Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Multiechelon Lot Sizing: New Complexities and Inequalities

Ming Zhao, Minjiao Zhang

informs
http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–18
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Multiechelon Lot Sizing: New Complexities and Inequalities

Ming Zhao,[a] Minjiao Zhang[b]

[a] Lerner College of Business and Economics, University of Delaware, Newark, Delaware 19716; [b] Department of Economics, Finance, and Quantitative Analysis, Michael J. Coles College of Business, Kennesaw State University, Kennesaw, Georgia 30144

Contact: mzhao@udel.edu, https://orcid.org/0000-0002-2664-5267 (MZhao); mzhang16@kennesaw.edu (MZhang)

Received: May 18, 2017
Revised: February 1, 2018; June 11, 2018; November 10, 2018
Accepted: March 8, 2019
Published Online in Articles in Advance: February 18, 2020

Subject Classifications: analysis of algorithms: computational complexity; programming: integer, algorithms, cutting plane
Area of Review: Optimization

https://doi.org/10.1287/opre.2019.1867

Abstract. We study a multiechelon lot-sizing problem for a serial supply chain that consists of a production level and several transportation levels, where the demands can exist in the production echelon as well as in any transportation echelons. With the presence of stationary production capacity and general cost functions, our model integrates production, inventory, and transportation decisions and generalizes existing literature on many multiechelon lot-sizing models. First, we answer an open question in the literature by showing that multiechelon lot sizing with intermediate demands (MLS) is NP-hard. Second, we develop polynomial time algorithms for both uncapacitated and capacitated MLS with a fixed number of echelons. The run times of our algorithms improve on those of many known algorithms for different MLS models. Third, we present families of valid inequalities for MLS that generalize known inequalities. For the uncapacitated case, we develop a polynomial-time separation algorithm and efficient separation heuristics. Finally, we demonstrate the effectiveness of a branch-and-cut algorithm using proposed inequalities to solve large multi-item MLS problems.

Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2019.1867.

Keywords: lot sizing • dynamic programming • network flow • polynomial-time algorithms • valid inequalities

1. Introduction

In production planning, the economic lot-sizing problem considers a manufacturer that produces and stores a single product to meet time-varying demand over a given finite planning horizon. The problem is to determine production quantities for each period such that all demands are satisfied on time with minimum total production and inventory holding costs. The production plan is of great importance owing to the balance between the fixed setup cost and the inventory holding cost. Production quantity and the resulting inventory level affect the product availability and customer service level, making them key elements for business competitiveness.

If production planning only happens at one level without involving transportation decisions, the associated lot-sizing problem is a single-echelon problem, where a level is referred to as an *echelon*. However, in practice, it is common to see a multilevel or multiechelon serial supply chain integrating production, inventory, and transportation decisions (van Hoesel et al. 2005). Contemporary research provides substantial evidence that integrating these decisions can improve efficiency and effectiveness when resources are limited and costs are nonlinear (i.e., exhibit economies of scale). For example, Kaminsky and Simchi-Levi (2003) describe a multiechelon serial supply chain in the pharmaceutical industry where products are manufactured in a series of production facilities with values adding on, and intermediate goods need to be transported between these facilities. Additionally, a multiechelon serial supply chain will occur, for instance, when products are distributed over great distances. In such a case, companies use central warehouses close to the production facilities and a number of local stocking points close to the customers in different areas. Hence, the produced items can be stored at the production facilities or transported to a central warehouse in the first level. Before reaching the final retail level, products are either stored at an intermediate distribution level, such as distribution center and wholesaler, or transported to the next level. In a long serial supply chain, it is advantageous to transport larger quantities of the products over long distances to storage facilities in intermediate echelons before distributing the products to the retailer level. Such a serial supply chain model can be represented by a generic multiechelon lot-sizing problem consisting of a manufacturer, several indistinguishable intermediate production or transportation levels, and a retailer level. The objective is to minimize the system-wide cost, including production, transportation, and inventory holding costs, while satisfying all demands.

As one of the most widely studied problems, the multiechelon lot-sizing problem has been considered primarily under the assumption that demands occur only at the final echelon. In this paper, we study a multiechelon lot-sizing problem in series with possible demands at intermediate echelons. Although integrating production, transportation, and inventory decisions becomes more challenging, it is of particular importance in supply chain systems with multichannel or intermediate products demand. For example, a retailer such as Microsoft or Apple that manufactures products overseas may ship products from its distribution centers directly to the customers who order through an official website, as well as to a brick-and-mortar store (e.g., Best Buy) to meet demand. In a value-added production system, a product needs to be transported through a sequence of production facilities. The intermediate goods created in this production system usually have their own demands, and it is important to fulfill the demand for both the intermediate and final products.

In practice, regardless of production or transportation echelons, capacities need to be imposed at each echelon in a serial supply chain. Following most of the literature on tractable cases, we concentrate on a multiechelon lot-sizing problem with a stationary capacity at the production echelon only and intermediate demand, which is referred as a *multiechelon capacitated lot-sizing problem* (MCLS), and its uncapacitated variation, denoted as *multiechelon uncapacitated lot-sizing problem* (MULS). As in standard lot-sizing problems, all cost functions are assumed to be nondecreasing in the amount produced, stored, or shipped. In particular, we consider concave cost functions and special cases such as linear production and transportation costs with a fixed charge. Before our literature review, we present the mathematical formulation of MCLS and MULS. We provide all technical proofs in an e-companion and use the prefix "EC" when referring to sections in the e-companion.

1.1. Mathematical Model and Notations

Let T be the length of the planning horizon and L be the number of echelons in a serial supply chain, where manufacturing occurs at the first echelon and products are transported from one echelon to the next echelon to satisfy demands. For each echelon $i \in [1, L]$ in each period $t \in [1, T]$, we define the following notation:

- d_t^i: demand faced by the customer at echelon i in period t.
- $\mathbb{C}$: constant production capacity at the production echelon (i.e., the first echelon).
- x_t^i: production or transportation quantity in period t at echelon i. If $i = 1$, it is the production quantity; otherwise, it is the transportation quantity from echelon $i-1$ to i.
- $p_t^i(x_t^i)$: production or transportation cost function at echelon i in period t for nonnegative amount x_t^i. Arguably, the concave cost structure is quite realistic (e.g., economies of scale) and therefore demands attention. As one of the most important cost structures, the fixed-charge case has received substantial attention in the literature. In such a case, we have $p_t^i(x_t^i) = f_t^i \delta(x_t^i) + c_t^i x_t^i$, where f_t^i is the fixed cost, c_t^i is production or transportation cost, and $\delta(x)$ is an indicator function taking the value 1 if $x > 0$ and 0 otherwise.
- I_t^i: inventory quantity held at echelon i at the end of period t.
- $h_t^i(I_t^i)$: concave inventory holding cost function at echelon i for nonnegative amount I_t^i at the end of period t. A widely used holding cost function is a linear function that we simply denote $h_t^i(I_t^i) = h_t^i \cdot I_t^i$.

As an example, Figure 1 shows the network flows of an eight-period, three-echelon lot-sizing problem. Throughout this paper, for an echelon i, we refer to echelon j as a *lower* echelon (or level) if $j > i$ and a *higher* echelon (or level) if $j < i$ according to their positions in the network (see Figure 1). We let $[i, j]$ denote the interval $\{i, i+1, \ldots, j\}$ for $i \le j$ and $[i, j] = \emptyset$ for $i > j$. Given the notations just defined, in MCLS, we minimize the total cost of the production and transportation plan through a serial structure supply chain, as follows:

$$\min \sum_{i=1}^{L} \sum_{t=1}^{T} \left(p_t^i(x_t^i) + h_t^i(I_i^i)\right) \tag{1a}$$

$$\text{s.t.} \quad I_{t-1}^i + x_t^i = I_t^i + x_t^{i+1} + d_t^i \quad \forall i \in [1, L-1], t \in [1, T], \tag{1b}$$

