0$, we have:

$$F(v) = \min_{0 \leq u < v} \{F(u) + Z_{uv}\} \qquad v = 1, \ldots, T, \tag{1}$$

$$F(0) = 0. \tag{2}$$

Value Z_{uv} is the cost associated with an optimal production plan over production sequence S_{uv}. This can be regarded as a forward recursion, which is more appropriate in our case due to $I_T \geq 0$. Wagner and Whitin (1958), Zangwill (1966), and Florian and Klein (1971) also used this dynamic programming recursion formulation, either as a forward or backward recursion.

This dynamic programming recursion requires the cost computation of $T(T+1)/2$ production sequences. The structure of an optimal production plan for a production sequence is first studied as follows.

Claim 1. *For production sequence $S_{uv}, v < T$ and an optimal production plan, there exists at most one period $t, u+1 \leq t \leq v$ in which the production level x_t is strictly between l_t and u_t, and the production level in all other periods $i, u+1 \leq i \leq v, i \neq t$ is either 0, l_i, or u_i.*

Proof. Let I^*, x^* be an optimal solution for the production sequence in question. Let us assume we produce in periods $\{t_1, t_2, \ldots, t_z\}$, i.e., $l_{t_i} \leq x^*_{t_i} \leq u_{t_i}$ for $i = 1, \ldots, z$ and $x^*_i = 0$ for all other time periods. Finding an optimal production plan for the production sequence S_{uv} is equivalent to solving the minimum cost network flow problem with concave cost functions on the network given in Figure 1. The demand of the horizontally aligned nodes, which correspond to time periods, equals to d_i. The bottom source node S has a surplus of d_{uv}. The flow of the inventory arcs must be in $[0, \infty)$ with cost $h_i()$. The production arcs from the source node to the time period nodes must have flow within $[l_i, u_i]$ for $i = t_1, t_2, \ldots, t_z$ and within $[0, 0]$ for the remaining time periods. The cost of these arcs is $p_i()$. The optimal solution I^*, x^* represents a feasible and optimal flow in this network. Since the costs are concave, there exist an extreme point solution I, x (see, e.g., Bazaraa et al. (1993)). Extreme point network flow solutions are cycle free (see, e.g., Ahuja et al. (1993)), i.e., the arcs with $l_i < x_i < u_i, I_i > 0$ form a spanning tree. All other arcs have flow of either 0, l_i, or u_i.

By definition of a production sequence, $I_i > 0$ for every arc and thus all these arcs are in the spanning tree. Furthermore, let us suppose that there are two periods s_1 and s_2 with $l_{s_1} < x_{s_1} < u_{s_1}, l_{s_2} < x_{s_2} < u_{s_2}$. Thus the corresponding production arcs of s_1 and s_2 are also in the tree. Obviously, this is a contradiction since we have identified a cycle in the spanning tree. Therefore, there is at most one period i with $l_i < x_i < u_i$. □

Claim 2. *For the last production sequence S_{uT} there exists an optimal solution such that if $I_T > 0$, the production level in each period $i, u+1 \leq i \leq T$, is either 0 or l_i.*

Proof. Suppose this is not the case. Let us consider an optimal solution such that $I_T > 0$, and at least one period i in production sequence S_{uT} has $x_i > l_i$. Let t_0 be the last period in which the production level is larger than the MOQ. Let $t_1 < t_2 < \cdots < t_k$, $t_1 > t_0$, denote all of the following periods which produce at their MOQ's. Let $\alpha = \min\{x_{t_0} - l_{t_0}, I_{t_1}, \ldots, I_{t_k}, I_T\}$. Obviously, according to our choice we have $\alpha > 0$. Note also that since S_{uT} is a production sequence, we have $I_i > 0$ for every i. Let us consider a new production plan $\bar{x}$ defined by $\bar{x}_i = x_i$ if $i \neq t_0$, and $\bar{x}_i = x_i - \alpha$ if $i = t_0$. The corresponding inventory levels are denoted by $\bar{I}$. It is not difficult to see that $\bar{x}$ is also a feasible and optimal solution. Since x is an optimal solution, and $\bar{x}$ is obtained from x by simply reducing the production level at period t_0 by α, $\bar{x}$ is also optimal. (Note that the cost functions are non-decreasing.) We consider the following three cases.

Case 1. We have $\alpha = I_T$. In this case, $\bar{I}_T = 0$. And thus Claim 1 applies.

Case 2. We have $\alpha = I_{t_i}, 1 \leq i \leq k$. In this case, $\bar{I}_{t_i} = 0$, and S_{t_iT} becomes a production sequence where $I_T > 0$ and the production level in each period s, $t_i \leq s \leq T$, is either 0 or l_s. And thus Claim 2 applies.

Case 3. We have $\alpha = x_{t_0} - l_{t_0}$. In this case, $\bar{x}_{t_0} = l_{t_0}$. Thus we have identified an optimal solution with one more period which produces at the MOQ.

We can repeat the process to obtain a solution satisfying the statement. □

We have not yet used any assumptions of constant MOQ's and capacities. Claims 1 and 2 hold in general. Let us now assume that $l_i = L, u_i = U$ for every $i = 1, \ldots, T$. The framework of the algorithm is based on the dynamic programming recursion (1) and (2) which contains $T(T+1)/2$ production sequences. Therefore, our main concern is to find an optimal production plan for a production sequence S_{uv}. We consider two cases: $v < T$ and $v = T$.

Let first $v < T$. By definition we have $I_u = I_v = 0$. Let us consider all possible values of I_g, $g \in \{u+1, .., v-1\}$. Let there be M periods in which the production level is L, N periods in which the production level is at capacity U, and at most one period in which the production level is ε with $L < \varepsilon < U$. The production amount in all other periods is 0. We have $ML + NU + \varepsilon = d_{u+1,v}$, where $M \geq 0$, $N \geq 0$, and $0 \leq M + N \leq T$. Therefore, I_g takes one of the following two forms. If the production level of ε is not yet observed, we have $I_g = \bar{M}L + \bar{N}U - d_{u+1,g}$. If ε has already been used, we have $I_g = \sum_{i=u+1}^{g} x_i - d_{u+1,g} = \bar{M}L + \bar{N}U + \varepsilon - d_{u+1,g} = \bar{M}L + \bar{N}U + (d_{u+1,v} - ML - NU) - d_{u+1,g} = d_{g+1,v} - (M - \bar{M})L - (N - \bar{N})U$, where $0 \leq \bar{M} \leq M$, $0 \leq \bar{N} \leq N$, and $I_g > 0$. There are at most T possible values for both M and N, so together with $0 \leq M + N \leq T, 0 \leq \bar{M} \leq M, 0 \leq \bar{N} \leq N$, there are at most $T(T+1)/2 + 4T^2$ possible values for I_g. In other words, the number of all possible values for I_g is $O(T^2)$. We denote by K_g all possible inventory levels of a period $g \in \{u+1, \ldots, v-1\}$, and construct a network as in Figure 2. The dependency on g comes from the fact that $I_g > 0$. The nodes and arcs are defined as follows.

- **Nodes**: There are $v - u - 1$ layers of nodes between u and v, each one corresponding to the end of a time period. The nodes in each layer g correspond to K_g, which are labeled in Figure 2.. They represent all possible values of $I_g, g \in \{u+1, \ldots, v-1\}$. The source node at the utmost left corresponds to $I_u = 0$, and the sink node at the utmost right corresponds to $I_v = 0$.

