Multi-period lot-sizing with supplier selection: Structural results, complexity and algorithms

Meichun Lin [a], Woonghee Tim Huh [a], Guohua Wan [b,*]

a Sauder School of Business, University of British Columbia, BC V6T 1Z4, Canada

b* Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China

Published in Operations Research Letters (2021) 49 (4), 602-609. DOI: 10.1016/j.orl.2021.05.013

Abstract: We consider a multi-period lot-sizing problem with multiple products and multiple suppliers. Demand is deterministic and time-varying. The objective is to determine order quantities to minimize the total cost over a finite planning horizon. This problem is strongly NP-hard. For a special case, we extend the classical zero-inventory-ordering principle and solve it by dynamic programming. Based on this new extension, we also develop a heuristic algorithm for the general problem and computationally show that it works well.

Keywords: Lot-sizing and scheduling, Algorithm, Computational complexity, Supply chain

1. Introduction

We study a multi-period lot-sizing problem with multiple products and multiple suppliers, where demand for products is deterministic and known in advance over the planning horizon. When a product is purchased from a supplier, it incurs a supplier-dependent fixed cost, and a per-unit variable cost that is specific to the product and the supplier. We also consider a product-dependent holding cost for any unit remaining at the end of a period. The objective of the problem is to determine when to order as well as how many units of products to order from each supplier to minimize the total cost over the planning horizon. This is one of the most general versions of the economic lot-sizing (ELS) problem involving several suppliers, and this particular extension, proposed by [2], is called the *multi-period lot-sizing with supplier selection problem* (MLSSP).

The ELS problem and its extensions have been studied extensively in the past half-century (see [4], [5] for detailed surveys), and heuristics to solve these problems are included in modern MRP and ERP systems [14], [16]. A common approach to solve the ELS problem is to use dynamic programming, dating back to the basic single-product single-supplier problem [17]. This problem exhibits the well-known zero-inventory property, which says that there should be no inventory when an order is placed. The solution approach is based on dynamic programming, which decomposes time periods based on when an order is placed.

While some ELS problems can be solved via dynamic programming, other problems are difficult to solve and they are proven to be NP-hard. Examples include systems with nonlinear production costs [9], capacitated production [3], and multiple echelons [19]. For the variant of ELS problem that we study, MLSSP, we formally establish that it is NP-hard in the strong sense. Since it is challenging to solve MLSSP, the existing approaches are based on heuristics. An enumerative approach with a partial search tree is used in [2], and additional constraints are introduced in [6] to restrict the feasible region. For variants of MLSSP, genetic algorithms are adapted in [8] and [15].

We take a different approach. Since the general MLSSP problem is intractable, we impose an additional assumption that the per-unit variable cost depends on both the product and the supplier in an additive manner. In other words, the per-unit variable cost P_{ij} associated with product i and supplier j can be written in the form of $P_{ij} = \tilde{P_i} + \hat{P_j}$. It is reasonable to adopt this assumption, for example, when the products are bulk materials such as iron ore, coal, and steel, in which case a large portion of the purchase costs of those products is the transportation cost, heavily dependent on the locations of the suppliers, and the costs of the same product across suppliers are almost the same. This assumption is applicable to a construction company named HNSL First Engineering Co. (Henan province, China) when it was building a large bridge during 2010-2012. It needed substantial amounts of cement and steel material and there were several suppliers at various locations from which the company could purchase cement and steel. (The third author provided consultancy for this project during 2010-2011.)

With this assumption, we can show that while the optimal solution does not satisfy the classical zero-inventory-ordering property, it exhibits a generalization of this property, which we call the *echelon* zero-inventory-ordering property. We show that products can be ordered such that the set of products to be ordered in each period has an echelon structure. This is a key observation that enables us to develop a tractable dynamic programming formulation. This formulation is unlike any other dynamic programming formulation for the ELS problem and its extensions in the sense that it does not decompose the problem based on time periods only, but rather it is based on a two-dimensional decomposition consisting of time periods and the set of products to be ordered. This novel approach enables us to attain a dynamic programming solution that is more sophisticated than any other dynamic programming solution in the ELS literature. Based on this approach, we further develop an efficient heuristic for the general problem, which turns out to perform well in the computation experiments.

We note that the MLSSP problem is closely related to the joint replenishment problem (JRP) which also deals with multiple products. The JRP system has only one supplier but there are fixed costs of ordering associated with each product as well as with joint ordering (we refer to [13] for a detailed survey). The MLSSP problem is different from the JRP problem since it has multiple suppliers instead of one supplier, and it has supplier-specific fixed costs but not product-specific costs. Thus, the complexity proof [1] or heuristics for the JRP cannot be adopted directly to our problem.

2. Model

We consider a multi-period lot-sizing problem with multiple products and multiple suppliers, where demand is deterministic and time-varying over a finite planning horizon. The purchase cost is supplier-dependent and consists of a fixed ordering cost, which is independent of the amount ordered, and a per-unit variable cost. Any unit remaining at the end of a period incurs a product-dependent holding cost. All demand must be satisfied without any delay, and product replenishment is instantaneous. The problem is to determine the order quantity for each product in each period so as to minimize the total cost of product purchase and inventory holding. This problem is called the multi-period lot-sizing with supplier selection problem (MLSSP). We adopt the notations used in [2] as follows.

The set of indices:

- $i = 1, \ldots, I$, indexes products, where $I \geq 1$ is the total number of products;
- $j = 1, \ldots, J$, indexes suppliers, where $J \geq 1$ is the total number of suppliers;
- $t = 1, \ldots, T$, indexes time in a forward manner, where T is the planning horizon.

The exogenous parameters:

- O_j: fixed ordering cost from supplier j, incurred if any positive amount of any product is ordered from supplier j;
- P_{ij}: unit purchase price for product i from supplier j;
- H_i: unit holding cost for product i over a single period;
- D_i^t: demand for product i in period t.

We assume that quantities $\{H_i\}$ are positive and all other quantities are non-negative, and that there is no inventory at the beginning of the planning horizon (i.e., period 1). It is also required that all demands must be satisfied without any backlog. The following are decision variables and intermediate variables:

- $x_{ij}^t \geq 0$: quantity of product i to be ordered from supplier j at the beginning of period t;
- $y_j^t \in \{0, 1\}$: a binary indicator variable for whether a positive amount of order is placed to supplier j in period t, i.e., $y_j^t = 1$ if and only if $\sum_i x_{ij}^t > 0$;
- $z_i^t \geq 0$: an intermediate variable indicating the amount of inventory for product i at the end of period t. Then we have $z_i^t = \sum_{t'=1}^t \sum_j x_{ij}^{t'} - \sum_{t'=1}^t D_i^{t'}$, which is the difference between the cumulative order quantity up to period t and the cumulative demand up to period t.