$$I_{t-1}^L + x_t^L = I_t^L + d_t^L \quad \forall t \in [1, T], \tag{1c}$$

$$x_t^1 \le \mathbb{C} \quad \forall t \in [1, T], \tag{1d}$$

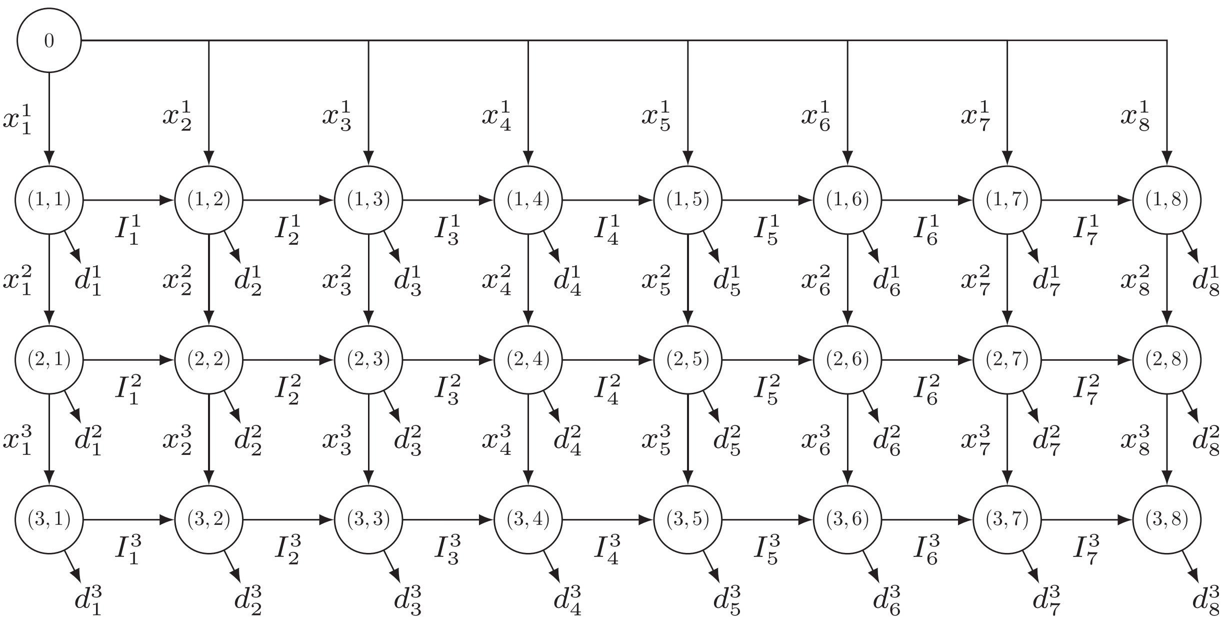
$$I_0^i = I_T^i = 0 \quad \forall i \in [1, L], \tag{1e}$$

$$x_t^i \ge 0, I_t^i \ge 0 \quad \forall i \in [1, L], t \in [1, T], \tag{1f}$$

where objective function (1a) is to minimize the total cost, including production cost, transportation cost, and inventory holding cost; constraints (1b) and (1c) balance inventory over time and echelons; inequalities (1d) constrain production capacity; constraints (1e) restrict the initial and ending inventories to be zero; and inequalities (1f) are nonnegativity constraints. Throughout this paper, we refer to a cost structure, with fixed-charge production and transportation costs and linear holding costs, as the *fixed-charge cost structure* for short.

In this paper, we are interested in exact methods for MCLS, where the capacity appears only at the production level, and its variation, MULS with

Figure 1. Eight-Period, Three-Echelon Lot-Sizing Network



intermediate demands, where the inequalities (1d) are dropped. For both MCLS and MULS, we focus mainly on general concave cost structure and fixed-charge cost structure. The problems we study generalize lot-sizing problems with a fixed number of echelons and demands occurring only at the final echelon. To simplify the comparison with the literature, we introduce some additional notation:

• When the number of echelons L is specified and fixed, we prefix "L-" to the abbreviations. For example, L-ULS and L-CLS are uncapacitated and capacitated L-echelon lot-sizing problems, respectively.

• When demand occurs only at the final echelon, we append "-F" to the abbreviations. For example, we have abbreviations MCLS-F, L-ULS-F, and L-CLS-F.

For notational convenience, we define $d^i(s,t) = \sum_{j=s}^{t} d_j^i$ (i.e., the cumulative demand at echelon i in periods from s to t), and it is 0 if $s > t$. Let $\mathscr{L}_0$ be a subset of the set $[1,L]$ such that every echelon $i \in \mathscr{L}_0$ is just a transshipment echelon without intermediate demands; that is, $d^i(1,T) = \sum_{j=1}^{T} d_j^i = 0$ $\forall i \in \mathscr{L}_0$ and $\mathscr{L}_1 = [1,L] \setminus \mathscr{L}_0$. We define $L_1 = |\mathscr{L}_1|$ and $L_0 = |\mathscr{L}_0|$ as the number of echelons with and without demand, respectively. Without loss of generality, we can assume that the final echelon $L \in \mathscr{L}_1$; otherwise, the problem can be reduced to an $L-1$ echelon lot-sizing problem. We also assume that $\sum_{\ell=1}^{L} d^\ell(1,t) \le t\mathbb{C}$, $t \in [1,T]$, to ensure the feasibility of MCLS.

1.2. Literature Review

The study of lot-sizing problems starts from the seminal paper by Wagner and Whitin (1958), in which the authors propose an $O(T^2)$ algorithm to solve the uncapacitated single-echelon lot-sizing problem (1-ULS) based on the properties of the extreme solutions. Decades later, the algorithm for the 1-ULS was improved by Federgruen and Tzur (1991), Wagelmans et al. (1992), and Aggarwal and Park (1993) to $O(T\log T)$. Krarup and Bilde (1977) develop an uncapacitated facility location formulation for 1-ULS, which is a compact extended formulation so that the linear relaxation yields an optimal solution satisfying all integrality restrictions. Barany et al. (1984) develop the so-called (ℓ,S) inequalities, which are sufficient to describe the convex hull of 1-ULS. Additionally, valid inequalities and the computational complexity of 1-ULS with inventory bounds (Atamtürk and Küçükyavuz 2005, 2008) and backlogging (Küçükyavuz and Pochet 2009) are studied in the literature.

The single-echelon lot-sizing problem with varying capacity is known to be NP-hard; see Florian et al. (1980) and Bitran and Yanasse (1982). However, by considering a constant capacity over the planning horizon (i.e., 1-CLS), Florian and Klein (1971) develop an $O(T^4)$ dynamic programming algorithm to solve the problem, which is improved to $O(T^3)$ by van Hoesel and Wagelmans (1996) when the holding costs are linear. An interesting extension of constant capacity is considering constant batch size. When backlogging is not allowed, Pochet and Wolsey (1993) give an $O(T^3)$ algorithm and propose the most important family of inequalities for 1-CLS, the so-called (k,l,S,I) inequality. With backlogging, Van Vyve (2007) also gives an $O(T^3)$ algorithm for a general number of maximal batches. The polytope of the varying capacity case of lot-sizing problems is studied by Atamtürk and Muñoz (2004). However, their results do not generalize the (k,l,S,I) inequality.

Another variant of capacitated lot sizing is a multi-item problem in which multiple items share the same limited resources, such as machine setup time. Fragkos et al. (2016) apply a branch-and-price algorithm to a horizon decomposition method, which partitions the single-echelon multiperiod problem into subproblems on consecutive overlapping intervals with smaller size. This horizon decomposition method is shown to be efficient and can be generalized to the problems with a generic constraint structure. In Akartunalı et al. (2016), the convex hull of the two-period subproblem is further studied, and a column-generation method is developed to generate violated valid inequalities for the two-period subproblems to effectively solve the original multiperiod problem. Comparing these period-wise decomposition methods for the single-echelon problem with such a big bucket capacity constraint, our study focuses on the multiechelon case to address a global supply chain with many levels of suppliers. We refer readers to Pochet and Wolsey (2006) for a detailed study of many variants of the lot-sizing problems.

The uncapacitated multiechelon lot-sizing problem is first studied by Zangwill (1969), with demand occurring at the final echelon only. The MULS-F is modeled as a network flow problem in a two-dimensional grid, and an $O(LT^4)$ dynamic programming algorithm is proposed by Zangwill (1969). Later, Love (1972) gives an $O(LT^3)$ algorithm by exploiting a nested structure based on the assumption that the production costs are nonincreasing over time and the holding costs are nondecreasing over echelons. Because of its importance in applications, 2-ULS-F receives a lot of attention. van Hoesel et al. (2005) show that 2-ULS-F can be solved in $O(T^3)$ time, which is improved to $O(T^2 \log T)$ by Melo and Wolsey (2010). Lee et al. (2003) study an application of 2-ULS-F with backlogging allowed in the last echelon and outbound shipment. This results in a stepwise and nonconcave transportation cost for the shipment between the two echelons. Lee et al. (2003) show that the problem can be solved in $O(T^6)$ time. Hwang (2010) generalizes the model by considering general concave production costs. Van Vyve et al. (2014) develop exact and approximate extended formulations for several variants of the two-echelon discrete lot-sizing problem with one item at the first echelon and multiple items at the second echelon, such as the uncapacitated cases, constant capacitated cases, and the cases with startup costs. To the best of our knowledge, Zhang et al. (2012) are the first to study the complexity of the multiechelon lot-sizing problem with intermediate demands, more specifically 2-ULS. Zhang et al. (2012) show that 2-ULS can be solved in $O(T^4)$ time with a dynamic programming algorithm, which implies an extended formulation with $O(T^4)$ variables and $O(T^4)$ constraints.