- **Arcs**: Between any two layers j and $j+1$, $j = \{u, \ldots, v-1\}$, there is an arc from node $n_1 = (r_{k_1}, j)$, $r_{k_1} \in K_j$ to node $n_2 = (r_{k_2}, j+1)$, $r_{k_2} \in K_{j+1}$ if $r_{k_1} - r_{k_2} = d_{j+1}$ or $L \leq r_{k_2} + d_{j+1} - r_{k_1} \leq U$. The former case corresponds to no production in period d_{j+1}, and the cost of this arc is the holding cost $h_j(r_{k_1})$. The latter case means that there is production in

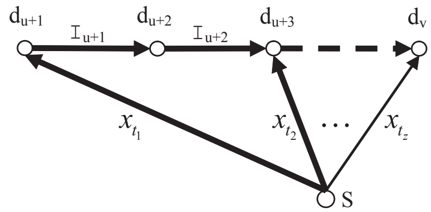

Figure 1: Minimum Cost Flow Network

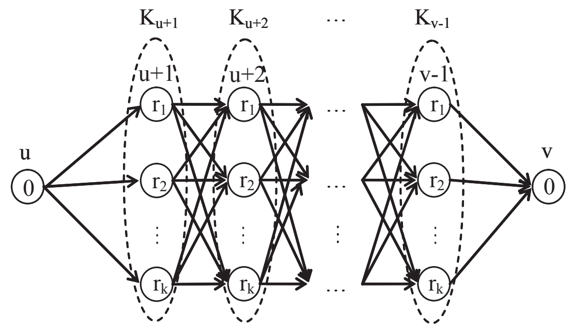

Figure 2: Network

period d_{j+1} and the cost of this arc is $h_j(r_{k_1}) + p_{j+1}(r_{k_2} + d_{j+1} - r_{k_1})$, which is the holding cost through period j and the corresponding production cost at the beginning of period $j+1$.

It is easy to see that the shortest path from node u to node v corresponds to an optimal production plan over the production sequence S_{uv}. There are at most $T+1$ layers and each layer has at most $O(T^2)$ nodes. Hence, there are $O(T^3)$ nodes and $O(T^5)$ arcs. Since the network is acyclic, the complexity of the shortest path algorithm is thus $O(T^5)$.

For the production sequence S_{uT}, there are two possibilities. The first is $I_T = 0$, which is no different from the previous case. The other case is $I_T > 0$. In this case, Claim 2 holds. Let us consider all possible values of I_g, $g \in \{u+1, .., T\}$. Let us assume that in an optimal production plan of S_{uT}, there are M periods in which the production level is L, and the production amount in all other periods is 0. Therefore, we have $I_g = \sum_{i=u+1}^{g} x_i - d_{u+1,g} = ML - d_{u+1,g}$. There are at most T possible values for M. We construct a network similar to Figure 2. However, since $I_T > 0$, the last period T has more than one possible inventory value. Thus, we need to add a pseudo period $T+1$ with value 0 at the end and connect it to every node in period T. The cost of an arc from node (r_k, T) to the pseudo node $T+1$ is $h_T(r_k)$. Then, the shortest path from u to $T+1$ corresponds to an optimal production plan over the production sequence S_{uT} when $I_T > 0$. The complexity of finding such a shortest path is $O(T^3)$. Therefore for production sequence S_{uT}, the shortest path over the two networks yields an optimal production plan with complexity $O(T^5)$.

The overall algorithm solves the recursion (1) and (2). The total computation time of this algorithm is $O(T^{5+2}) = O(T^7)$. Thus this algorithm is polynomial.

3.2 The Infinite Capacity Case

The problem studied in this section has general MOQ's but has no capacities. We exhibit a polynomial algorithm under certain assumptions. We require the following assumptions.

Assumption A1. *All cost functions are linear.*

Based on Assumption A1, the objective function can be restated as

$$\sum_{i=1}^{T} \hat{p}_i x_i + \sum_{i=1}^{T} \hat{h}_i I_i = \sum_{i=1}^{T} \hat{p}_i x_i + \sum_{i=1}^{T} \hat{h}_i (\sum_{j=1}^{i} x_i - d_{1,i}) = \sum_{i=1}^{T} (\hat{p}_i + \sum_{j=i}^{T} \hat{h}_j) x_i - \sum_{i=1}^{T} \hat{h}_i d_{1,i},$$

where $p_i(x_i) = \hat{p}_i x_i$ and $h_i(I_i) = \hat{h}_i I_i$. Let $p_i = \hat{p}_i + \sum_{j=i}^{T} \hat{h}_j$. The problem can now be regarded as a lot-sizing problem with linear production cost p_i and zero holding cost. It reads

$$\begin{aligned} &\min \sum_{i=1}^{T} p_i x_i \\ &\sum_{j=1}^{i} x_j \geq d_{1,i} \qquad i = 1, \ldots, T \\ &x_i \in [l_i, \infty) \cup \{0\} \qquad i = 1, \ldots, T. \end{aligned} \tag{3}$$

The next assumption imposes a relationship between p_i and l_i.

Assumption A2. *For the set of periods $\{t_k | k = 0, \cdots, n\}$ defined by*

$$t_0 = \max\{\operatorname{argmin}_{1 \leq i \leq T} p_i\}, \cdots, t_k = \max\{\operatorname{argmin}_{1 \leq i < t_{k-1}} p_i\}, \cdots, t_n = 1$$

we assume that (1) $l_{t_n} \geq l_{t_{n-1}} \geq \cdots \geq l_{t_0}$, and (2) $l_{t_k} \leq l_j$ for every j and k such that $k \in \{1, 2, \ldots, n\}$ and $t_k < j < t_{k-1}$, or $k = 0$ and $j > t_k$.

We give a simpler condition that implies Assumption A2 as follows.

Proposition 2. *Assumption A2 is fullfilled if $p_1 \geq p_2 \geq \cdots \geq p_T$ and $l_1 \geq l_2 \geq \cdots \geq l_T$.*

Proof. In this case, by definition we have $t_0 = T, t_1 = T - 1, \ldots, t_{n-1} = 2, t_n = 1$, i.e., the set $\{t_0, t_1, \ldots, t_n\}$ contains every time period in the problem. The first condition of Assumption A2 is satisfied since $l_1 \geq l_2 \geq \cdots \geq l_T$. The second condition is satisfied automatically since there is no such period j. □

We note that Proposition 2 requires non-increasing production costs as well as non-increasing MOQ's, which occurs often in practice. With these assumptions, we make the following observation.

Claim 3. *There exists an optimal solution x^* such that if $i \notin \{t_0, t_1, \ldots, t_k, \ldots, t_n\}$, then $x_i^* = 0$.*

Proof. Let x be an optimal solution with at least one period $i \notin \{t_0, t_1, \ldots, t_k, \ldots, t_n\}$ such that $x_i > 0$. For one such i, suppose $t_k < i < t_{k-1}$. Consider solution x^* such that $x_j^* = x_j$ if $j \neq i$ and $j \neq t_k$, $x_{t_k}^* = x_{t_k} + x_i$, and $x_i^* = 0$, which means that x^* is obtained by moving forward the production amount x_i from period i to period t_k. According to Assumption A2, x^* is a feasible solution because $x_{t_k}^* \geq x_i \geq l_i \geq l_{t_k}$. It is very easy to see that (3) holds for x^*. Furthermore, the cost of x^* is lower than or equal to the cost of x since $p_{t_k} \leq p_i$, which comes from the definition of t's. By repeating this for every such i, we obtain an optimal solution that satisfies Claim 3. □

Claim 3 states that in an optimal solution, the only possible periods in which we would produce are $\{t_0, t_1, \ldots, t_k, \ldots, t_n\}$. Note that $t_n < t_{n-1} < \cdots < t_0$. Based on this observation, we can adjust the demand as $\bar{d}_{t_i} = d_{t_i, t_{i-1}-1}$. For ease of notation, we relabel these periods as periods $1, 2, \ldots, T'$ to obtain the following equivalent problem:

$$\begin{aligned} &\min \sum_{i=1}^{T'} p_i x_i \\ &\sum_{j=1}^{i} x_j \geq \bar{d}_{1,i} \qquad i = 1, \ldots, T' \end{aligned} \tag{4}$$

$$x_i \in [l_i, \infty) \cup \{0\} \qquad i = 1, \dots, T',$$

where $p_1 \geq p_2 \geq \cdots \geq p_{T'} > 0$ and $l_1 \geq l_2 \geq \cdots \geq l_{T'}$.