Note that we have to determine the ordering quantities, $\{x_{ij}^t\}$. The other quantities, $\{y_j^t\}$ and $\{z_i^t\}$, are determined by the choice of $\{x_{ij}^t\}$. The objective is to minimize the total cost given by:

$$\sum_t \sum_j \sum_i P_{ij} x_{ij}^t + \sum_t \sum_j O_j y_j^t + \sum_t \sum_i H_i z_i^t ,$$

where the first term is the variable purchase cost, the second term is the fixed ordering cost, and the last term is the inventory holding cost.

The problem can be formulated as a mixed integer program. Let $M_i^t = \sum_{t'=t}^T D_i^{t'}$, which is the total demand for product i between periods t and T. Then, it is straightforward to see that the order quantity in period t for product i should not exceed M_i^t, implying $x_{ij}^t \leq M_i^t$. The mixed integer programming formulation for MLSSP is given below:

$$\begin{aligned} \min \quad & \sum_t \sum_j \sum_i P_{ij} x_{ij}^t + \sum_t \sum_j O_j y_j^t + \sum_t \sum_i H_i z_i^t \\ \text{s.t.} \quad & z_i^t = \sum_{t'=1}^t \sum_j x_{ij}^{t'} - \sum_{t'=1}^t D_i^{t'} \geq 0 \qquad \forall i, t \\ & 0 \leq x_{ij}^t \leq M_i^t y_j^t \qquad \forall i, j, t \\ & y_j^t \in \{0, 1\} \qquad \forall j, t. \end{aligned}$$

According to [2], an optimal ordering policy should satisfy the following two properties. First, in any given period, all units of any particular product should be ordered from one supplier instead of spreading them among multiple suppliers because once multiple fixed ordering costs are paid for, we can purchase all units from the supplier with the lowest variable cost. Second, no inventory of a product is carried into a period in which a positive quantity of that product is ordered; if a positive amount of product i is ordered in period t and its inventory is carried into period t, then it is better off if the last order for the product before period t is changed. (Specifically, it can be decreased such that the inventory carried into period t becomes zero, or it can be increased such that no order is placed for product i in period t.) This is indeed the zero-inventory-ordering property for each product. However, it does not specify any relationship between the carried-over inventory of one product and the ordering of another product. In other words, it allows the possibility that when product i is ordered in period t, there might be some units of another product i' carried over from period $t-1$ to period t. We refer to it as the *product-specific* zero-inventory-ordering property.

We note that MLSSP is strongly NP-hard. One way to see this result is to note that MLSSP with a single period is a Simple Plant Location Problem (SPLP), which contains the set covering problem [11]. Since the set covering problem is known to be strongly NP-hard [10], SPLP and thus MLSSP are strongly NP-hard. Alternatively, we can establish a reduction from the Exact Cover by 3-Sets (X3C) problem, a strongly NP-hard problem [10] (see Appendix 1). Given the intractability of MLSSP, we consider a special case of MLSSP where a polynomial-time solution can be found in the next section.

3. Additive per-unit purchase cost: two-dimensional dynamic programming formulation

Since MLSSP is strongly NP-hard, a polynomial-time algorithm for solving it to optimality is not expected unless $P = NP$. We proceed to identify sufficient conditions under which we can find polynomial-time solution algorithms. More specifically, we focus on properties of the input parameters, such as independency on the product index i or the supplier index j. Note that even if parameters H_i's are independent of i or O_j's are independent of j, MLSSP is still strongly NP-hard. In this section, we study the case where the variable cost parameter P_{ij} is additive with respect to the product index i and the supplier index j, that is, $P_{ij} = \tilde{P}_i + \hat{P}_j$. We show a polynomial-time algorithm for this case.

We first study the product-specific component of the variable purchase cost, i.e., $\tilde{P}_i$. The total cost associated with the product-specific components of the variable purchase cost incurred during the whole planning horizon does not depend on the decision of which suppliers to purchase from. Since it is optimal to have no inventory at the end of the planning horizon, the total amount of product i purchased during the planning horizon is the same as the total demand of product i between periods 1 and T, i.e., $\sum_t \sum_j x_{ij}^t = \sum_t D_i^t$. Thus, the total product-specific component of the variable cost is

$$\sum_t \sum_j \sum_i \tilde{P}_i x_{ij}^t = \sum_i \tilde{P}_i \sum_t \sum_j x_{ij}^t = \sum_i \tilde{P}_i \sum_t D_i^t = \sum_i \sum_t \tilde{P}_i D_i^t,$$

which is independent of the decision variables, and therefore, it can be omitted in the analysis. Thus, we proceed by assuming

$$P_{ij} = \hat{P}_j \qquad \text{for any } i \in \{1, \ldots, I\}. \tag{1}$$

We index products $\{1, \ldots, I\}$ based on non-increasing unit holding costs, i.e.,

$$H_1 \geq H_2 \geq \cdots \geq H_I. \tag{2}$$

Note that we do not need to consider supplier j if there exists supplier j' satisfying $O_{j'} \leq O_j$ and $\hat{P}_{j'} \leq \hat{P}_j$; in that case, supplier j' dominates j in terms of both the fixed cost and the purchase cost. Without loss of generality, we proceed by assuming

$$(O_{j'} - O_j) \cdot (\hat{P}_{j'} - \hat{P}_j) < 0 \quad \text{for every distinct pair } j, j' \in \{1, \ldots, J\}. \tag{3}$$

With this assumption, it can be shown that at most one supplier is used in each period under the optimal policy. If we order from two or more suppliers in a period, the total cost could have been reduced by grouping some orders. This property is formally stated below.

Proposition 1. *Suppose (1) and (3) hold. Then, there exists an optimal solution such that at most one supplier is used in each period, i.e., for each t,* $\sum_j y_j^t \in \{0, 1\}$.