Kaminsky and Simchi-Levi (2003) study a three-echelon lot-sizing model with capacities, which can be reduced to a two-echelon model. With linear holding costs and no speculative motives assumption, the model can be solved in $O(T^8)$ time, where no speculative motives implies that it is optimal to transport to the lower echelons as late as possible (i.e., $c_t^\ell + h_t^{\ell+1} \geq c_{t+1}^\ell + h_t^\ell$ in the case of the fixed-charge cost structure). van Hoesel et al. (2005) provide a detailed analysis of the capacitated multiechelon lot-sizing problem MCLS-F. They show that in the case of fixed-charge transportation costs without speculative motives, MCLS-F can be solved in polynomial time, $O(T^7 + LT^4)$, and the algorithm complexity can be improved to $O(T^6)$ when $L = 2$. However, they provide only a pseudo-polynomial algorithm with complexity $O(LT^{2L+3})$ for MCLS-F with general concave costs. Later, this result is significantly improved by Hwang et al. (2013) because the MCLS-F is proved to be polynomial solvable in $O(LT^8)$ time using a new concept called *basis paths*.

It is well known that the lot-sizing problem with concave costs can be seen as a minimum concave cost network flow problem in a two-dimensional grid with only one source. Because much of the literature on network flow problems focuses on more general settings where the network could have multiple sources, we find only two papers, by He et al. (2015) and Ahmed et al. (2016), closely related to our work. For more research on the general network flow problem, we refer to the literature in those two papers. He et al. (2015) focus on the grid network with multiple sources at the first level. They show that MULS, as a special case, is polynomial solvable when the number of echelons L is fixed, but the computational complexity is not specified. Ahmed et al. (2016) study a similar grid network with flow capacities and classify many NP-hard and polynomial solvable cases. They show that the minimum concave cost network flow problem in a two-dimensional grid with two sources at the top level is NP-hard. However, the complexity of solving the single-source problem, MULS, is unknown.

Despite the extensive study of lot-sizing problems, the polyhedral study of multiechelon lot-sizing problems has received little attention in the literature, except for the valid inequalities for the two-echelon production planning problem with a complex assembly structure developed by Gaglioppa et al. (2008) and a large family of two-echelon inequalities for 2-ULS proposed by Zhang et al. (2012).

1.3. Main Contributions and Outline

This paper contributes both to theory and computation. In Section 2, we prove that the MULS with a fixed-charge cost structure is NP-hard and provide a list of the computational complexities of our proposed algorithms for solving MULS and

Table 1. Computational Complexity Results

Model	Cost structure	Complexity
MULS	General concave costs	$O(LT^{3L_1+1})$
	Fixed-harge cost structure	$O(L\min(T^L \log T, T^{3L_1+1}))$
MCLS	General concave costs	$O(LT^{2L_1L+2})$
	Fixed-charge transportation costs and no speculative motives	$O(T^{4L_1+2} + LT^{3L_1+1})$

MCLS (see Table 1). Note that our algorithms outperform the best-known algorithms proposed in the literature (see Table 2). In Section 3, we study the valid inequalities for MULS and MCLS, and our results generalize many known inequalities, such as (ℓ, S), two-echelon, and (k, l, S, I) inequalities. The efficacy of applying our inequalities in solving large instances of the multi-item, multiechelon lot-sizing problem is demonstrated in Section 4. We demonstrate that the branch-and-cut algorithm with the proposed inequalities is four times faster than CPLEX and approximately two times faster than the best-known algorithms in the literature. For ease of reading, we discuss our major theoretical findings and computational study in the text but include the complete proofs of all theorems, propositions, and corollaries and detailed computational results in the e-companion. Last, we conclude our paper in Section 5.

2. Computational Complexities

In this section, we prove that MULS is NP-hard and develop efficient polynomial algorithms for MULS and MCLS given a fixed number of echelons. By considering intermediate demands, our results (see Table 1) generalize and outperform much existing research (see Table 2), including that of Zangwill (1969), Lee et al. (2003), van Hoesel et al. (2005), and Zhang et al. (2012), which are special cases of our model.

2.1. NP-Hardness

Much effort has been spent in finding efficient polynomial-time algorithms or proving NP-hardness for the dynamic lot-sizing problem and its extensions to certain network flow problems. On the one hand, if demand occurs at the final echelon only, then MULS-F and MCLS-F are polynomial solvable, as shown by Zangwill (1969) and Hwang et al. (2013), respectively. On the other hand, Ahmed et al. (2016) show that the minimum concave cost flow in a two-dimensional grid is NP-hard if there are two sources at the first level. An obvious gap between these results is the unknown complexity of the MULS. The main result of this subsection, Theorem 1, aims to close this gap.

Theorem 1. *The MULS with a fixed-charge cost structure is NP-hard.*

The proof of Theorem 1 is a reduction from planar 3SAT and can be found in Section EC.1 of e-companion.

2.2. Uncapacitated Cases

First, we generalize the approach of Zangwill (1969) on MULS-F to MULS by considering intermediate demands. We then propose a novel algorithm to solve MULS with a fixed-charge cost structure, which outperforms the one suggested by Zhang et al. (2012). We show that our approach provides a better extended formulation and can be extended to solve 2-ULS with backlogging and outbound transportation, where the transportation cost functions are stepwise and nonconcave.

Unlike lot-sizing problems with demands at the final echelon only, our problem requires characterizing demand satisfaction status at each echelon, which motivates us to introduce L-dimensional vectors. Hence, we define a set $\mathcal{V} = \{v \in [0, T]^L : v_1 \le \cdots \le v_L\}$. For notational convenience, we denote $\mathbf{0} \in \mathcal{V}$ with $\mathbf{0}_i = 0\ \forall i \in \mathcal{L}$, $\mathbf{1} \in \mathcal{V}$ with $\mathbf{1}_i = 1\ \forall i \in [1, L]$, and $\mathbf{T} \in \mathcal{V}$ with $\mathbf{T}_i = T\ \forall i \in [1, L]$.

Table 2. Computational Complexity Comparisons Between Our Results and the Best-Known Results

Model		The best-known results	Results of this paper
3-ULS-F[a]	$O(T^4)$	By Zangwill (1969)	$O(T^3 \log T)$
MCLS-F[b]	$O(LT^{2L+3})$	By van Hoesel et al. (2005)	$O(LT^{2L+2})$
	$O(LT^8)$	By Hwang et al. (2013)	
MCLS-F[c]	$O(T^7 + LT^4)$	By van Hoesel et al. (2005)	$O(T^6 + LT^4)$
2-CLS-F[c]	$O(T^6)$	by van Hoesel et al. (2005)	$O(T^5)$
2-ULS[a]	$O(T^4)$	By Zhang et al. (2012)	$O(T^2 \log T)$

[a]With fixed-charge cost structure.
[b]With general concave cost functions.
[c]With fixed-charge transportation costs and no speculative motives.

2.2.1. General Concave Costs. Following the traditional view of Zangwill (1969), the MULS with concave cost functions can be seen as a minimum concave flow problem in a two-dimensional grid. The shipment pattern of any extreme solution can be characterized similarly, except that satisfying intermediate demands requires more detailed analysis (see Section EC.2.1 of the e-companion).

Theorem 2. *The MULS with concave cost functions can be solved in $O(LT^{3L_1+1})$ time.*

In the case of MULS-F, we have $L_1 = 1$, so the computational complexity is $O(LT^4)$, the result derived by Zangwill (1969). Theorem 2 indicates that the complexity of MULS depends only on the number of echelons with demands, regardless of which echelons have demands.

2.2.2. Fixed-Charge Cost Structure. As one of the most important cost structures, the fixed-charge cost structure receives a lot of attention in the literature on dynamic lot-sizing problems. It is well known that linear holding costs can be dropped and replaced by variable production and transportation costs because $I_t^i = \sum_{\tau=1}^{t}(x_\tau^i - x_\tau^{i+1}) - d^i(1,t)\ \forall i \in [1, L-1]$ and $I_t^L = \sum_{\tau=1}^{t} x_\tau^L - d^L(1,t)$. Hence, in this subsection, we assume that the holding costs are 0 without loss of generality.

In the case of the single-echelon lot-sizing problem with a fixed-charge cost structure, the idea of regeneration points and intervals is used to reformulate the problem as a shortest-path problem on a directed graph. We will generalize regeneration points and intervals to the multiechelon lot-sizing problem as regeneration vectors and arcs, which can be used to reformulate the problem as finding a shortest path on a directed graph. This novel approach is very different from the traditional method that considers MULS as a minimum cost network flow on a two-dimensional grid. We show that the resulting dynamic programming recursions outperform many of the best-known algorithms.