This problem has infinite capacity, non-increasing linear production cost as well as non-increasing MOQ's. Next, we show that problem (4) is polynomially solvable by first observing a property of a production sequence, based on which the polynomial algorithm is developed.

Claim 4. *For a production sequence S_{uv}, either $u+1 = v$, $d_{u+1} = 0$ and thus the optimal solution is $x_{u+1} = 0$; or there exists a k, $u+1 \leq k \leq v$, and an optimal solution x such that $x_{k+1} = x_{k+2} = \cdots = x_v = 0$ and for time periods $u+1, u+2, \dots, k$ solution x follows the following greedy algorithm:*

$x_{u+1} = \max\{l_{u+1}, d_{u+1}\}$, *and*

$$x_i = \begin{cases} 0 & \text{if } \sum_{j=u+1}^{i-1} x_j \geq \bar{d}_{u+1,i}, \text{i.e.}, I_{i-1} \geq \bar{d}_i, \\ \max\{l_i, \bar{d}_{u+1,i} - \sum_{j=u+1}^{i-1} x_j = \bar{d}_i - I_{i-1}\} & \text{otherwise}, \end{cases}$$

for $i = u+2, u+3, \dots, k$, where we redefine $\bar{d}_k$ to be $\bar{d}_{k,v}$.

Proof. The statement trivially holds if $v = u+1$ since there is only one period in this production sequence. Let x^* be an optimal solution to the production sequence S_{uv} and k the last period with $x_k^* > 0$. If $v > u+1$ and $k = u+1$, it is also trivial since the only production occurs in the first period. Let us thus assume that $v > u+1$ and $k > u+1$.

Let x be the solution according to the greedy algorithm stated in the claim. It is not difficult to see that x is a feasible solution. Let $i, u+1 \leq i \leq k$, be the first period in which $x_i^* \neq x_i$. Among all optimal solutions x^*, we select the one with i as large as possible. If no such i exists, there is nothing to show. Otherwise, we must have $x_i^* > x_i$. To see this, if $x_i = 0$, then $x_i^* > 0 = x_i$ since $x_i^* \neq x_i$. If $x_i > 0$, then $I_{i-1}^* = I_{i-1} < \bar{d}_i$. Consider first $x_i = \bar{d}_i - I_{i-1}$. Then $x_i^* \geq \bar{d}_{u+1,i} - \sum_{j=u+1}^{i-1} x_i^* = \bar{d}_{u+1,i} - \sum_{j=u+1}^{i-1} x_i = \bar{d}_i - I_{i-1} = x_i$. Let now $x_i = l_i$. If $x_i^* = 0$, then $I_i^* = I_{i-1}^* - \bar{d}_i = I_{i-1} - \bar{d}_i < 0$ yields a contradiction. Thus $x_i^* \geq x_i$. Next, consider $\varepsilon = x_i^* - x_i > 0$. We distinguish four cases.

Case 1. If $i = k$, then we can reduce x_k^* by ε to x_k. This new solution is feasible and it has lower cost than x^*, which is a contradiction.

Case 2. If $i < k$, then we find the next period j, $j > i, j \leq k$ with $x_j^* > 0$. Such a period must exist since $x_k^* > 0$. Since we deal with a production sequence, $I_{j-1}^* > 0$. If $\varepsilon \geq I_{j-1}^*$ and $x_i > 0$, we can reduce x_i^* by I_{j-1}^* and increase x_j^* by I_{j-1}^*. Note that $\varepsilon \geq I_{j-1}^*$ and $x_i > 0$ imply $x_i^* - I_{j-1} \geq x_i \geq l_i$, and $x_j^* > 0$ implies $x_{j(new)}^* > x_j^* \geq l_j$. Therefore we get a new feasible solution with one more production sequence (since the inventory in period $j-1$ is reduced to 0) and no higher production cost (since we delay the production of amount I_{j-1}^* from period i to period j). This breaks S_{uv} into two production sequences, contradicting our selection of x^*.

Case 3. If $i < k$, $\varepsilon \geq I_{j-1}^*$ and $x_i = 0$, where j is the period identified in Case 2, then we have $\varepsilon = x_i^* \geq I_{j-1}^*$. If $x_i^* - l_i \geq I_{j-1}^*$, we can move I_{j-1}^* units of production from i to j. This again results in a new feasible solution with one more production sequence and no higher production cost, which is a contradiction. Another possibility is that $x_i^* = l_i$. If this is the case, there are three subcases as follows. Let $q > i$ be the smallest time period such that $x_q > 0$.

(1) If there is no such q, it means that no production is needed from period $i-1$ onwards. Thus reducing x_i^* to 0 gives a new feasible solution $x_{(new)}^*$ of no higher cost.

(2) If $j \le q$, we move the production of x_i^* units from period i to j. This is a new feasible solution since $x_{j(new)}^* > x_j^* \ge l_j$ and $I_{j-1(new)}^* = I_{j-1} \ge 0$. Note that $I_{j-1(new)}^* = I_{j-1}$ because $x_{(new)}^*$ agrees with x from period $u+1$ to at least period $j-1$, and the last inequality follows from feasibility of x.

(3) If $j > q$, we move the production of x_i^* units from period i to q. This is feasible since from $l_i \ge l_q$ we obtain $x_{q(new)}^* = l_i \ge l_q$ and $I_{q-1(new)}^* = I_{q-1} \ge 0$. Note that $I_{q-1(new)}^* = I_{q-1}$ because $x_{(new)}^*$ agrees with x from period $u+1$ to at least period $q-1$, and the last inequality follows from feasibility of x.

In all these three subcases, we have identified a new feasible solution $x_{(new)}^*$ with no higher production cost and one more period where the greedy solution x and the new optimal solution match, i.e., $x_{i(new)}^* = x_i$, which contradicts the selection of x^*.

The last possibility in Case 3 is $0 < x_i^* - l_i < I_{j-1}^*$. In this case we can obtain a new feasible solution $x_{(new)}^*$ with no higher cost by moving the production of $x_i^* - l_i$ units from period i to j. This results in $x_{i(new)}^* = l_i, x_i = 0$, which again falls into the above three subcases.

Case 4. If $i < k$, and $\varepsilon < I_{j-1}^*$, where j is the period identified in Case 2, we reduce x_i^* by ε and increase x_j^* by ε. Again we have identified a new feasible solution without increasing the cost by postponing some production to a later period. The fact that $x_{i(new)}^* = x_i$ contradicts the choice of x^*.

This completes the proof. □

Note that for the last production sequence S_{uT}, the inventory at the end of period T may not be zero due to the MOQ requirements. It is not difficult to see that Claim 4 and its proof also apply in this case.

We can now develop a polynomial algorithm using the same dynamic programming framework (1) and (2). There are $T(T+1)/2$ production sequences to consider. For each production sequence S_{uv}, we use the algorithm described in Claim 4 to find a feasible solution for each possible value of k, $k = u+1, u+2, \ldots, v$ and then we choose the lowest cost one to obtain Z_{uv}. It is not difficult to see that the computation time of this polynomial algorithm is $O(T^4)$.

4 Fully Polynomial Approximation Scheme

We have shown in Section 2 that the lot-sizing problem with general MOQ requirements is NP-hard. In this section, we exhibit an FPTAS for a slightly more general version that allows backlogging.