Proof. We prove it by contradiction. Suppose that in an optimal solution, there are two products, say products 1 and 2, to be purchased from distinct suppliers j and j' respectively in some period t. We note that a product cannot be ordered from different suppliers in the same period, as discussed in Section 2. Suppose strictly positive amounts have been purchased for these products in period t. Without loss of generality, assume $\hat{P}_j \leq \hat{P}_{j'}$. Then, (3) implies $\hat{P}_j < \hat{P}_{j'}$. Thus, we can eliminate the total amount of product 2 purchased from supplier j' and instead purchase it from supplier j. By doing so, the total variable purchase cost decreases, contradicting the optimality of the solution. Therefore we conclude that at most one supplier is used in any period. □

The next proposition shows that products $\{1, \ldots, i\}$ should not be carried into a period in which product i is ordered. For convenience, we define $z_i^t = 0$ for $t = 0$.

Proposition 2. *Suppose (1) and (2) hold. Then, there exists an optimal solution such that if product* $i \in \{1, \ldots, I\}$ *is purchased in period* $t \in \{1, \ldots, T\}$*, then there should be no inventory for products* $\{1, 2, \ldots, i\}$ *carried into period t, i.e.,* $z_{i'}^{t-1} \cdot x_{ij}^t = 0$ *holds for any product* $i' \in \{1, \ldots, i\}$ *and any supplier* $j \in \{1, \ldots, J\}$.

Proof. Consider the optimal solution with the least number of strictly positive z_i^t variables. Suppose that, by way of contradiction, product i is purchased from supplier j in period t, and inventory of product $i' \in \{1, \ldots, i\}$ is carried into period t, i.e., $x_{ij}^t > 0$ and $z_{i'}^{t-1} > 0$. By the product-specif zero-inventory-ordering property given in [2], product i and i' must be distinct, implying, $i' < i$. Since $z_{i'}^{t-1} > 0$, there exists at least one period before t with a positive order for product i'. Let period s be the last such period before t, and let j' denote the supplier used in period s.

Since product i' is not purchased in period t, the cost incurred by purchasing one unit of product i' in period s and carrying it into period t must be smaller than the cost of purchasing it in period t, i.e., $\hat{P}_{j'} + (t - s)H_{i'} < \hat{P}_j$; otherwise, we can either reduce the total cost or obtain another optimal solution with fewer strictly positive z_i^t variables. Similarly, since product i is purchased in period t, we can show $\hat{P}_{j'} + (t - s)H_i \geq \hat{P}_j$.

These two inequalities above imply $H_{i'} < H_i$, where $i' < i$. This contradicts the indexing of products given in (2), i.e., $H_1 \geq H_2 \geq \cdots \geq H_I$. Based on this contradiction, we conclude that the inventory of products $\{1, \ldots, i\}$ should not be carried into this period in the optimal solution with the least number of strictly positive z_i^t variables. □

An implication of the structure of the optimal policy given in Proposition 2 is that product 1 is not carried into a period in which any order takes place. In other words, whenever a positive quantity of some product is ordered, product 1 is also ordered. However, it is possible that only product 1 is ordered in a period. The property shown in Proposition 2 is a type of zero-inventory-ordering property. It is stronger than the *product-specific* zero-inventory-ordering property given in [2] since it factors in the ordering pattern of other products. However, it is weaker than the *complete* zero-inventory-ordering property in which every product is ordered every time an order is placed. It shows that if product i is ordered in a period, products $\{1, \ldots, i\}$ should not be carried into this period. In other words, whenever product i is ordered, all products with lower indices are also ordered. We refer to this as an *echelon* zero-inventory-ordering property.

Note that we are the first to present such a type of nuanced generalization of the classical zero-inventory property proposed by [17], which allows us to restrict to $O(I)$ sets of product in each period, rather than 2^I possible combinations of products. With the multiple-product multiple-supplier extensions considered in the literature, the *complete* zero-inventory-ordering property no longer holds, and the *product-specific* zero-inventory-ordering property is too weak to be algorithmically useful. Thus, other structural properties are needed to solve the problem. The structural property that we have identified here, the *echelon* zero-inventory-ordering property, is one of such properties that enable us to construct a new approach to the dynamic programming algorithm extending the Wagner-Whitin algorithm.

The key idea in our dynamic programming approach is how we make the problem smaller in the divide-and-conquer step. We divide the problem not only by partitioning time periods but also by partitioning products. The classical Wagner-Whitin algorithm and its extensions in the literature use partitioning based on time periods only (see, e.g., [7,18]). In other words, existing algorithms make the problem smaller by considering iteratively shorter time intervals, but in each interval, all products are considered. Our approach is different. A smaller problem is defined not only by the time interval but also by a subset of products to be considered. When we partition products, we use the *echelon* zero-inventory-ordering property. For some products, inventory is carried over the entire interval under consideration, and the cost associated with these products can be easily computed by adding the cost of purchasing at the beginning and the cost of holding inventory over the interval. For the remaining products, we can apply the *echelon* zero-inventory-ordering property to one of the periods in the given interval, enabling us to partition the time interval into smaller intervals. This forms the basis for our recursive formulas.

We use the following notations. For $\ell, i \in \{1, \dots, I\}$ where $\ell \le i$, and $s, t \in \{1, \dots, T\}$ where $s \le t$, define:

- $D_i^{s,t}$: cumulative demand for product i from period s to period t, i.e., $D_i^{s,t} = \sum_{\tau=s}^{t} D_i^{\tau}$;
- $D_{\ell,i}^{s,t}$: total demand for products $\{\ell, \dots, i\}$ from period s to period t, i.e., $D_{\ell,i}^{s,t} = \sum_{k=\ell}^{i} D_k^{s,t}$;
- $\overline{H}_i^{s,t}$: total holding cost for product i from period s to period t if an order is placed in period s to cover demand until period t, i.e., $\overline{H}_i^{s,t} = H_i \sum_{\tau=s}^{t} (\tau - s) D_i^{\tau}$;
- $\overline{H}_{\ell,i}^{s,t}$: total holding cost for products $\{\ell, \dots, i\}$ from period s to period t if an order is placed in period s to cover demand until period t, i.e., $\overline{H}_{\ell,i}^{s,t} = \sum_{k=\ell}^{i} \overline{H}_k^{s,t}$.

The above quantities depend on exogenous parameters only and can be computed easily. The quantities that are determined in our dynamic programming algorithm are as follows. Let $i \in \{1, \dots, I\}$, $j \in \{1, \dots, J\}$, and $s, t \in \{1, \dots, T\}$ where $s \le t$. All of these quantities assume zero inventory of products $\{1, \dots, i\}$ at the beginning of period s and also at the end of period t.