Definition 1. A vector $v \in [0,T]^L$ is a *regeneration* vector if $v \in \mathcal{V}$ and $I_{v_i}^i = 0$ for all $i \in [1,L]$. Let $\mathcal{A}_i = \{(v,w) : v, w \in \mathcal{V}, v_j = w_j\ \forall j \neq i \text{ and } v_i < w_i\}$ for all $i \in [1,L]$. A pair of two regeneration vectors (v,w) forms a regeneration arc at echelon i with $i \in [1,L]$ if $(v,w) \in \mathcal{A}_i$.

Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where the arc set $\mathcal{A} = \bigcup_{i=1}^{L} \mathcal{A}_i$. We define the cost of arc $(v,w) \in \mathcal{A}_i$ for any $i \in [1,L]$ as

$$C(v,w) = d^i(v_i+1, w_i)\sum_{\ell=1}^{i} c^\ell_{v_\ell+1} + \begin{cases} f^i_{v_i+1} & \text{if } d^i(v_i+1,w_i) + \sum_{\ell=i+1}^{L} d^\ell(1,v_\ell) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The arc (v,w) assumes that the demands in periods $v_i+1, \ldots, w_i$ at echelon i are all satisfied by the production in period v_1+1 and transportation in period $v_\ell+1$ at each echelon $\ell\ \forall \ell \in [1,i]$. Thus, the arc cost includes the production and transportation costs at all echelons $\ell \in [1,i]$ in period $v_\ell+1$ for demand $d^i(v_i+1,w_i)$ and the fixed cost at echelon i in period v_i+1 if necessary. The costs can be decomposed into each potential regeneration arc because the variable costs are additive. One of our main theorems shows the following.

Theorem 3. *The MULS with a fixed-charge cost structure can be solved as a shortest-path problem on a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with the source node* **0** *and the sink node* **T**.

The following example shows that an extreme solution is related to a path composed of regeneration vectors and arcs.

Example 1. Given a two-echelon, six-period, uncapacitated lot-sizing problem, we can view it as a network flow problem in a two-dimensional grid, as shown in Figure 2. According to Theorem 3, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with node set $\mathcal{V} = \{v \in [0,6] \times [0,6] : v_1 \leq v_2\}$ and arc set $\mathcal{A} = \{(v,w) : v,w \in \mathcal{V}, v_1 = w_1, v_2 < w_2\} \cup \{(v,w) : v,w \in \mathcal{V}, v_1 < w_1, v_2 = w_2\}$, as shown in Figure 3. The solution shown in Figure 2 is an extreme solution whose total cost is

$$(d^1(1,3)+d^2(1,5))c_1^1 + f_1^1 + d^2(1,2)c_1^2 + f_1^2 + d^2(3,5)c_3^2 + f_3^2 + d_4^1 c_4^1 + f_4^1 + (d^1(5,6)+d_6^2)c_5^1 + f_5^1 + d_6^2 c_6^2 + f_6^2.$$

It corresponds to the path from (0, 0) to (6, 6) as depicted in Figure 3, with the same total cost according to the arc costs marked by each arc. To simplify the notation, we assume that all demands are positive at each echelon in each period, so the fixed cost has to be considered for each arc in the path.

Remark 1. In Example 1,

- the demand $d^2(1,2)$ in Figure 2 is satisfied by production and transportation at period 1, which correspond to the arc ((0, 0), (0, 2)) in Figure 3 with cost $d^2(1,2)(c_1^1 + c_1^2) + f_1^2$;

Figure 2. An Extreme Solution of a Two-Echelon, Six-Period, Uncapacitated Lot-Sizing Problem

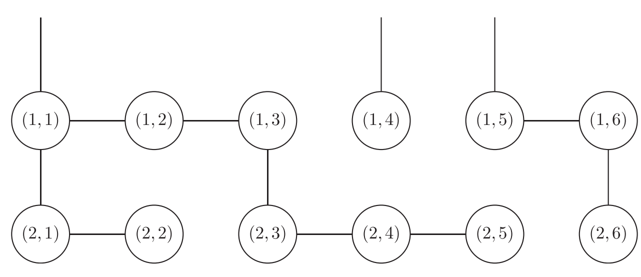
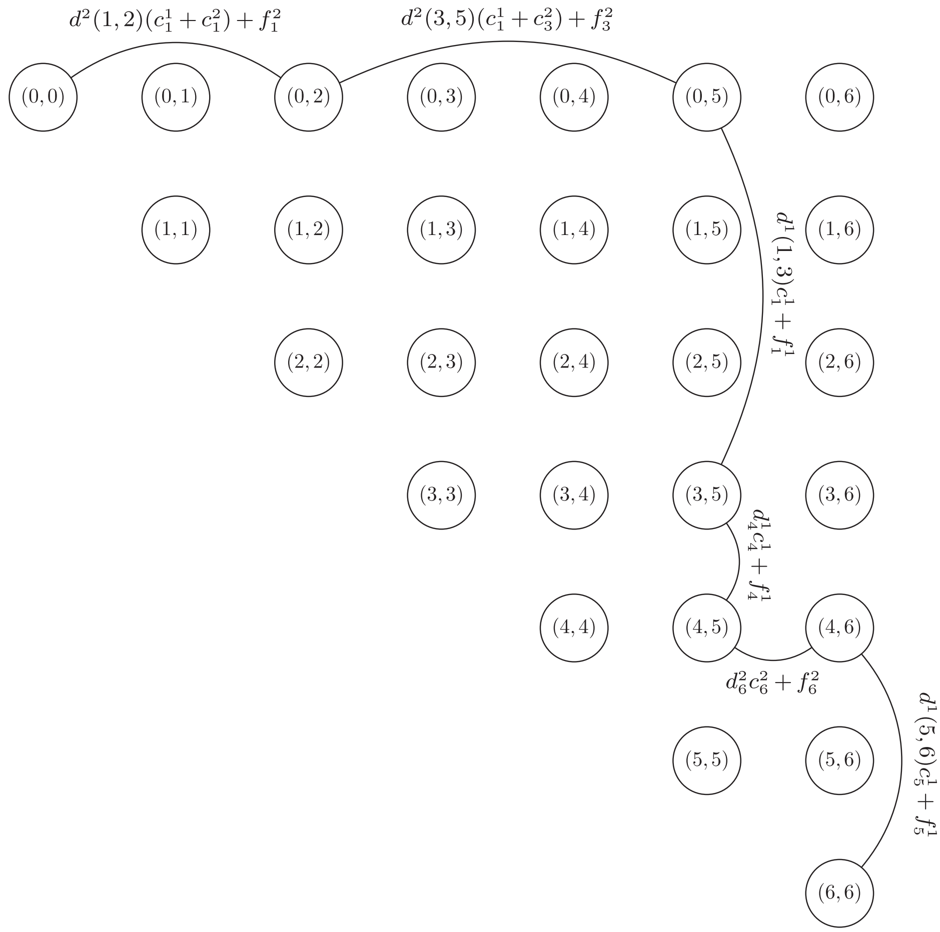


Figure 3. A Path in the Graph $\mathcal{G}$ Corresponds to the Extreme Solution in Figure 2



• the demand $d^2(3,5)$ is satisfied by production at period 1 and transportation at period 3, which correspond to the arc $((0,2),(0,5))$ in Figure 3 with cost $d^2(3,5)(c_1^1 + c_3^2) + f_3^2$; and

• the demand $d^1(1,3)$ is satisfied by production at period 1, which corresponds to the arc $((0,5),(3,5))$ in Figure 3 with cost $d^1(1,3)c_1^1 + f_1^1$, and a similar relationship follows between Figures 2 and 3.

It is worth noting that the demand d_6^2 is satisfied by production at period 5 and transportation at period 6, which correspond to the arc $((4,5),(4,6))$. The reason that an arc corresponding to d_6^2 does not follow immediately from an arc corresponding to $d^2(3,5)$ is that arc $((4,5),(4,6))$ provides the production and transportation information of d_6^2. Thus, the associated cost of satisfying d_6^2 can be calculated correctly. Note that a path consisting of arcs $((0,0),(0,2)),((0,2),(0,5)),((0,5),(2,5))$ corresponds to a solution that is not extreme. The path indicates that the total production in period 1 is $d^1(1,2) + d^2(1,5)$. Because d_3^1 is not satisfied by the production at period 1, we need a positive production in period 3 to fulfill its demand (i.e., $x_3^1 > 0$). To satisfy the demands $d^2(3,5)$ in periods 3–5, we must hold positive ending inventory in period 2 (i.e., $I_2^1 > 0$). Because $I_2^1 x_3^1 > 0$, the zero inventory ordering (ZIO) property (see Section EC.2.2 in the e-companion) is violated, and the solution is not extreme. Although some paths do not suggest extreme solutions, the shortest path will correspond to a solution with minimum cost because the proof of Theorem 3 shows that the total cost of a path is equal to the total cost of the corresponding solution.