Kovalyov (1996) proposed a rounding technique to construct an FPTAS for a generalized knapsack problem. His method is applicable to problems with discrete domain sets, but is limited to nondecreasing cost functions. His algorithm is inapplicable to our problem, since we consider both production and backlog costs, and thus, having production at a lower level does not generally guarantee a lower overall cost. Furthermore, his algorithm is developed for a single constraint. van Hoesel and Wagelmans (2001) developed an FPTAS for the model with monotone concave cost functions, but without MOQ's. They showed that their FPTAS is rather general, and can be applied to many cases. Chubanov et al. (2006) generalized van Hoesel and Wagelmans (2001) by relaxing the concavity requirement. Their algorithm relies on the special structure of the underlying recursive functions to achieve fully polynomial running time without changing the feasible domain of the problem. They use scaling to obtain the desirable approximation ratio. No MOQ's are considered. Chubanov et al. (2008) generalized the single-item capacitated lot-sizing problem to the case of a non-uniform resource usage for production. A similar problem was discussed but it does not yield an FPTAS for our problem. The running time depends on N, which in our case is pseudopolynomial in the input size. Halman et al. (2008) proposed a completely different FPTAS for certain inventory problems. Their approach is more general since they allow inventory shortages and stochastic demand distributions, but requires a convex procurement cost function. They do not address the minimum order requirement, and such an extension is non-trivial. Chauhan et al. (2005) study the knapsack problem with MOQ's. Their approach cannot handle backlogging since in this case the state space of the underlying dynamic program becomes multidimensional without an efficient way to trim it. In addition, in presence of backlogging, the objective function is not a separable function of production quantities. Ng et al. (2008) extend the result by Chauhan et al. (2005) by dropping some of the assumptions. Since they apply the same framework, it is not applicable to our problem for the same reason.

Both approaches from Chubanov et al. (2006) and van Hoesel and Wagelmans (2001) are inapplicable to our problem without major modifications. Most technical results of Chubanov et al. (2006) rely on decrementing the order quantity, which may be prohibited by the MOQ requirement. Also, it may not be true that non-stable points of the recursive cost functions are sufficient to guarantee the FPTAS result when the MOQ requirement is imposed. Halman et al. (2008) cannot handle MOQ's and, in addition, the approach breaks in the presence of fixed procurement costs. The approach of van Hoesel and Wagelmans (2001) breaks down when any inventory level at a period may no longer be able to freely attend any value less than or equal to the maximum inventory level at the period given a limited budget and MOQ requirements. This property is crucial in their treatment. While the result of Chubanov et al. (2006) is more general than the result of van Hoesel and Wagelmans (2001), we focus on the methodology of van Hoesel and Wagelmans (2001) to develop a new FPTAS that is necessary to address the MOQ requirements. Although the high level ideas of our FPTAS are identical to the framework by van Hoesel and Wagelmans (2001), the difference between the two approaches is significant.

4.1 Preliminaries

Our approach to obtain the FPTAS is to first reformulate and simplify the model by assuming linear costs as in Section 3.2, and then to construct a "dual" problem. By doing so, we are able to develop a pseudo-polynomial dynamic programming algorithm whose running time only depends on the selected upper bound on the optimal objective value. In each time period in the planning horizon, the dual problem is to maximize the production subject to a budget allowance.

Beginning with the problem described in Section 3.2 with the addition of backlogging, we stress that the underlying assumption is to have linear cost functions. We show in Section 4.4 that the inclusion of fixed procurement costs does not require significant changes. Formally, we consider

$$z^* = \min_{\substack{x_i \in [l_i, u_i] \cup \{0\} \;\; i=1,2,\dots,T \\ y_i \geq 0 \;\; i=1,2,\dots,T}} \left\{ \sum_{t=1}^{T} (p_t x_t + b_t y_t) \,\middle|\, \sum_{t=1}^{i} x_t + y_i \geq d_{1,i} \qquad i = 1, 2, \dots, T \right\}$$

where y_i is the backlogging amount in time period i, and b_i the backlogging cost in the same time period. We note that $y_i = (d_{1,i} - \sum_{j=1}^{i} x_j)^+$ for every i, and we do not require demand to be satisfied at the end.

With this model at hand, we develop the dual problem for the dynamic programming algorithm. The concept of the dual problem is to maximize the total production given an allowable budget $b \in \{0, 1, 2, \dots, B\}$ for the first $t \in \{1, 2, \dots, T\}$ periods. We let B be an integer upper bound on z^*, and let $F_t(b)$ be the maximum value of the production given budget b. We have

$$F_t(b) = \max_{\substack{x_i \in [l_i, u_i] \cup \{0\} \;\; i=1,\ldots,t \\ y_i \geq 0 \;\; i=1,\ldots,t}} \left\{ \sum_{i=1}^{t} x_i \,\middle|\, \sum_{i=1}^{t} (p_i x_i + b_i y_i) \leq b \text{ and } \sum_{j=1}^{i} x_j + y_i \geq d_{1,i} \qquad i = 1, \ldots, t \right\}$$

The budget constraint $\sum_{i=1}^{t}(p_i x_i + b_i y_i) \leq b$ requires the total cost for the first t periods not to exceed budget b. If the feasible set of $F_t(b)$ is empty, we define $F_t(b)$ to be $-\infty$. We also assume that a maximum over an empty set is $-\infty$. If demand has to be satisfied (no backlogs) at the end of the time horizon, we only need to select the best solution associated with any cost equal to $\min\{b | F_T(b) \geq d_{1T}\}$.

4.2 Pseudo-polynomial Algorithm

We analyze x_t for each period t to develop a dynamic programming algorithm. The goal is to write $F_t(b)$ in terms of $F_{t-1}(a)$ for $a \leq b$ while satisfying all the necessary constraints in the dual problem. Based on the structure of $F_t(b)$, the following recursive structure can be easily verified:

$$F_t(b) = \max_{\substack{x_t \in [l_t, u_t] \cup \{0\} \\ a, y_t \geq 0}} \left\{ F_{t-1}(a) + x_t \,\middle|\, F_{t-1}(a) + x_t + y_t \geq d_{1,t} \text{ and } a + p_t x_t + b_t y_t \leq b \right\}$$

where a can be interpreted as the maximum allowable budget for the first $t-1$ periods. The boundary condition is $F_0(a) = 0$ for any a.

Let us first assume that $F_t(b)$ is feasible. There are three cases to consider: 1) x_t is zero when it is not optimal to produce in period t, 2) x_t attends the maximum production quantity u_t, and 3) x_t is in $[l_t, u_t)$, the case when the production quantity is restricted by the budget.

Case 1. Let $x_t = 0$. Without any production, if $\max_{a \leq b} F_{t-1}(a)$ is no smaller than $d_{1,t}$, we can simply set $a = b$ to maximize $F_{t-1}(a)$. If $F_{t-1}(b) < d_{1,t}$, backlogging is required, i.e., $y_t > 0$. We know that the backlogging quantity y_t must equal to $d_{1,t} - F_{t-1}(a)$ if there exists an a such that $F_{t-1}(a) > -\infty$. By monotonicity of $F_{t-1}(a)$ in a, we only need to search for a such that $F_{t-1}(a)$ is maximized while the constraint $a + b_t(d_{1,t} - F_{t-1}(a)) \leq b$ is satisfied. If there exist multiple values of a returning the same $F_{t-1}(a)$, we select the largest one. Following this analysis, we can write

$$F_t(b) \leq \begin{cases} F_{t-1}(b) & \text{if } F_{t-1}(b) \geq d_{1,t}, \\ \max\limits_{0 \leq a \leq b} \{F_{t-1}(a) | a + b_t(d_{1,t} - F_{t-1}(a)) \leq b\} & \text{if } F_{t-1}(b) < d_{1,t}, \\ -\infty & \text{otherwise.} \end{cases} \tag{5}$$

Case 2. Let $x_t = u_t$. we can follow the analysis in Case 1 to obtain

$$F_t(b) \leq \begin{cases} u_t + F_{t-1}(b - p_t u_t) & \text{if } F_{t-1}(b - p_t u_t) \geq d_{1,t} - u_t, \\ u_t + \max\limits_{0 \leq a \leq b - p_t u_t} \{F_{t-1}(a) | a + b_t(d_{1,t} - u_t - F_{t-1}(a)) \leq b - p_t u_t\} & \\ & \text{if } F_{t-1}(b - p_t u_t) < d_{1,t} - u_t, \\ -\infty & \text{otherwise.} \end{cases} \tag{6}$$