- $C_i^{s,t}$: minimum total cost for products $\{1, \dots, i\}$ over periods $\{s, \dots, t\}$;
- $C_{ij}^{s,t}$: minimum total cost for products $\{1, \dots, i\}$ over periods $\{s, \dots, t\}$ with an order from supplier j in period s;
- $L_{ij}^{s,t}$: minimum total cost for products $\{1, \dots, i\}$ over periods $\{s, \dots, t\}$ with an order from supplier j in period s and strictly positive inventory of at least one product at the end of every period except t.

Note that the optimal cost that we consider in this problem is $C_I^{1,T}$. From Proposition 1, at most one supplier is used in period s, and we have

$$C_i^{s,t} = \min_{1 \le j \le J} C_{ij}^{s,t}. \tag{4}$$

For the remainder of this section, we calculate $C_{ij}^{s,t}$ for given i, j and $s \le t$.

If $s = t$, $L_{ij}^{s,t}$ and $C_{ij}^{s,t}$ are single-period costs, and we have

$$L_{ij}^{s,s} = C_{ij}^{s,s} = O_j + \hat{P}_j \cdot D_{1,i}^{s,s}.$$

Also, if $i = 1$, $L_{ij}^{s,t}$ is the cost associated with a single product i only, and thus we have

$$L_{1j}^{s,t} = O_j + \hat{P}_j \cdot D_1^{s,t} + \overline{H}_1^{s,t}, \tag{5}$$

since there is no order for product 1 in any periods from $s+1$ to t.

Next, we present recursion relations to compute $L_{ij}^{s,t}$ and $C_{ij}^{s,t}$ quantities, where $s < t$. We consider the following two cases separately.

Case 1. In the first case we consider, there is a period between s and $t-1$ in which the ending inventory is zero for all products in $\{1, \dots, i\}$. Let r denote the first such period. If $r = s$, the total cost incurred in period s is $L_{ij}^{s,r} = L_{ij}^{s,s}$. If $r > s$, then there exists a positive amount of inventory in every period between s and $r-1$, implying that the cost associated with products $\{1, \dots, i\}$ incurred between periods s and t is $L_{ij}^{s,r}$. The minimum total cost from periods $r+1$ and t is given by $C_i^{r+1,t}$. Thus, the optimal cost for products $\{1, \dots, i\}$ from period s to t is

$$C_{ij}^{s,t} = \min_{s \le r < t} L_{ij}^{s,r} + C_i^{r+1,t}. \tag{6}$$

This is the case where the *complete* zero-inventory-ordering property takes place in some period $r+1$ between periods $s+1$ and t, where the product space is restricted to $\{1, \dots, i\}$. In an example illustrated in Fig. 1, with $i = 5$, $s = 1$, $t = 10$ and $r = s+3$, the *complete* zero-inventory-ordering property holds in period $s+4$, where all products in $\{1, \dots, i\}$ are ordered, and this enables us to divide the interval $\{s, \dots, t\}$ into two shorter ones $\{s, \dots, s+3\}$ and $\{s+4, \dots, t\}$.

Case 2. In the second case we consider, there is a positive end-of-period inventory of at least one product in $\{1, \dots, i\}$ in every period between s and $t-1$. The minimum total cost for products $\{1, \dots, i\}$ over periods $\{s, \dots, t\}$ is then given by $L_{ij}^{s,t}$, i.e.,

$$C_{ij}^{s,t} = L_{ij}^{s,t}. \tag{7}$$

The value of $L_{ij}^{s,t}$ with $i = 1$ is given above in equation (5). To compute $L_{ij}^{s,t}$ for $i > 1$, we consider the following two subcases.

The first subcase corresponds to the event that none of the products $\{1, \dots, i\}$ are ordered between $s+1$ and t. Then, the corresponding cost $L_{ij}^{s,t}$ is

$$O_j + \hat{P}_j \cdot D_{1,i}^{s,t} + \overline{H}_{1,i}^{s,t}. \tag{8}$$

For the second subcase, we proceed by assuming that there exists at least one order between $s+1$ and t. Let ℓ be the product with the highest index such that a positive order of products $\{1, \dots, \ell\}$ is placed between $s+1$ and t. Let s' be the first period index in $\{s, s+1, \dots, t\}$ in which the end-of-period inventory of products $\{1, \dots, \ell\}$ becomes zero. Note $1 \le \ell < i$ and $s \le s' < t$.

- Consider products $\{1, \dots, \ell\}$. Since there exists a positive amount of inventory in every period between s and $s'-1$ by the definition of s', the minimum total cost from period s to s' is $L_{\ell j}^{s,s'}$. Note that $L_{\ell j}^{s,s'}$ includes the fixed ordering cost in period s. The minimum total cost incurred from period $s'+1$ to t is $C_{\ell}^{s'+1,t}$.
- Consider products $\{\ell+1, \dots, i\}$. Since these products are not ordered between period $s+1$ and t, they are purchased in period s in sufficient quantities to cover demand until period t. The sum of the purchase cost in period s and the holding cost between periods s and t is $\hat{P}_j \cdot D_{\ell+1,i}^{s,t} + \overline{H}_{\ell+1,i}^{s,t}$. (The fixed cost of ordering in period s is already factored into $L_{\ell j}^{s,s'}$ above.)

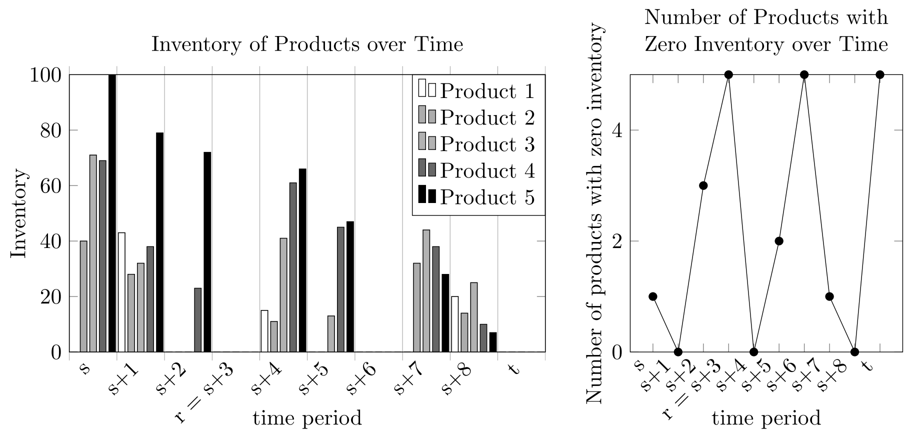

Fig. 1. An Example with $i=5$, $s=1$, $t=10$ and $r=s+3$.