Remark 2. The cost of arc (v,w) in (2) contains a fixed cost $f_{v_i+1}^i$ if the value of $d^i(v_i+1,w_i) + \sum_{\ell=i+1}^L d^\ell(1,v_\ell)$ is nonzero. The term $\sum_{\ell=i+1}^L d^\ell(1,v_\ell)$ is necessary in case echelon i has no demand. For example, Figure 4 shows an extreme solution of a two-echelon, two-period, uncapacitated lot-sizing problem where $d_1^1 = d_2^1 = 0$ and $d_1^2 = d_2^2 = 1$.

The corresponding path in graph $\mathcal{G}$ is $((0,0),(0,1))$, $((0,1),(0,2)),((0,2),(2,2))$. To calculate $C_{((0,2),(2,2))}$ as in (2), where $i = 1$, $v = (0,2)$, and $w = (2,2)$, we have $d^i(v_i+1,w_i) = d^1(1,2) = 0$ and $\sum_{\ell=i+1}^L d^\ell(1,v_\ell) = d^2(1,2) = 2$. Then we get $C_{((0,2),(2,2))} = f_1^1$, where the fixed cost f_1^1 is necessary in the cost calculation because of the production at period 1. However, if the term $\sum_{\ell=i+1}^L$

$d^\ell(1,v_\ell)$ is neglected, we will get $C_{((0,2),(2,2))}=0$, which results in an incorrect cost calculation.

For a given vector $v \in \mathcal{V}$, let $G(v)$ represent the minimum cost of a path in $\mathcal{G}$ from v to $\mathbf{T}$. For notational convenience, we denote $v_{L+1}=T$ and $(v_{-i},\alpha)\in\mathcal{V}$ as a vector equal to v, except that the ith component is α. The shortest-path algorithm implies that

$$\begin{aligned} G(v) &= \min_{w:(v,w)\in\mathcal{A}} C(v,w)+G(w) \\ &= \min_{i\in[1,L]:v_i<v_{i+1}} \min_{w:(v,w)\in\mathcal{A}_i} C(v,w)+G(w) \qquad (3) \\ &= \min_{i\in[1,L]:v_i<v_{i+1}} \min_{\substack{w:w_{-i}=v_{-i} \text{ and} \\ v_i<w_i\le v_{i+1}}} C(v,w)+G(v_{-i},w_i). \qquad (4) \end{aligned}$$

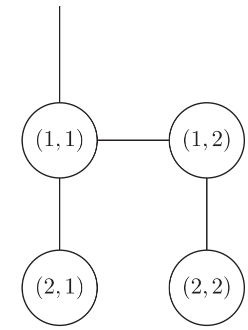
Because $(v,w)\in\mathcal{A}$, we know that $(v,w)\in\mathcal{A}_i$ for $i\in[1,L]$. From the definitions of $\mathcal{A}_i$ and $\mathcal{V}$, we have $v_{i+1}=w_{i+1}\ge w_i>v_i$. Thus, Equations (3) and (4) hold. The arc cost $C(v,w)$ equals a possible fixed cost plus $d^i(v_i+1,\mu)\sum_{\ell=1}^{i} c^\ell_{v_\ell+1}$, where the fixed cost $f^i_{v_i+1}$ occurs if we have production or transportation at echelon i in period v_i+1. Apparently, the fixed cost has to be considered if $d^i_{v_i+1}>0$. Otherwise, in the case of $d^i_{v_i+1}=0$, we can compare the costs by having $G(v)=G(v_{-i},v_i+1)$ (i.e., ignoring the 0 demand) or having production/transportation in period v_i+1 at echelon i. Note that $(v_{-i},v_i+1)\in\mathcal{V}$ because $v_{i+1}>v_i$ in (4). In contrast to the arc cost, here the way of dealing with fixed cost follows Wagelmans et al. (1992). Therefore, $G(v)$ satisfies a dynamic programming recursion as follows, and consequently, we have Corollary 1:

$$G(v)=\min_{i\in[1,L]:v_i<v_{i+1}} \begin{cases} \min\limits_{v_i<\lambda\le v_{i+1}} \left[f^i_{v_i+1}+d^i(v_i+1,\lambda) \sum\limits_{\ell=1}^{i} c^\ell_{v_\ell+1}+G(v_{-i},\lambda)\right] & \text{if } d^i_{v_i+1}>0, \\ \min\left\{G(v_{-i},v_i+1), \min\limits_{v_i<\lambda\le v_{i+1}} \left[f^i_{v_i+1}+ d^i(v_i+1,\lambda)\sum\limits_{\ell=1}^{i} c^\ell_{v_\ell+1}+G(v_{-i},\lambda)\right]\right\} & \text{if } d^i_{v_i+1}=0. \end{cases} \qquad (5)$$

Corollary 1. *The MULS with a fixed-charge cost structure can be solved in* $O(\min(LT^{3L_1+1},\ LT^L\log T))$ *time.*

Remark 3. The traditional method (see Zangwill 1969, Melo and Wolsey 2010, Zhang et al. 2012) applies dynamic programming recursions on a two-dimensional grid with nodes in the form of (echelon, period). The method does not fully explore the characterization of extreme solutions and has difficulty in generalizing to higher-dimensional problems. Our algorithm is to find the shortest path on an L-dimensional graph with node set $\mathcal{V}$. As evidence of its efficacy, the complexity result in Theorem 1 improves many previous studies (see Table 2).

Figure 4. An Extreme Solution of a Two-Echelon, Two-Period, Uncapacitated Lot-Sizing Problem



Because the MULS with a fixed-charge cost structure can be solved by finding a shortest path in the graph $\mathcal{G}$ with $O(T^L)$ nodes and $O(LT^{L+1})$ arcs, we can have an extended formulation (see EC.2.4 in the e-companion) with $O(LT^{L+1})$ variables and $O(T^L)$ constraints. For a review of extended formulations, we refer readers to Pochet and Wolsey (2006). Zhang et al. (2012) establish a hierarchy of formulations by studying their relative strength when $L=2$. Van Vyve et al. (2014) propose extended formulations of two-echelon lot-sizing problems with multiple items at the final echelon and startup costs. Our extended formulation is very different and obtained from the shortest-path algorithm for the purpose of efficiently addressing the existence of intermediate demands. In particular, our extended formulation has fewer variables and constraints than the one in Zhang et al. (2012). However, when demand occurs only at the final echelon, the proposed extended formulation is probably not the best one compared with existing extended formulations.

2.3. Capacitated Cases

Capacitated lot-sizing problems are much more complicated than their uncapacitated counterparts mainly because the zero inventory ordering property does not hold after imposing a capacity. However, starting with Florian and Klein (1971), many researchers find that the production and transportation quantities can be enumerated in polynomial time if the capacity is stationary. van Hoesel et al. (2005) provide a detailed study on capacitated multiechelon lot-sizing problems with demand occurring at the final echelon only. They show that in the view of the network flow problem, the flow corresponding to any extreme solution can be decomposed into a sequence of so-called subplans, each of which has at most one positive production arc where the production quantity is strictly less than the capacity. This property of the subplan allows them to enumerate all possible values of cumulative production and transportation quantities for each subplan. They are then able to solve MCLS-F through a two-phase dynamic programming by considering adjacent subplans.