Case 3. When $x_t = [l_t, u_t)$, we have the following recursion:

$$F_t(b) \leq \max_{b-p_t(u_t-1)\leq a\leq b-p_t l_t} F_{t-1}(a) + x_t(a) \tag{7}$$

where

$$x_t(a) = \begin{cases} -\infty & \text{if } \phi_t < l_t, \\ \phi_t & \text{if } F_{t-1}(a) + \phi_t \geq d_{1t}, \\ \phi_t & \text{if } F_{t-1}(a) + \phi_t < d_{1t} \text{ and } \phi_t \leq \varphi_t, \\ \varphi_t & \text{if } F_{t-1}(a) + \phi_t < d_{1t},\ \ \phi_t > \varphi_t, \text{ and } \varphi_t \geq l_t, \\ -\infty & \text{otherwise,} \end{cases} . \tag{8}$$

$\phi_t = \min\{(b-a)/p_t, u_t - 1\}$ is the upper bound on the production without backlogs, and $\varphi_t = (b-a-b_t(d_{1t}-F_{t-1}(a)))/(p_t-b_t)$ is the maximum production with positive backlogs. The following two claims establish this relationship.

Claim 5. *If $F_{t-1}(a) + \phi_t < d_{1t}$, for any optimal solution x_t, choosing $y_t = d_{1t} - F_{t-1}(a) - x_t$ is optimal.*

Proof. Suppose that we have an optimal solution $(\bar{x}_t, \bar{y}_t)$ with $\bar{y}_t > d_{1t} - F_{t-1}(a) - \bar{x}_t = y_t$ and $\bar{x}_t = x_t$. Then, we must have $a + p_t x_t + b_t y_t = a + p_t \bar{x}_t + b_t y_t < a + p_t \bar{x}_t + b_t \bar{y}_t \leq b$. We consider $\tilde{x}_t = (b - a - b_t y_t)/p_t$ and $\tilde{y}_t = d_{1t} - F_{t-1}(a) - \tilde{x}_t$. Note that $\tilde{x}_t > x_t$, and we have $F_{t-1}(a)+\tilde{x}_t+\tilde{y}_t = F_{t-1}(a)+\tilde{x}_t+d_{1t}-F_{t-1}(a)-\tilde{x}_t = d_{1t}$. Thus, $(\tilde{x}_t, \tilde{y}_t)$ satisfies $\tilde{y}_t = d_{1t}-F_{t-1}(a)-\tilde{x}_t$, is feasible to both budget and demand constraints, and yields a larger objective value. This is a contradiction. Hence, solution $(\bar{x}_t, \bar{y}_t)$ is not optimal. □

Claim 6. *Given any a, $x_t(a)$ yields the maximum production level.*

Proof. Suppose $F_{t-1}(a) + \phi_t \geq d_{1t}$. Clearly taking $y_t = 0$ and $x_t = \phi_t$ is feasible and optimal since x_t is bounded by ϕ_t and $a+p_t\phi_t+b_t(0) \leq b$ by our definition of ϕ_t. Now suppose $F_{t-1}(a)+\phi_t < d_{1t}$. We must have $F_{t-1}(a) + x_t < d_{1t}$ for any $x_t \in [l_t, \phi_t]$ (if $\phi_t < l_t$, it is easy to see that no feasible solution exists). By Claim 5, we can assume $y_t = d_{1t} - F_{t-1}(a) - x_t > 0$, and thus, we have $x_t \leq \frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t}$.

Suppose $a + p_t\phi_t + b_t(d_{1t} - F_{t-1}(a) - \phi_t) \leq b$. We have

$$\phi_t \leq \frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t}.$$

Together with the fact that $x_t \leq \phi_t$, $x_t = \phi_t$ maximizes the production.

Now suppose $a + p_t\phi_t + b_t(d_{1t} - F_{t-1}(a) - \phi_t) > b$. We have

$$\phi_t > \frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t} \geq x_t.$$

Hence, taking $x_t = \frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t}$ yields optimality.

Note that if $\frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t} < l_t$, no feasible solution exists. Suppose such a solution $\tilde{x}_t$ exists. Then, we have $\tilde{x}_t > \frac{b-a-b_t(d_{1t}-F_{t-1}(a))}{p_t-b_t}$, which implies $a + p_t\tilde{x}_t + b_t(d_{1t} - F_{t-1}(a) - \tilde{x}) > b$. Together with the fact that $\tilde{y}_t$ is lower bounded by $d_{1t} - F_{t-1}(a) - \tilde{x}$ by Claim 5, we conclude that $\tilde{x}_t$ is infeasible for any choices of $\tilde{y}_t$. Hence, any solution (x_t, y_t) is infeasible. □

Next, we summarize the findings.

Proposition 3. *We have*

$$F_t(b) = \max\{(5), (6), (7)\}, \tag{9}$$

where by (5), (6), and (7) we refer to the corresponding right-hand sides in these expressions.

Proof. We have already argued that $F_t(b) \leq \max\{(5), (6), (7)\}$ if $F_t(b)$ is feasible. If $F_t(b)$ is infeasible, the same inequality holds by definition. It remains to show that (5), (6), and (7) are all less than or equal to $F_t(b)$.

The first one is easy to establish by defining $a = b$, $x_t = 0$ and $y_t = 0$ if $F_{t-1}(b) \geq d_{1,t}$. If $F_{t-1}(b) < d_{1,t}$ and there exists a such that $a + b_t(d_{1,t} - F_{t-1}(a)) \leq b$, we can take $x_t = 0$ and $y_t = d_{1,t} - F_{t-1}(a')$ for a' being optimal in the underlying expression.

To show that expression (6) is less than or equal to $F_t(b)$, we either set $x_t = u_t$, $y_t = 0$ or $x_t = u_t$, $y_t = d_{1,t} - u_t - F_{t-1}(a')$ based on the two cases in (6). We can apply similar arguments to those used in the last case by choosing $x_t = x_t(a')$ according to (8) and setting $y_t = (d_{1t} - F_{t-1}(a') - x_t)^+$. $\square$

The dynamic programming algorithm computes $F_t(b)$ for all $t = 1, \ldots, T$ and $b = 0, 1, 2, \ldots, B$, starting at the first period. We have $\phi' = \min\{b/p_1, u_1\}$ and $\varphi' = (b - b_1 d_1)/(p_1 - b_1)$. Then, the boundary condition is

$$F_1(b) = \begin{cases} \phi' & \text{if } l_1 \leq \phi' \text{ and } \phi' \geq d_1, \\ \varphi' & \text{if } l_1 \leq \phi' < d_1 \text{ and } \varphi' \geq l_1, \\ 0 & \text{if } l_1 \leq \phi' < d_1, \varphi' < l_1, \text{ and } b_1 d_1 \leq b, \\ 0 & \text{if } l_1 > \phi' \text{ and } b_1 d_1 \leq b, \\ -\infty & \text{otherwise.} \end{cases}$$

This can be showed by following the similar arguments presented in Claim (6). The running time to evaluate $F_1(b)$ is clearly $O(B)$. Using (9) for the remaining periods, we can easily see that the running time for the entire backward recursion algorithm is $O(TB^2)$, where the extra order of B results from the search of a in (5), (6), and (7).