By choosing the best s' and ℓ, the optimal cost under the second subcase is

$$\min_{s\le s'<t}\ \min_{1\le \ell<i}\ \left\{L_{\ell j}^{s,s'}+C_{\ell}^{s'+1,t}\right\}+\left\{\hat{P}_j\cdot D_{\ell+1,i}^{s,t}+\overline{H}_{\ell+1,i}^{s,t}\right\}. \quad (9)$$

Hence, from considering the two subcases above, we have

$$L_{ij}^{s,t}=\text{minimum of (8) and (9)}. \quad (10)$$

Furthermore, combining (6) and (7) and recalling (4), we summarize as follows:

$$C_{ij}^{s,t}=\min_{s\le r\le t} L_{ij}^{s,r}+C_i^{r+1,t} \qquad \text{and} \qquad C_i^{s,t}=\min_{1\le j\le J} C_{ij}^{s,t} \quad (11)$$

where we conveniently define $C_i^{s,t}=0$ whenever $s>t$.

It can be verified that the values of $D_i^{s,t}$, $\overline{H}_i^{s,t}$, $D_{\ell,i}^{s,t}$ and $\overline{H}_{\ell,i}^{s,t}$ can be computed in $O(I^2T^2)$ time. Furthermore, it takes $O(I^2JT^3)$ time to compute $C_{ij}^{s,t}$, $L_{ij}^{s,t}$ and $C_i^{s,t}$ values. Thus, the computational complexity of our algorithm is $O(I^2JT^3)$.

Theorem 3. *If (1) holds, there exists a polynomial-time algorithm for solving MLSSP.*

4. General MLSSP problem: heuristic algorithm

4.1. Description

In this section, we consider the general problem and develop a heuristic for it based on the algorithm proposed for the special case in Section 3. In the general problem, we take into account the product-specific unit purchase price P_{ij} rather than separable $P_{ij}=\tilde{P}_i+\hat{P}_j$ that we have considered in Section 3.

We first illustrate by an example that key properties identified in Section 3 do not continue to hold for the general MLSSP problem. We note that Proposition 1 does not necessarily hold for the general problem, i.e., two separate orders may be placed from different suppliers in a given period, which can be illustrated by the example below. Note also that in an optimal solution to the general problem, every supplier may offer different unit purchase prices for different products, and separate orders may be placed from different suppliers in a period. Thus, the ordering policy for the general problem may not have an easily identifiable structure such as the one given by the special case in Section 3, and the *echelon* zero-inventory-ordering property may not hold for our problem, as shown in the following example.

Example 4.1. Consider a four-period problem with two products and two suppliers, where $O_1=4$, $O_2=5$, $P_{11}=P_{22}=0$, $P_{12}=P_{21}=10$, $H_1=3$, $H_2=2$ and $D_1^t=D_2^t=1$, $t=1,\ldots,4$.

One can check that both suppliers should be used in any period when both products are ordered under the optimal policy, implying that Proposition 1 does not hold. This is because of the relatively high cost of ordering product 1 (respectively, 2) from supplier 2 (respectively, 1). Moreover, we can deduce that any product i, if purchased, should be ordered from supplier $j=i$. Then, product 1 and 2 can be considered separately. It is thus not surprising to find that the *echelon* zero-inventory-ordering property does not hold in period 4, where only product 2 is purchased under the optimal policy.

Based on these observations, we develop our heuristic in the following way. Since considering every combination of suppliers in every period is computationally expensive, we allow up to a specific number of suppliers in every period in our heuristic. Specifically, a fixed k is exogenously given, and we allow up to k suppliers in every period. Furthermore, with regard to the product set to be ordered, we first sequence the products, which is an extension of (2) in Section 3, and then assume that the *echelon* zero-inventory-ordering property holds with this sequence of products in the solutions given by our heuristic. Under these assumptions, we can apply dynamic programming to generate an efficient heuristic for our general problem similar to the algorithm proposed in Section 3. The proposed heuristic runs in polynomial time.

4.2. Heuristic algorithm

In the heuristic, we still use the *echelon* zero-inventory-ordering property to develop the recursion formulas. Note that we have considered only one supplier used in each period for the special case in Section 3, but we need to consider the supplier set used in each period for the general problem under the assumption of allowing up to a specific number of suppliers. Thus we modify the algorithm proposed in Section 3 to develop our heuristic. As stated below, there are two phases in our heuristic.

Phase 1. The first phase is the sequencing phase to determine the sequence of the products when $I\ge 2$, which plays a crucial role in the *echelon* zero-inventory-ordering property. In this phase, we first consider each product in turn and solve the single-product lot-sizing with supplier selection problem. Note that Jaruphongsa et al. [12] consider a dual-sourcing version of this problem and propose a dynamic programming algorithm for it based on the *product-specific* zero-inventory-ordering property, which can be easily extended to solve our problem with supplier selection. We then calculate the number of orders under the optimal policy for each product and initialize by indexing products with a non-increasing number of orders. Here we expect a product with more orders in the single-product problem would still be ordered more in the multi-product problem, which would have a smaller index according to the *echelon* zero-inventory-ordering property. This is not necessarily the best sequence of products, and thus, our final step adjusts the sequence to achieve a better performance in the second phase. Specifically, for i proceeding from 1 to $I-1$, we swap products i and $i+1$ if the minimum total cost given by the second phase with $k=1$ can be reduced by doing so. This provides a new sequence of products.

Phase 2. For the second phase, fix k where $1 \le k \le \min\{I, J\}$. Let $\{\Omega_m,\ m=1,\dots,M\}$ be all possible supplier sets when allowing up to k suppliers in every period. Then

$$M = \binom{J}{1} + \dots + \binom{J}{k}. \tag{12}$$

If ordered from a supplier set Ω_m, product i must be purchased from the supplier with the least unit purchase price for it and we denote the supplier by $j(m,i)$. Thus, $P_{i,j(m,i)} = \min_{j\in\Omega_m} P_{ij}$.