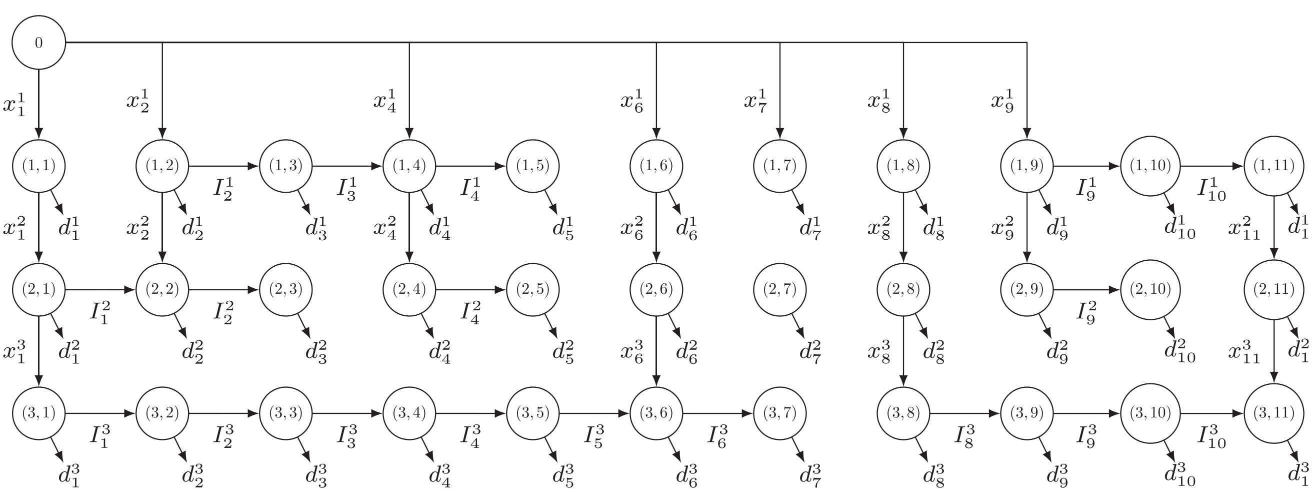
A more general approach is proposed in this subsection to solve MCLS. In particular, the definition of a subplan needs to be adjusted for MCLS because of the intermediate demands. It turns out that, in our case, the cumulative production and transportation quantities have a much richer structure. By enumerating all allowable values, we are able to solve MCLS in $O(LT^{2L_1L+2})$ time for general concave costs, which outperforms the approach proposed by van Hoesel et al. (2005) for solving MCLS-F. Throughout this subsection, for two L-dimensional vectors u and v, we denote $u \le v$ if $u_\ell \le v_\ell$ $\forall \ell \in [1, L]$ component-wise. Similarly, $\min(u, v)$ is an L-dimensional vector whose ℓth component equals $\min(u_\ell, v_\ell)$ $\forall \ell \in [1, L]$. We also define a relaxed set of $\mathcal{V}$ by fixing the components corresponding to the transshipment echelons to 0 (i.e., $\overline{\mathcal{V}} = \{v \in [0, T]^L : v_i \le v_j \forall i \le j \in \mathcal{L}_1$ and $v_i = 0\, \forall i \in \mathcal{L}_0\}$). For notational convenience, we denote $\bar{\mathbf{1}} \in \overline{\mathcal{V}}$ with $\bar{\mathbf{1}}_i = 1$ $\forall i \in \mathcal{L}_1$ and $\overline{\mathbf{T}} \in \overline{\mathcal{V}}$ with $\overline{\mathbf{T}}_i = T$ $\forall i \in \mathcal{L}_1$. We define $\mathcal{D}_i(\bar{v}, \bar{w}) = \sum_{\ell=i}^{L} d^\ell(\bar{v}_\ell + 1, \bar{w}_\ell)$ $\forall i \in [1, L]$ and vector $\vec{\mathcal{D}}(\bar{v}, \bar{w}) = (\mathcal{D}_1(\bar{v}, \bar{w}), \dots, \mathcal{D}_L(\bar{v}, \bar{w}))$.

2.3.1. Subplans and Relaxed Subplans. We consider the flows in the network corresponding to the extreme points, where the network consists of nodes (i, t) at echelon i in period t (see Figure 5). After removing all production arcs, the network is decomposed into several connected components. We then connect each isolated node without demand to the node on its left (connecting to its above node if the node is in period 1) and define each connected component as a *subplan* (v, w) with $v, w \in \mathcal{V}$ and $v \le w$. We also require that, for any $j \in [1, L-1]$, we have either $v_{j+1} < w_j$ or $v_\ell = w_\ell$ $\forall \ell \in [j+1, L]$ owing to the arborescent structure of an extreme solution in the subplan. Note that if $v_{j+1} \ge w_j$, then, in the subplan, there is no flow that can connect to the nodes $(j+1, v_{j+1}+1), \dots, (j+1, w_{j+1})$. Hence, the subplan only contains nodes at echelons $1, \dots, j$, and we need to have $v_\ell = w_\ell$ $\forall \ell \in [j+1, L]$. We say that two subplans (v', w') and (v'', w'') are consecutive if $w' = v''$. For example, in Figure 5, if $d_7^1 > 0$, then we have three consecutive subplans $((0,0,0), (6,7,7))$, $((6,7,7), (7,7,7))$, and $((7,7,7), (11,11,11))$. Otherwise, we have only two consecutive subplans $((0,0,0), (7,7,7))$ and $((7,7,7), (11,11,11))$. As common in network flow problems, we define an arc as *free* if it carries an amount of flow that is positive and strictly less than its capacity. Because the flow corresponding to the extreme solution is acyclic, at most, one free production arc enters the subplan.

The definition of a subplan was first introduced by van Hoesel et al. (2005), except that their requirement is $v_{i+1} < w_i$ $\forall i \in [1, L-1]$ for a subplan (v, w). For example, in Figure 5, $((6,7,7), (7,7,7))$ is no longer a valid subplan because $v_2 = w_1 = 7$. Thus, we have only two consecutive subplans $((0,0,0), (7,7,7))$ and $((7,7,7), (11,11,11))$ when $d_7^1 > 0$. Apparently, we could have two free production arcs, for example, x_4^1 and x_7^1, in the subplan $((0,0,0), (7,7,7))$. The example shows that with the requirements $v_{i+1} < w_i$ $\forall i \in [1, L-1]$ in the case of having intermediate demands, we cannot ensure that at most one free production arc enters each subplan, which is the key property we need for subplans. Therefore, a different (relaxed) subplan definition is necessary.

In summary, any extreme solution can be decomposed into a sequence of consecutive subplans, and at most one free production arc enters the subplan. Even though we need two vectors $v, w \in \mathcal{V}$ for a subplan, the total demand of a subplan depends only on $\bar{v}, \bar{w} \in \overline{\mathcal{V}}$, where $\bar{v}_\ell = v_\ell$ and $\bar{w}_\ell = w_\ell$ $\forall \ell \in \mathcal{L}_1$. Hence, we define the pair $(\bar{v}, \bar{w})$ as a relaxed subplan. We know

Figure 5. The Flow Corresponding to an Extreme Solution to the 3-CLS with 11 Periods



that the subplan (v, w) satisfies, for any $j \in [1, L-1]$, either $v_{j+1} < w_j$ or $v_\ell = w_\ell \ \forall \ell \in [j+1, L]$. This implies that $(\bar{v}, \bar{w})$ satisfies, for any $j \in [1, L-1] \cap \mathcal{L}_1$, either $v_{j+1} < w_j$ or $v_\ell = w_\ell \ \forall \ell \in [j+1, L]$. Similarly, two relaxed subplans $(\bar{v}', \bar{w}')$ and $(\bar{v}'', \bar{w}'')$ are consecutive if $\bar{w}' = \bar{v}''$. On the basis of the preceding discussions, we have the following.

Proposition 1. *The total demand $\mathcal{D}_1(\mathbf{0}, \overline{\mathbf{T}})$ can be decomposed into the demands of a sequence of consecutive relaxed subplans. The demand of a relaxed subplan is served by at most one positive production quantity whose value is strictly less than the capacity.*

Although the concepts of subplans and relaxed subplans are related, they differ fundamentally. A subplan is a subnetwork connected by a flow that indicates a production and transportation plan of a subproblem, whereas a relaxed subplan is just a pair of vectors that indicate the demand quantities. When the demand occurs at the final echelon only, a relaxed subplan is identified by a pair $(\bar{v}, \bar{w})$ whose only nonfixed components are at the final echelon. Then it is equivalent to represent the relaxed subplan as $(\bar{v}_L, \bar{w}_L)$.

2.3.2. General Concave Costs. Given a relaxed subplan $(\bar{v}, \bar{w})$, we define a graph $\mathcal{G}_{(\bar{v},\bar{w})}$ with nodes (t, X, τ) where

- $t \in [\bar{v}_1, \bar{w}_L]$ is a time period;
- $X = (X_1, \ldots, X_L)$ is a cumulative quantity vector such that X_1 is a cumulative production quantity up to and including period t, and X_i, $\forall i \in [2, L]$, is a cumulative transportation quantity from echelon $i-1$ to echelon i up to and including period t;
- $\tau \in [\bar{v}_1, \bar{w}_L]$ if $X_1 = \mathcal{D}_1(\bar{v}, \bar{w})$, and $\tau = 0$ otherwise.

Let $\overline{X}$ be a cumulative quantity vector up to and including period $t-1$ in the relaxed subplan $(\bar{v}, \bar{w})$. For all $t, \overline{X}, X$ and τ, we define arcs in $\mathcal{G}_{(\bar{v},\bar{w})}$ as follows:

- $((t-1, \overline{X}, 0), (t, X, 0))$ if $\overline{X}_1 \le X_1 < \mathcal{D}_1(\bar{v}, \bar{w})$;
- $((t-1, \overline{X}, 0), (t, X, t))$ if $\overline{X}_1 < \mathcal{D}_1(\bar{v}, \bar{w}) = X_1$;
- $((t-1, \overline{X}, \tau), (t, X, \tau))$ if $\overline{X}_1 = X_1 = \mathcal{D}_1(\bar{v}, \bar{w})$.