4.2.1 Dynamic Programming Algorithm

We can develop an FPTAS directly by modifying the aforementioned pseudo-polynomial algorithm to run over a trimmed set of budget values, which are polynomially many, and are multiples of an integer constant K that depends only on the size of the problem instance and ϵ. Let us consider $b = 0, K, 2K, \ldots, UK$ for a fixed K and integer U. Then, for any $\eta = 0, 1, 2, \ldots, U$, we have the following adapted problem:

$$\bar{F}_t(\eta) = \max_{\substack{x_i \in [l_i, u_i] \cup \{0\} \ i=1,\ldots,t \\ y_1, \ldots, y_t \geq 0 \\ n_1^1, \ldots n_t^1, n_1^2, \ldots n_t^2 \in \mathbb{Z}^+}} \left\{ \sum_{i=1}^{t} x_i \ \middle| \ \begin{array}{ll} \sum_{j=1}^{i} x_j + y_i \geq d_{1i} & i = 1, \ldots, t \\ \sum_{i=1}^{t} (n_i^1 + n_i^2) = \eta & \\ p_i x_i \leq n_i^1 K & \\ b_i y_i \leq n_i^2 K & \end{array} \right\}.$$

We can interpret $\bar{F}_t(\eta)$ as maximizing production subject to budget not exceeding ηK. If $\bar{F}_t(\eta)$ is infeasible, we define $\bar{F}_t(\eta) = -\infty$.

Given any multiplier η, we first analyze the normalized production and backlogging costs separately, and show that the solution associated with these normalized costs is feasible to the adapted recursion. Let n_t^1 be the production cost multiplier and n_t^2 be the backlogging cost multiplier at time t.

Claim 7. *If $t \geq 2$ and $\bar{F}_t(\eta)$ is feasible, there exists a solution such that $n_i^1 + n_i^2 = \lceil p_i x_i/K \rceil + \lceil b_i y_i/K \rceil$ for $i = 2, 3, \ldots, t$.*

Proof. Given that $\bar{F}_t(\eta)$ is feasible, the problem of finding n_i^1 and n_i^2 for $i = 1, 2, \ldots, t$ is subject to $\sum_{i=1}^{t}(n_i^1 + n_i^2) = \eta$, $n_i^1 \geq \lceil p_i x_i/K \rceil$, $n_i^2 \geq \lceil b_i y_i/K \rceil$ and $n_i^1, n_i^2 \in \mathbb{Z}^+$ for $i = 1, 2, \ldots, t$. Since the problem is feasible, we have $\sum_{i=1}^{t} \lceil p_i x_i/K \rceil + \lceil b_i y_i/K \rceil \leq \eta$. Thus, we can set $n_1^1 = \lceil p_1 x_1/K \rceil + \eta - \sum_{i=1}^{t} \left(\lceil p_i x_i/K \rceil + \lceil b_i y_i/K \rceil\right)$, $n_1^2 = \lceil b_1 y_1/K \rceil$, and $n_i^1 = \lceil p_i x_i/K \rceil$ and $n_i^2 = \lceil b_i y_i/K \rceil$ for $i = 2, 3, \ldots, t$. It is then easy to see that $n_1^1 + n_1^2 = \lceil p_1 x_1/K \rceil + \lceil b_1 y_1/K \rceil + \eta - \sum_{i=1}^{t} \left(\lceil p_i x_i/K \rceil + \lceil b_i y_i/K \rceil\right) \geq \lceil p_1 x_1/K \rceil + \lceil b_1 y_1/K \rceil$. □

Next, we show how to adapt (9) to achieve an FPTAS for the lot-sizing problem with MOQ. The modifications are made separately for each case, where we assume that all encountered solutions satisfy Claim 7 for $t \geq 2$.

Case 1. When $x_t = 0$, by the choice of the solution from Claim 7, the backlogging cost is no more than some multiple of K, e.g, $b_t y_t \leq n_t^2 K$, which yields $n_t^2 = \lceil b_t y_t/K \rceil$. Together with the demand constraint requiring $y_t = d_{1,t} - \bar{F}_{t-1}(\eta')$, we follow the structure of (5) to obtain

$$\bar{F}_t(\eta) = \begin{cases} \bar{F}_{t-1}(\eta) & \text{if } \bar{F}_{t-1}(\eta) \geq d_{1,t}, \\ \max\limits_{0 \leq \eta' \leq \eta} \{\bar{F}_{t-1}(\eta') | \eta' + \lceil b_t(d_{1,t} - \bar{F}_{t-1}(\eta'))/K \rceil \leq \eta\} & \text{if } \bar{F}_{t-1}(\eta) < d_{1,t}, \\ -\infty & \text{otherwise.} \end{cases} \tag{10}$$

Case 2. When $x_t = u_t$, we have $n_t^1 = \lceil p_t u_t/K \rceil$, $n_t^2 = \lceil b_t y_t/K \rceil$, and $y_t = d_{1,t} - u_t - \bar{F}_{t-1}(\eta')$. The first two equations follow from the definition of n_t^1 and n_t^2, whereas the last equation follows from the demand constraint. Similarly to (6), the equation reads

$$\bar{F}_t(\eta) = \begin{cases} u_t + \bar{F}_{t-1}(\eta - \lceil p_t u_t/K \rceil) & \text{if } \bar{F}_{t-1}(\eta - \lceil p_t u_t/K \rceil) \geq d_{1,t} - u_t, \\ u_t + \max\limits_{\substack{0 \leq \eta' \leq \eta - \lceil p_t u_t/K \rceil \\ \eta' + \lceil b_t(d_{1,t} - u_t - \bar{F}_{t-1}(\eta'))/K \rceil \leq \eta - \lceil p_t u_t/K \rceil}} \bar{F}_{t-1}(\eta') & \text{if } \bar{F}_{t-1}(\eta - \lceil p_t u_t/K \rceil) < d_{1,t} - u_t, \\ -\infty & \text{otherwise.} \end{cases} \tag{11}$$

Case 3. When $x_t \in [l_t, u_t)$, we have $n_t^1 = \lceil p_t x_t/K \rceil$, $n_t^2 = \eta - \eta' - n_t^1$, and $l_t \leq x_t \leq \min\{K(\eta - \eta')/p_t, u_t - 1\}$. Note that given any η, $K(\eta - \eta')/p_t$ is the production budget, and $\eta' = \sum_{i=1}^{t-1}(n_i^1 + n_i^2)$ is the budget that is reserved for periods from 1 to $t-1$. Since we are only interested in η' that yields $K(\eta - \eta')/p_t \leq u_t - 1$, it follows that $\eta' \in [\eta - \lfloor p_t(u_t - 1)/K \rfloor, \eta - \lceil p_t l_t/K \rceil]$. Then, we write

$$\bar{F}_t(\eta) \leq \max_{\eta - \lfloor p_t(u_t-1)/K \rfloor \leq \eta' \leq \eta - \lceil p_t l_t/K \rceil} \bar{F}_{t-1}(\eta') + \bar{x}_t(\eta') \tag{12}$$

where

$$\bar{x}_t(\eta) = \begin{cases} -\infty & \text{if } \phi_t < l_t, \\ \bar{\phi}_t & \text{if } \bar{F}_{t-1}(\eta') + \bar{\phi}_t \geq d_{1t}, \\ \bar{\phi}_t & \text{if } \bar{F}_{t-1}(\eta') + \bar{\phi}_t < d_{1t} \text{ and } \bar{\phi}_t \leq \bar{\varphi}_t, \\ \bar{\varphi}_t & \text{if } \bar{F}_{t-1}(a) + \bar{\phi}_t < d_{1t},\ \ \bar{\phi}_t > \bar{\varphi}_t,\ \text{and } \bar{\varphi}_t \geq l_t, \\ -\infty & \text{otherwise,} \end{cases}$$

and $\bar{\phi}_t = \min\{\lfloor K(\eta - \eta')/p_t \rfloor, u_t - 1\}$, $\bar{\varphi}_t = \lfloor (K(\eta - \eta') - b_t(d_{1t} - \bar{F}_{t-1}(\eta')))/(p_t - b_t) \rfloor$.

Similarly to Proposition 3, we obtain the following result.

Proposition 4. *We have*

$$\bar{F}_t(\eta) = \max\{(10), (11), (12)\}, \tag{13}$$

where by (10), (11), and (12), we refer to the right-hand sides in these expressions. We also obtain $\bar{F}_1(\eta) = F_1(\eta K)$ *for* $\eta = 0, 1, 2, \ldots, U$.