Now, we discuss how we adapt the dynamic programming formulation given in Section 3. For integers $i \in \{1,\dots,I\}$, $m \in \{1,\dots,M\}$ and $s,t \in \{1,\dots,T\}$ where $s \le t$, we consider $\hat{C}_{im}^{s,t}$ and $\hat{L}_{im}^{s,t}$ rather than $C_{ij}^{s,t}$ and $L_{ij}^{s,t}$. These quantities assume zero inventory of products $\{1,\dots,i\}$ at the beginning of period s and also at the end of period t.

- $\hat{C}_{im}^{s,t}$: minimum total cost for products $\{1,\dots,i\}$ over periods $\{s,\dots,t\}$ with orders from supplier set Ω_m in period s;
- $\hat{L}_{im}^{s,t}$: minimum total cost for products $\{1,\dots,i\}$ over periods $\{s,\dots,t\}$ with orders from supplier set Ω_m in period s and strictly positive inventory of at least one product at the end of every period except t.

Similar to (8) and (9) for the special case in Section 3, we can obtain the value of $\hat{L}_{im}^{s,t}$ by computing the minimum total cost in the subcase corresponding to the event that none of the products $\{1,\dots,i\}$ are ordered between $s+1$ and t,

$$\sum_{j\in\Omega_m} O_j + \sum_{i'=1}^{i} P_{i',j(m,i')} \cdot D_{i'}^{s,t} + \overline{H}_{1,i}^{s,t}, \tag{13}$$

and the one when there exists at least one order between $s+1$ and t,

$$\min_{s\le s'<t}\ \min_{1\le \ell<i}\ \{\hat{L}_{\ell m}^{s,s'} + C_{\ell}^{s'+1,t}\} + \{\sum_{i'=\ell+1}^{i} P_{i',j(m,i')} \cdot D_{i'}^{s,t} + \overline{H}_{\ell+1,i}^{s,t}\}. \tag{14}$$

Similar to the formulas (10) and (11), we have the recursion formulas below:

$$\hat{L}_{im}^{s,t} = \text{minimum of (13) and (14)},$$
$$\hat{C}_{im}^{s,t} = \min_{s\le r\le t} \hat{L}_{im}^{s,r} + C_i^{r+1,t} \qquad \textit{and}$$
$$C_i^{s,t} = \min_{1\le m\le M} \hat{C}_{im}^{s,t},$$

where $\hat{L}_{im}^{s,s} = \hat{C}_{im}^{s,s} = \sum_{j\in\Omega_m} O_j + \sum_{i'=1}^{i} P_{i',j(m,i')} \cdot D_{i'}^{s}$ for any i, m, s, and $\hat{L}_{1m}^{s,t} = \sum_{j\in\Omega_m} O_j + P_{1,j(m,1)} \cdot D_1^{s,t} + \overline{H}_1^{s,t}$ for any m and $s<t$. Note that when considering the optimal solution over the interval $\{s,\dots,t\}$, we consider a supplier set rather than a single supplier in period s, which is the only difference between these formulas and those stated in Section 3.

Table 1
Distributions to Generate Parameters.

Parameter	Distribution
Fixed ordering cost	*int*[1000, 2000]
Unit purchase price	*int*[20, 50]
Unit holding cost	$U[1,5]$
Demand	*int*[1, 200]

Notes. $U[a,b]$ refers to the uniform distribution between a and b, and $int[a,b]$ refers to the uniform integer distribution between a and b (including both a and b).

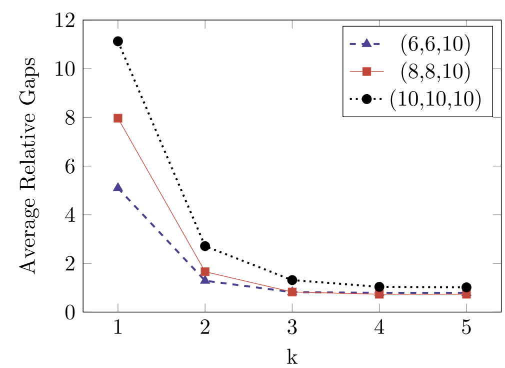

Fig. 2. Average Relative Gaps (%) with Different k. *Notes.* Each curve depicts the average relative gaps between the results given by the heuristics and the optimal ones given by CPLEX for 20 instances. $(I; J; T)$ refers to the problem with I products, J suppliers, and T time periods.

We can easily construct an algorithm based on the recursion formulas above. The algorithm runs in $O(I^2MT^3)$, where M, as defined in (12), is a polynomial function of the number of suppliers J with degree k for relatively small k.

4.3. Computational experiments

In this section, we conduct computational experiments to evaluate the performance of the proposed heuristic algorithm. After showing how the result of the heuristic changes as the maximum number of suppliers allowed, denoted by k, increases, we compare the performance of our heuristic with that of the heuristic proposed by [2], which we denote by *B&L*. Problem instances are generated in a similar way to those in [2], and distributions to generate parameters are presented in Table 1.

One way to assess the quality of our heuristic is to compute relative gaps between heuristic performance and the optimal values, and we first consider cases in which the optimal solutions can be obtained by CPLEX within 10 hours. We solve the problem using our heuristic with a fixed k described in Section 4.2. Our solution is an upper bound of the optimal cost since it is a feasible solution to the general problem, and we can find the average relative gaps over 20 instances for each problem size.

The heuristic policy in Section 4.2 does not specify the value of k, the maximum number of suppliers allowed in each period. We first consider how our heuristic performs as we change k. In Fig. 2, we plot the average relative gaps of our heuristic versus k for three sizes of problem. We observe that the average relative gap for each case decreases as k increases, which is as expected. As more suppliers are allowed in each period, the results given by our heuristic improve. We also notice that each curve is convex in k, which implies that the marginal benefit of an additional supplier diminishes as k increases. This makes it reasonable for us to choose a proper k, for example, $k=3$ in the problems tested in Fig. 2.

Then we compare the performance of our heuristic to *B&L*. We use *B&L* as the benchmark since this is the only algorithm that runs in polynomial time. There are two phases in *B&L*. In the

Table 2
Performance of Our Heuristic Compared to *B&L*.