Note that τ will be used as an indicator to identify the last production period. All arcs with given $t, \overline{X}, X$ indicate the same planning that at echelon $i \in [1, L]$ in period t, the production (or transportation if $i > 1$) quantity is $X_i - \overline{X}_i$, and the inventory quantity is $X_i - X_{i+1} - d^i(\bar{v}_i + 1, \min(t, \bar{w}_i))$, except that, at echelon L, the inventory quantity is $X_L - d^L(\bar{v}_L + 1, t)$. Hence, arcs with given $t, \overline{X}, X$ have the same arc cost as follows:

$$\sum_{i=1}^{L} p_t^i(X_i - \overline{X}_i) + \sum_{i=1}^{L-1} h_t^i(X_i - X_{i+1} - d^i(\bar{v}_i + 1, \min(t, \bar{w}_i))) + h_t^L(X_L - d^L(\bar{v}_L + 1, t)). \quad (6)$$

It is important to note that the choices of X are not arbitrary because production quantities are bounded by the capacity and restricted by Proposition 1. In particular, Remark EC.3 in the e-companion shows that the number of allowable values for (X, τ) is $O(T^{2L_1(L-1)+1})$. Because $\overline{X}$ and X are two cumulative quantity vectors in two adjacent periods, the choices of $(\overline{X}, X, \tau)$ can be further limited, and Proposition EC.3 shows that the number of allowable values for $(\overline{X}, X, \tau)$ is $O(T^{2L_1(L-1)+1})$. Because arcs in $\mathcal{G}_{(\bar{v},\bar{w})}$ include all allowable values for $(\overline{X}, X)$ in two adjacent periods, a path from node $(\bar{v}_1, \mathbf{0}, 0)$ to node $(\bar{w}_L, \vec{\mathcal{D}}(\bar{v}, \bar{w}), \tau)$, for some $\tau > 0$, corresponds to a solution that satisfies demand $\mathcal{D}_1(\bar{v}, \bar{w})$ in the relaxed subplan $(\bar{v}, \bar{w})$, and τ is the last production period based on our arc definition.

Let $\mathcal{G}^c = \bigcup_{(\bar{v},\bar{w})} \mathcal{G}_{(\bar{v},\bar{w})}$ be a graph that is the union of $\mathcal{G}_{(\bar{v},\bar{w})}$ for all relaxed subplans. We denote each node (t, X, τ) in $\mathcal{G}_{(\bar{v},\bar{w})}$ as a node $(t, X, \tau)_{(\bar{v},\bar{w})}$ in $\mathcal{G}^c$. Additionally, we add a source node $(0, \mathbf{0}, 0)_{(\mathbf{0},\mathbf{0})}$ and a sink node $(T, \mathbf{0}, 0)_{(\overline{\mathbf{T}},\overline{\mathbf{T}})}$ to the graph $\mathcal{G}^c$.

First, we connect the source node to nodes $(0, \mathbf{0}, 0)_{(\mathbf{0},\bar{v})}$ for all $\bar{v} \in \overline{\mathcal{V}}$ with 0 arc cost. So a path from the source node to node $(\bar{v}_L, \vec{\mathcal{D}}(\mathbf{0}, \bar{v}), \tau)_{(\mathbf{0},\bar{v})}$, for some $\bar{v}$ and $\tau > 0$, corresponds to a solution that satisfies demand $\mathcal{D}_1(\mathbf{0}, \bar{v})$ of the relaxed subplan $(\mathbf{0}, \bar{v})$.

Next, for any two consecutive relaxed subplans $(\bar{u}, \bar{v})$ and $(\bar{v}, \bar{w})$, we connect each node $(\bar{v}_L, \vec{\mathcal{D}}(\bar{u}, \bar{v}), \tau)_{(\bar{u},\bar{v})}$ with $\tau > 0$ in $\mathcal{G}_{(\bar{u},\bar{v})}$ to a node $(\hat{\tau}, \mathbf{0}, 0)_{(\bar{v},\bar{w})}$ with $\hat{\tau} = \max(\tau, \bar{v}_1)$ in $\mathcal{G}_{(\bar{v},\bar{w})}$ through a dummy arc with cost 0. Recall that τ in a node $(\bar{v}_L, \vec{\mathcal{D}}(\bar{u}, \bar{v}), \tau)_{(\bar{u},\bar{v})}$ indicates the last production period for satisfying demand in $\mathcal{D}_1(\bar{u}, \bar{v})$ in the relaxed subplan $(\bar{u}, \bar{v})$. Each dummy arc ensures that the production used to satisfy demand in $\mathcal{D}_1(\bar{v}, \bar{w})$ occurs after period $\hat{\tau} = \max(\tau, \bar{v}_1)$. After subgraphs in $\mathcal{G}^c$ are connected by dummy arcs, a path from the source node to node $(T, \vec{\mathcal{D}}(\bar{w}, \overline{\mathbf{T}}), \tau)_{(\bar{w},\overline{\mathbf{T}})}$, for some $\bar{w}$ and $\tau > 0$, corresponds to a solution that satisfies the total demand $\mathcal{D}_1(\mathbf{0}, \overline{\mathbf{T}})$.

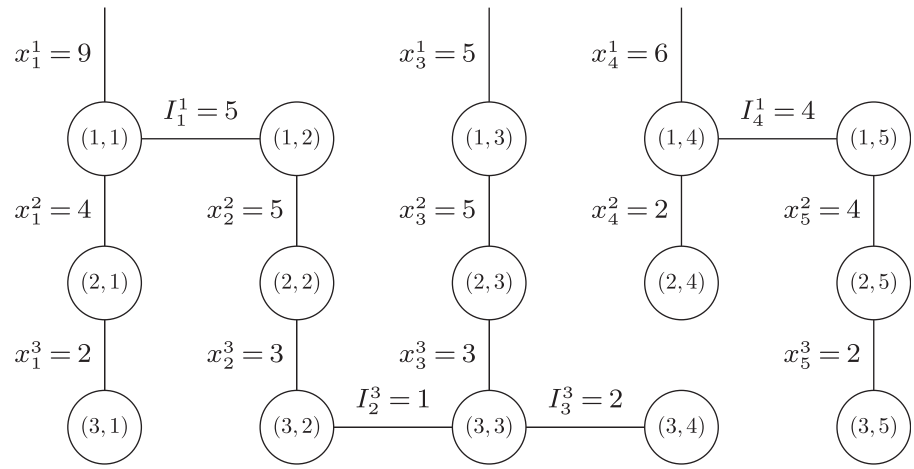
Finally, we connect nodes $(T, \vec{\mathcal{D}}(\bar{w}, \overline{\mathbf{T}}), \tau)_{(\bar{w},\overline{\mathbf{T}})}$ for all $\bar{w} \in \overline{\mathcal{V}}$ and $\tau > 0$ to the sink node with 0 arc cost. Therefore, the MCLS can be solved as a shortest-path problem from the source node to the sink node on a directed acyclic graph $\mathcal{G}^c$. Because there are $O(T^{3L_1+1})$ combinations of $(\bar{u}, \bar{v}, \bar{w}, \tau)$, the number of dummy arcs in $\mathcal{G}^c$ is $O(T^{3L_1+1})$. Also, there are $O(T^{2L_1+1+2L_1(L-1)+1}) = O(T^{2L_1L+2})$ combinations of $(\bar{v}, \bar{w}, t, \overline{X}, X, \tau)$. Because $T^{3L_1+1} \le T^{2L_1L+2}$, the total number of arcs (as well as nodes) in $\mathcal{G}^c$ is $O(T^{2L_1L+2})$. Note that the cost (6) for each arc can be calculated in $O(L)$ time. So we have the following theorem.

Theorem 4. *The MCLS can be solved as a shortest-path problem on a directed acyclic graph $\mathcal{G}^c$ in $O(LT^{2L_1L+2})$ time.*

Example 2 illustrates the relationship between an extreme solution of an MCLS and a path in graph $\mathcal{G}^c$.

Example 2. Given an extreme solution of a three-echelon, five-period, capacitated lot-sizing problem as shown in Figure 6, with $\mathbb{C} = 9$, $d_t^1 = 0$, and $d_t^\ell = 2$ for

Figure 6. An Extreme Solution of a Three-Echelon, Five-Period, Capacitated Lot-Sizing Problem



$t = 1, \dots, 5$ and $\ell = 2, 3$, we have source node $(0, (0, 0, 0), 0)_{((0,0,0),(0,0,0))}$ and sink node $(5, (0, 0, 0), 0)_{((0,5,5),(0,5,5))}$ in $\mathcal{G}^c$. Table 3 contains nine arcs in graph $\mathcal{G}^c$ corresponding to the extreme solution in Figure 6. Because the total demand can be decomposed into demands of two relaxed subplans $((0, 0, 0), (0, 3, 4))$ and $((0, 3, 4), (0, 5, 5))$, arcs 2–5 and arcs 7 and 8 are in two subgraphs $\mathcal{G}_{((0,0,0),(0,3,4))}$ and $\mathcal{G}_{((0,3,4),(0,5,5))}$, respectively. Arc 6 serves as a dummy arc to link two subgraphs. Arc 1 links the source node to $\mathcal{G}_{((0,0,0),(0,3,4))}$, and arc 9 links the sink node to $\mathcal{G}_{((0,3,4),(0,5,5))}$.