To prove that this algorithm exhibits an FPTAS, we need to show that given any budget, which is a multiple of K, to the original problem, there is a feasible solution in the adapted problem with the required budget bounded by multiples of T.

Proposition 5. *For every integer* η, *we have* $\bar{F}_T(\eta + 2T) \geq F_T(\eta K)$.

Proof. Let x^* be an optimal solution to $F_T(\eta K)$, which clearly satisfies $\sum_{t=1}^{T}(p_t x_t^* + b_t y_t^*) \leq \eta K$. Let us define $n_t^1 = \lceil p_t x_t^*/K \rceil$ and $n_t^2 = \lceil b_t y_t^*/K \rceil$ for every $t = 1, 2, \ldots, T$. This gives a solution to $\bar{F}_T(\sum_{t=1}^{T}(n_t^1 + n_t^2))$. Observe that $\sum_{t=1}^{T} \lceil p_t x_t^*/K \rceil + \lceil b_t y_t^*/K \rceil \leq \sum_{t=1}^{T}(p_t x_t^*/K + b_t y_t^*/K + 2) \leq \eta + 2T$, which completes the proof. □

The next proposition is similar to a result in van Hoesel and Wagelmans (2001), which establishes an upper bound on the value of the multiplier. This upper limit is essential in proving the required computational time for the FPTAS.

Proposition 6. *There exists an integer* $\eta \in \{0, 1, \ldots, \lceil B/K \rceil + 4T\}$ *with* $\bar{F}_T(\eta) > -\infty$. *Moreover, the smallest such value* η^* *does not exceed* $z^*/K + 4T$.

Proof. Let us denote by p_t^* the budget allocation for production in time period t based on an optimal solution, and b_t^* the budget allocation for backlogging. Clearly, $z^* = \sum_{t=1}^{T}(p_t^* + b_t^*)$. Let us define $n_t^1 = \lceil p_t^*/K \rceil$, $n_t^2 = \lceil b_t^*/K \rceil$ and $\bar{\eta} = \sum_{t=1}^{T}(n_t^1 + n_t^2)$. From Proposition 5, we obtain $\bar{F}_T(\bar{\eta} + 2T) \geq F_T(\bar{\eta} K) \geq F_T(z^*) > -\infty$ since $z^* = \sum_{t=1}^{T}(p_t^* + b_t^*) \leq \sum_{t=1}^{T} \lceil p_t^*/K \rceil K + \lceil b_t^*/K \rceil K = \bar{\eta} K$. We also observe that $\bar{\eta} + 2T = \sum_{t=1}^{T}(\lceil p_t^*/K \rceil + \lceil b_t^*/K \rceil) + 2T \leq \sum_{t=1}^{T}(p_t^*/K + b_t^*/K + 2) + 2T = z^*/K + 4T \leq \lceil B/K \rceil + 4T$. Thus, $\bar{\eta} + 2T$ is an integer smaller than or equal to $\lceil B/K \rceil + 4T$ with $\bar{F}_T(\bar{\eta} + 2T) > -\infty$. This together with $\eta^* \leq \bar{\eta} + 2T$ yields $\eta^* \leq z^*/K + 4T$. □

4.3 The Approximation Scheme

4.3.1 A Polynomial Ratio Approximation Algorithm

We provide a simple polynomial time approximation algorithm to compute an integer upper bound B on z^*, which is at most $2Tz^*$. The algorithm is adapted directly from van Hoesel and Wagelmans (2001). The algorithm finds the smallest L such that the solution is feasible, and in each time period neither the production cost nor the backlogging cost exceed L. Let l be any value of L. We can determine if there exists a feasible solution by checking both the upper bound on the production levels and the upper bound on the number of backlogs iteratively via a simple dynamic algorithm.

Let us first define the upper bound on the production levels by

$$\bar{c}_t(l) = \begin{cases} u_t & \text{if } l/p_t \geq u_t, \\ l/p_t & \text{if } l_t \leq l/p_t < u_t, \\ 0 & \text{otherwise.} \end{cases}$$

This is the maximum amount that can be produced with budget l. Clearly, the upper bound on the number of backlogs is l/b_t given that the backlogging cost must not exceed l. Furthermore, we denote the maximum production levels given l by $M_t(l) = M_{t-1}(l) + \bar{c}_t(l)$. The procedure to check if there exists a feasible production plan for a given budget l begins with $M_0(l) = 0$ and stops if $d_{1,t} - M_t(l) > l/b_t$ for any $t = 1, \ldots, T$. The overall algorithm to find L by bisection can be summarized as follows.

Step 1. Let $l = \max_{t=1,\ldots,T}\{p_t u_t + b_t d_{1,t}\}$.

Step 2. For $t = 1, \ldots, T$,

  Step 2a. Compute $\bar{c}_t(l)$ and $M_t(l) = M_{t-1}(l) + \bar{c}_t$.

  Step 2c. If $d_{1,t} - M_t(l) > l/b_t$, continue the binary search for l on the upper subinterval.

Step 3. If $d_{1,t} - M_t(l) \leq l/b_t$ for all $t = 1, \ldots, T$, continue the binary search in the lower subinterval.

Let L be the returned value of the algorithm. Clearly, $L \leq z^*$ since $l = z^*$ provides a feasible solution and L is the smallest such l. As we have to account for $2T$ many cost functions, the value of the solution produced by this algorithm is at most $2TL \leq 2Tz^*$. The running time of the algorithm is $O(T \log(\max_{t=1,\ldots,T}\{p_t u_t + b_t d_{1,t}\}))$, and hence polynomial.

4.3.2 The Fully Polynomial Time Approximation Scheme

We first present the approximation scheme, then we argue that it is fully polynomial.

Step 1. Compute an upper bound B on z^* by the algorithm presented in Section 4.3.1.

Step 2. Set $K = \max\{\lfloor \epsilon B/(8T^2) \rfloor, 1\}$.

Step 3. Calculate $\bar{F}_t(\eta)$ for all $t = 1, 2, \ldots, T$ and $\eta = 0, 1, 2, \ldots, \lceil B/K \rceil + 4T$ based on Section 4.2.1.

Step 4. Choose the smallest $b \in \{0, K, 2K, \ldots, (\lceil B/K \rceil + 4T)K\}$ such that $\bar{F}_T(b/K) > -\infty$.

Proposition 7. *The above algorithm provides a solution value that does not exceed $(1+\epsilon)z^*$, and its running time only depends on the size of the problem instance and $1/\epsilon$.*

Proof. The bound on the solution returned by the algorithm can be easily obtained from the fact that $z^* + 4TK \leq z^* + \epsilon B/(2T) \leq (1+\epsilon)z^*$ since $K \leq \epsilon B/(8T^2)$ and $B \leq 2Tz^*$. If $\lceil B/K \rceil < T$, then the running time $O(T(\lceil B/K \rceil + 4T)^2)$ is obviously polynomial. Let us define $UB := \max_{t=1,\ldots,T}\{p_t u_t + b_t d_{1,t}\}$, and assume that $\lceil B/K \rceil \geq T$. If $\epsilon B/(8T^2) > 1$, then $K > \epsilon B/(16T^2)$. Otherwise, we have $K \geq \epsilon B/(8T^2)$. In both cases, the running time is $O(T^5/\epsilon^2 + T \log UB) = O((T^5 \log UB)/\epsilon^2)$. □

The main difference from the results of van Hoesel and Wagelmans (2001) is the fact that we restrict the cost functions to be linear. This is necessary to develop a recursion to the dual problem for the lot-sizing problem with MOQ. The dual problem in turn provides a basis to formulate the pseudo-polynomial algorithm, which finally leads to the FPTAS.

Our result is also different from Halman et al. (2008) in that we do not account for the stochastic nature of the demand. Furthermore, we can handle the minimum order quantity and fixed procurement costs (as shown in the next section), which yields a non-convex procurement cost function that violates the underlying assumption of their model. This fixed cost feature is a very important generation of our approach (in addition to MOQ's).