Problem Size (I, J, T)	Our Heuristic			*B&L*		
	RG (wo., %)	RG (w., %)	Time (s)	RG (wo., %)	RG (w., %)	Time (s)
(3,3,10)	0.24	0.24	0.02	0.60	0.58	0.01
(3,3,15)	0.26	0.26	0.03	0.59	0.43	0.01
(4,4,10)	0.40	0.39	0.02	1.26	1.05	0.01
(4,4,15)	0.34	0.34	0.04	1.25	0.84	0.02
(5,5,10)	0.63	0.56	0.02	1.31	1.06	0.01
(5,5,20)	0.69	0.63	0.08	1.27	0.92	0.02
(8,8,15)	0.76	0.70	0.08	1.81	1.36	0.02
(8,8,20)	0.94	0.88	0.12	2.08	1.48	0.03
(10,10,15)	1.38	1.25	0.12	2.21	1.57	0.03
(10,10,20)	1.16	1.06	0.20	2.05	1.36	0.05
(10,10,50)	1.59	1.41	5.04	2.25	1.66	0.22

Notes. (I, J, T) refers to the problem with I products, J suppliers, and T time periods. *RG* refers to the average relative gaps between the results of the heuristics and the solutions given by CPLEX within 10 hours. *Time* refers to the average running time of the heuristics with the improvement phase. *wo.* and *w.* in the parathesis refer to the results of heuristics without and with the improvement phase, respectively. Note that except for 3 out of 20 instances with problem size $(10, 10, 50)$, all optimal values can be found by CPLEX within 10 hours. Computations were done on an Intel Core i5-7300U CPU at 2.60 GHz with 15.80 GB RAM.

first phase, they consider each product in turn and allocate it to a unique supplier over the whole planning horizon such that the total cost of it and products that have been allocated are minimal. In the second phase (the improvement phase), they check every inventory or purchased quantity in every period to see if it can be ordered from other suppliers with positive orders in this period and improve the solution. Once our solution is obtained, we can optionally perform a local search to possibly improve the solution similar to the second phase in *B&L*. We note that Cárdenas-Barrón et al. [6] propose a heuristic based on a “reduce and optimize” approach and show that the heuristic outperforms *B&L*. More specifically, they iteratively construct reduced feasible sets of binary variables and optimize the problem over those sets using CPLEX, with the initial set given by the Wagner-Whitin algorithm. This heuristic does not run in polynomial time since it can have as many integer variables as the optimal formulation. In fact, it runs slower than the optimal CPLEX solution for the problem sizes we consider. Thus we do not include it in our numerical analysis.

For these problems, we choose $k = 3$ and the results are shown in Table 2. We observe that our heuristic, even without the improvement phase, outperforms *B&L*. All of our 11 average relative gaps of 20 instances are smaller than those of their heuristic with and without the improvement phase, and all relative gaps between our solutions and the optimal ones are within 4.29%, which means that our algorithm works well; if adding the improvement phase, our heuristic performs better, and the relative gaps are within 3.43%. We note that our heuristic takes a longer time than *B&L*, but practically it is still reasonably fast.

4.3.1. Impact of increasing fixed ordering costs

In this section, we compare the performance of our heuristic and *B&L* under different values of the fixed ordering costs. More specifically, we generate $\{O_j\}_{j=1}^J$ from the distribution given in Table 1 and consider various fixed ordering costs in the form of $\{2^{n-1}O_j\}_{j=1}^J$, when the integer n varies from 1 to 5.

An example is shown in Fig. 3 to illustrate how the average relative gaps change as the fixed ordering costs increase. We can see that our heuristic, even without the improvement phase, works significantly better than *B&L* when the ordering fixed costs are large. The average relative gaps of our heuristic remain relatively stable under different fixed ordering costs, and the gaps are within 1.38% and 1.24% for the heuristic without and with the improvement phase, respectively. However, the average relative gaps of *B&L* increase greatly as the fixed costs increase, and the largest one is 8.31% when the fixed costs are $\{16O_j\}_{j=1}^J$.

To explain this phenomenon, we analyze the average number of orders placed by the optimal solutions and find that orders are placed frequently under $\{O_j\}_{j=1}^J$, with 1.85 orders per period. As the fixed ordering costs increase, the orders become more infrequent, and only the average of 0.7 orders per period are placed under $\{16O_j\}_{j=1}^J$. This implies that the idea behind *B&L*, i.e., ordering one product from one supplier over the entire horizon, leads to a larger loss as the fixed ordering costs increase and orders should be placed less frequently. In contrast, the performance of our heuristics remains relatively stable as the fixed ordering costs change. One can expect a more significant improvement given by our heuristics under higher fixed ordering costs.

4.3.2. Larger problems

We now consider larger problems where the optimal solutions for most cases cannot be obtained within 10 hours. Below, we briefly describe how we adapt our heuristic for solving large problems and discuss its performance compared to *B&L*.

We have noted earlier that there are exponentially many subsets of suppliers as the number of suppliers increases. In the heuristic policy described in Section 4.2, we limit the set of suppliers that we consider by limiting the maximum size of the suppliers in a period, which we denote by k. With an increased number of suppliers, even this approach leads to a larger number of suppliers, and thus, we take another approach to look for a set of suppliers in each period. The main idea is to partition products into smaller groups, and we assign a set of suppliers to each group of products. Then we apply our heuristic policy with $k = 3$ to each group.

To assess the performance of our improved heuristic, we test it for problem sizes considered in [2] and compute the relative gaps between our improved heuristic and theirs. The average relative gaps and average running times for 20 instances are shown in Table 3. The negative relative gap indicates that our heuristic policy outperforms *B&L*.

From Table 3, we observe that all average relative gaps are between −2.09% and −1.12%. The running times of our improved heuristic are within 12 minutes in average, which exceed the times of *B&L* due to the efforts on reducing the total cost.

5. Conclusion

In this paper, we have introduced the *echelon* zero-inventory-ordering property for a special case of MLSSP where the variable cost parameter is additive. As a generalization of the classical zero-inventory-ordering property, it can be used to develop a dynamic programming algorithm and solve the special case in a polynomial time and it may solve a lot-sizing problem not satisfying the *complete* zero-inventory-ordering property.

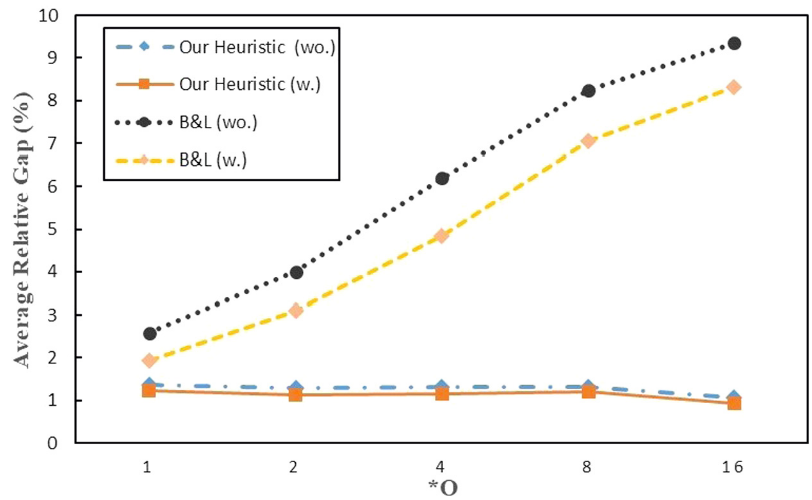

Fig. 3. Performance under Different Fixed Ordering Costs. *Notes*. Each curve depicts the average relative gaps between the results given by the heuristics and the optimal ones given by CPLEX for 20 instances. *wo*. and *w*. in the parathesis refer to the results of heuristics without and with the improvement phase. There are 10 products, 10 suppliers, and 10 periods in the example we consider.

Table 3
Performance for Large-Sized Problems.

Problem Size (I, J, T)	Relative Gap (%)			Time (s)	
	wo. VS. wo.	wo. VS. w.	w. VS. w.	Our Heuristic	*B&L*
(15,15,100)	-2.03	-1.30	-1.51	28.62	1.00
(20,20,100)	-2.09	-1.22	-1.40	37.74	1.04
(20,20,200)	-2.05	-1.12	-1.37	203.56	3.33
(50,50,200)	-2.07	-1.12	-1.22	635.85	11.91

Notes. (I, J, T) refers to the problem with I products, J suppliers, and T time periods. *Relative Gap* refers to the average relative gap between our heuristic and *B&L*. *wo.* and *w.* refer to the results of heuristics without and with the improvement phase. For example, "*wo. VS. w.*" means that our heuristic did not have the improvement phase while *B&L* did.

Our approach for the special case may not be easily extended to solve problems with time-varying variable costs. Nevertheless, our heuristic for the general problem can be easily modified to incorporate time-varying variable costs, and still runs in polynomial time. We believe the idea of the *echelon* zero-inventory-ordering property and allowing up to a specific number of suppliers can be applied in more complicated practical settings.

Acknowledgements

We thank the editor-in-chief, the associate editor, and two anonymous reviewers for their helpful and constructive suggestions, which help improve the paper significantly. A significant portion of this research was conducted when Meichun Lin was an undergraduate student supported by Mitacs, a non-profit organization based in Canada, through the Globalink Research Award. Woonghee Tim Huh acknowledges support from the NSERC Discovery Grants (RGPIN 2020-04213) and the Canada Research Chairs Program. Guohua Wan acknowledges support from NSF of China (71421002) and Shanghai Subject Chief Scientist Program (16XD1401700).

Appendix. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.orl.2021.05.013.

References

[1] E. Arkin, D. Joneja, R. Roundy, Computational complexity of uncapacitated multi-echelon production planning problems, Oper. Res. Lett. 8 (2) (1989) 61–66.
[2] C. Basnet, J.M.Y. Leung, Inventory lot-sizing with supplier selection, Comput. Oper. Res. 32 (1) (2005) 1–14.
[3] G.R. Bitran, H.H. Yanasse, Computational complexity of the capacitated lot size problem, Manag. Sci. 28 (10) (1982) 1174–1186.
[4] N. Brahimi, N. Absi, S. Dauzère-Pérès, A. Nordli, Single-item dynamic lot-sizing problems: an updated survey, Eur. J. Oper. Res. 263 (3) (2017) 838–863.
[5] N. Brahimi, S. Dauzere-Peres, N.M. Najid, A. Nordli, Single item lot sizing problems, Eur. J. Oper. Res. 168 (1) (2006) 1–16.
[6] L.E. Cárdenas-Barrón, J.L. González-Velarde, G. Treviño-Garza, A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem, Comput. Oper. Res. 64 (2015) 225–232.
[7] S. Chand, V.N. Hsu, S. Sethi, V. Deshpande, A dynamic lot sizing problem with multiple customers: customer-specific shipping and backlogging costs, IIE Trans. 39 (11) (2007) 1059–1069.
[8] M. Firouz, B.B. Keskin, S.H. Melouk, An integrated supplier selection and inventory problem with multi-sourcing and lateral transshipments, Omega 70 (2017) 77–93.
[9] M. Florian, J.K. Lenstra, A. Rinnooy Kan, Deterministic production planning: algorithms and complexity, Manag. Sci. 26 (7) (1980) 669–679.
[10] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to Computational Complexity, W.H. Freeman, New York, 1979.
[11] K. Jakob, P.M. Pruzan, The simple plant location problem: survey and synthesis, Eur. J. Oper. Res. 12 (1983) 36–81.
[12] W. Jaruphongsa, S. Cetinkaya, C.-Y. Lee, A dynamic lot-sizing model with multi-mode replenishments: polynomial algorithms for special cases with dual and multiple modes, IIE Trans. 37 (5) (2005) 453–467.
[13] M. Khouja, S. Goyal, A review of the joint replenishment problem literature: 1989–2005, Eur. J. Oper. Res. 186 (1) (2008) 1–16.
[14] R. Kian, U. Gürler, E. Berk, The dynamic lot-sizing problem with convex economic production costs and setups, Int. J. Prod. Econ. 155 (2014) 361–379.
[15] J. Rezaei, M. Davoodi, Multi-objective models for lot-sizing with supplier selection, Int. J. Prod. Econ. 130 (1) (2011) 77–86.
[16] W. Van den Heuvel, A.P.M. Wagelmans, Worst-case analysis for a general class of online lot-sizing heuristics, Oper. Res. 58 (1) (2010) 59–67.
[17] H.M. Wagner, T.M. Whitin, Dynamic version of the economic lot size model, Manag. Sci. 5 (1) (1958) 89–96.
[18] W.I. Zangwill, A deterministic multi-period production scheduling model with backlogging, Manag. Sci. 13 (1) (1966) 105–119.
[19] M. Zhao, M. Zhang, Multiechelon lot sizing: new complexities and inequalities, Oper. Res. 68 (2) (2020) 534–551.