4.4 Fixed Procurement Costs

Consider the case when a fixed procurement cost is incurred whenever an order is placed. This can be easily modeled by redefining the cost of production as

$$\bar{p}_t(x_t) = \begin{cases} 0 & \text{if } x_t = 0, \\ p_t x_t + S_t & \text{if } x_t \in [l_t, u_t] \end{cases}$$

where S_t is the fixed procurement cost for period t. As a result of this production function, (11) and (12) are modified as follows.

$$\bar{F}_t(\eta) \le \begin{cases} u_t + \bar{F}_{t-1}(\eta - \lceil (p_t u_t + S_t)/K \rceil) & \text{if } \bar{F}_{t-1}\left(\eta - \lceil \frac{p_t u_t + S_t}{K} \rceil\right) \ge d_{1,t} - u_t, \\ u_t + \max\limits_{\substack{0 \le \eta' \le \eta - \lceil (p_t u_t + S_t)/K \rceil \\ \eta' + \lceil (b_t(d_{1,t} - u_t - \bar{F}_{t-1}(\eta'))/K \rceil \le \eta - \lceil (p_t u_t + S_t)/K \rceil}} \bar{F}_{t-1}(\eta') & \text{if } \bar{F}_{t-1}\left(\eta - \lceil \frac{p_t u_t + S_t}{K} \rceil\right) < d_{1,t} - u_t, \\ -\infty & \text{otherwise.} \end{cases} \tag{14}$$

$$\bar{F}_t(\eta) \le \max_{\eta - \left\lfloor \frac{p_t(u_t - 1) + S_t}{K} \right\rfloor \le \eta' \le \eta - \left\lceil \frac{p_t l_t + S_t}{K} \right\rceil} \{\bar{F}_{t-1}(\eta') + \tilde{x}(\eta')\}. \tag{15}$$

Here, $\tilde{x}(a)$ has the same definition as $\bar{x}(a)$ but with $\bar{\phi}_t$ and $\bar{\varphi}_t$ replaced by $\tilde{\phi}_t = \min\{\lfloor (K(\eta - \eta') - S_t)/p_t \rfloor, u_t - 1\}$ and $\tilde{\varphi}_t = \lfloor (K(\eta - \eta') - S_t - b_t(d_{1t} - \bar{F}_{t-1}(\eta')))/(p_t - b_t) \rfloor$, respectively. The recursive formula (13) becomes $\bar{F}_t(\eta) = \max\{(10), (14), (15)\}$. To complete the FPTAS, we also need to include the fixed procurement costs in the algorithm in Section 4.3.1. We do so by redefining the maximum production level as

$$\bar{c}_t(l) = \begin{cases} u_t & \text{if } \frac{l - S_t}{p_t} \ge u_t \\ \frac{l - S_t}{p_t} & \text{if } l_t \le \frac{l - S_t}{p_t} < u_t \\ 0 & \text{otherwise.} \end{cases}$$

The trivial upper bound in Step 1 is now $\max_{t=1,\dots,T}\{p_t u_t + b_t d_{1,t} + S_t\}$. Then, we can follow the same procedure as the one described in Section 4.3.1 to obtain an integer upper bound B on z^*. In this way, an FPTAS for the MOQ problem with fixed procurement costs follows with the running time $O((T^5 \log UB)/\epsilon^2)$.

5 Conclusion

This paper considers a single item lot-sizing problem with both capacity constraints and minimum order quantity requirements. We prove that with constant capacities and minimum order quantities, the production sequences in the optimal solution have a special structure. Based on this, we developed a dynamic programming approach to solve this problem in polynomial time. We also identified a polynomial case of general minimum order quantities by imposing some assumptions on cost functions. Furthermore, we consider the NP-hard version of the underlying problem in which the capacity constraints and minimum order quantity requirements depend on the time period. By dualizing the problem, we successfully develop a fully polynomial approximation scheme to cope with its NP-hard nature, and hence, solve the problem approximately in polynomial time. We also show that our results can be easily extended to capture the fixed procurement costs.

References

Ahuja, R. K., Magnanti, T. L., Orlin, J. B., and Magnanti, T. (1993). *Network Flows Theory, Algorithms and Applications*. Prentice Hall, Inc.

Anderson, E. J. and Cheah, B. S. (1993). Capacitated lot-sizing with minimum batch sizes and setup times. *International Journal of Production Economics*, 30-31:137 – 152.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. (1993). *Nonlinear Programming Theory and Algorithms*. John Wiley and Sons, Inc.

Chauhan, S. S., Eremeev, A. V., Romanova, A. A., Servakh, V. V., and Woeginger, G. J. (2005). Approximation of the supply scheduling problem. *Operations Research Letters*, 33:249–254.

Chubanov, S., Kovalyov, M. Y., and Pesch, E. (2006). An FPTAS for a single-item capacitated economic lot-sizing problem with monotone cost structure. *Mathematical Programming Series A 106*, pages 453–466.

Chubanov, S., Kovalyov, M. Y., and Pesch, E. (2008). A single-item economic lot-sizing problem with a non-uniform resource: Approximation. *European Journal of Operational Research*, 189:877–889.

Constantino, M. (1998). Lower bounds in lot-sizing models: A polyhedral study. *Mathematics of Operations Research*, 23(1):101.

Fisher, M. and Raman, A. (1996). Reducing the cost of demand uncertainty through accurate response to early sales. *Operations Research*, 44:87–99.

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

Florian, M., Lenstra, J. K., and Kan, A. R. (1980). Deterministic production planning: Algorithms and complexity. *Management Science*, 26:669–679.

Halman, N., Klabjan, D., Mostagir, M., Orlin, J. B., and Simchi-Levi, D. (2008). A fully polynomial time approximation scheme for single-item stochastic inventory control problems with discrete demand. *Northwestern University*. Available at http://www.klabjan.dynresmanagement.com.

Kovalyov, M. Y. (1996). A rounding technique to construct approximation algorithms for knapsack and partition-type problems. *Applied Mathematics and Computer Science*, 6(4):789–801.

Lee, C.-Y. (2004). Inventory replenishment model: lot sizing versus just-in-time delivery. *Operations Research Letters*, 32(6):581 – 590.

Love, S. F. (1968). Dynamic deterministic production and inventory models with piecewise concave costs. *Department of Operations Research, Stanford University, Technical report*, 13. NSF grant GK-1402.

Mercé, C. and Fontan, G. (2003). Mip-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics*, 85(1):97 – 111.

Musalem, E. P. and Dekker, R. (2005). Controlling inventories in a supply chain: A case study. *International Journals of Production Economics*, 93:179–188.

Ng, C. T., Kovalyov, M. Y., and Cheng, T. C. E. (2008). An fptas for a supply scheduling problem with non-monotone cost functions. *Naval Research Logistics*, 55:194–199.

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

Pochet, Y. and Wolsey, L. A. (2007). Single item lot-sizing with non-decreasing capacities. *Available at SSRN: http://ssrn.com/abstract=1010623*.

Snyder, R. D. (1974). A note on fixed and minimum order quantity stock systems. *Operations Research Quarterly*, 25:635–639.

van Hoesel, S. P. M. and Wagelmans, A. P. M. (1996). An $O(t^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Science*, 42:142–150.

van Hoesel, S. P. M. and Wagelmans, A. P. M. (2001). Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. *Mathematics of Operations Research*, 26:339–357.

Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of economic lot size model. *Management Science*, 5(1):89–96.

Zangwill, W. I. (1966). A deterministic multiperiod production scheduling model with backlogging. *Management Science*, 13:105–119.

Zhao, Y. and Katehakis, M. N. (2006). On the structure of optimal ordering policies for stochastic inventory systems with minimum order quantity. *Probability in the Engineering and Informational Sciences*, 20:257–270.