For example, the node $(t, X, \tau)_{(\bar{v},\bar{w})} = (4, (14, 14, 8), 3)_{((0,0,0),(0,3,4))}$ indicates that at period 4 in relaxed subplan $((0, 0, 0), (0, 3, 4))$, we have the cumulative production quantity up to and including period 4 as $X_1 = 14 = x_1^1 + x_3^1$ (note that x_4^1 is not included because we consider relaxed subplan $((0, 0, 0), (0, 3, 4))$), the cumulative transportation quantity from echelon 1 to echelon 2 up to and including period 4 as $X_2 = 14 = x_1^2 + x_2^2 + x_3^2$, and the cumulative transportation quantity from echelon 2 to 3 up to and including period 4 as $X_3 = 8 = x_1^3 + x_2^3 + x_3^3$. Because $X_1 = 14$ indicates that all demands in the relaxed subplan $((0, 0, 0), (0, 3, 4))$ are produced, the last production period is $\tau = 3$.

Remark 4. For 1-CLS, we have $L = L_1 = 1$. Theorem 4 implies the same complexity $O(T^4)$ as in Florian and Klein (1971). In the case of 2-CLS-F (i.e., $L = 2$ and $L_1 = 1$), Theorem 4 indicates that the model can be solved in $O(T^6)$ time, which improves the complexity $O(T^7)$ proposed by van Hoesel et al. (2005).

2.3.3. Fixed-Charge Transportation Costs Without Speculative Motives.

The assumption of no speculative motives is commonly made for the production and inventory holding costs in traditional economic lot-sizing models and appears often in the literature. In the context of fixed-charge transportation costs, no speculative motives indicates that it is attractive to transport as late as possible. More formally, $c_t^\ell + h_t^{\ell+1} \geq c_{t+1}^\ell + h_t^\ell$ if inventory holding cost is linear. Following theorem 4.4 in van Hoesel et al. (2005), there exists an optimal solution satisfying the ZIO property (i.e., $I_{t-1}^i x_t^i = 0$ for $i \in [1, L]$ and $t \in [1, T]$) because of fixed-charge transportation costs and no

Table 3. A Path in Graph $\mathcal{G}^c$

Arc	From node	To node	
i	$(t-1, \overline{X}, \tau')_{(\bar{u},\bar{v})}$	$(t, X, \tau)_{(\bar{v},\bar{w})}$	Arc cost
1	$(0, (0, 0, 0), 0)_{((0,0,0),(0,0,0))}$	$(0, (0, 0, 0), 0)_{((0,0,0),(0,3,4))}$	0
2	$(0, (0, 0, 0), 0)_{((0,0,0),(0,3,4))}$	$(1, (9, 4, 2), 0)_{((0,0,0),(0,3,4))}$	$p_1^1(9) + h_1^1(5) + p_1^2(4) + p_1^3(2)$
3	$(1, (9, 4, 2), 0)_{((0,0,0),(0,3,4))}$	$(2, (9, 9, 5), 0)_{((0,0,0),(0,3,4))}$	$p_2^2(5) + p_2^3(3) + h_2^3(1)$
4	$(2, (9, 9, 5), 0)_{((0,0,0),(0,3,4))}$	$(3, (14, 14, 8), 3)_{((0,0,0),(0,3,4))}$	$p_3^1(5) + p_3^2(5) + p_3^3(3) + h_3^3(2)$
5	$(3, (14, 14, 8), 3)_{((0,0,0),(0,3,4))}$	$(4, (14, 14, 8), 3)_{((0,0,0),(0,3,4))}$	0
6	$(4, (14, 14, 8), 3)_{((0,0,0),(0,3,4))}$	$(3, (0, 0, 0), 0)_{((0,3,4),(0,5,5))}$	0
7	$(3, (0, 0, 0), 0)_{((0,3,4),(0,5,5))}$	$(4, (6, 2, 0), 4)_{((0,3,4),(0,5,5))}$	$p_4^1(6) + p_4^2(2) + h_4^1(4)$
8	$(4, (6, 2, 0), 4)_{((0,3,4),(0,5,5))}$	$(5, (6, 6, 2), 4)_{((0,3,4),(0,5,5))}$	$p_5^2(4) + p_5^3(2)$
9	$(5, (6, 6, 2), 4)_{((0,3,4),(0,5,5))}$	$(5, (0, 0, 0), 0)_{((0,5,5),(0,5,5))}$	0

Note. Except for special arcs 1, 6, and 9 that are defined in some specific ways, our arc definition shows that (a) "from node" and "to node" are in the form of $(t-1, \overline{X}, \tau')_{(\bar{u},\bar{v})}$ and $(t, X, \tau)_{(\bar{v},\bar{w})}$, respectively, and (b) if $\overline{X}_1 < X_1$ = total demand in a subplan, then $\tau' = 0$ and $\tau = t$; otherwise, $\tau' = \tau$.

speculative motives. Therefore, we can limit ourselves to the extreme solutions with the ZIO property. MCLS can be solved by decoupling the production echelon (the first echelon) from the rest of the model, and the echelons from level 2 to level L can be solved by using dynamic programming recursions in Section 2.2.1 because of the ZIO property. Note that the production costs and inventory holding costs are still assumed to be generally concave. Thus we have the following theorem.

Theorem 5. *The MCLS with fixed-charge transportation costs and no speculative motives can be solved in* $O(T^{4L_1+2} + LT^{3L_1+1})$ *time.*

Corollary 2. *The 2-CLS-F with fixed-charge transportation costs and no speculative motives can be solved in* $O(T^5)$ *time.*

Remark 5. Theorem 5 differs from Theorem 4 in that it shows a complexity that is independent of the number of echelons L. Under the condition of fixed-charge transportation costs with no speculative motives, MCLS can be solved within polynomial time when L_1 is fixed. Our results generalize and outperform algorithms in van Hoesel et al. (2005), where the authors have presented an $O(T^7 + LT^4)$ algorithm for solving MCLS-F and an $O(T^6)$ algorithm for solving 2-CLS-F.

3. Multiechelon Inequalities

In this section, we focus on developing valid inequalities for the multiechelon lot-sizing problem with a fixed-charge cost structure as follows:

$$\min \quad \sum_{i=1}^{L}\sum_{t=1}^{T}(c_t^i x_t^i + f_t^i y_t^i) \tag{7a}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} x_t^i = \sum_{\ell=i}^{L} d^\ell(1,T) \qquad \forall i \in [1,L], \tag{7b}$$

$$\sum_{j=1}^{t}(x_j^i - x_j^{i+1}) \ge d^i(1,t) \qquad \forall i \in [1,L-1],\ t \in [1,T], \tag{7c}$$

$$\sum_{j=1}^{t} x_j^L \ge d^L(1,t) \qquad \forall t \in [1,T], \tag{7d}$$

$$x_t^i \le \sum_{\ell=i}^{L} d^\ell(t,T) y_t^i \qquad \forall t \in [1,T], \tag{7e}$$

$$x_t^1 \le \mathbb{C} y_t^1 \qquad \forall t \in [1,T], \tag{7f}$$

$$x_t^i \ge 0, y_t^i \in \{0,1\} \qquad \forall i \in [1,L], t \in [1,T], \tag{7g}$$

where binary variables $y_t^i\ \forall i \in [1,L], t \in [1,T]$ are introduced to model the fixed-charge costs. The efficacy of the inequalities proposed in this section will be demonstrated in Section 4 by solving large instances of the multi-item, multiechelon lot-sizing problem.

3.1. Uncapacitated Case

We present a family of valid inequalities for MULS that generalizes many known inequalities.

Theorem 6. *For* $0 = k_0 \le k_1 \le \cdots \le k_L \le n$, *let* $[k_{i-1}+1, k_i] \subseteq T_i \subseteq [1,k_i]$ *and* $S_i \subseteq T_i\ \forall i \in [1,L]$. *We have an L-echelon inequality*

$$\sum_{i=1}^{L}\left(\sum_{t\in T_i\setminus S_i} x_t^i + \sum_{t\in S_i}\phi_t^i y_t^i\right) \ge \sum_{i=1}^{L} d^i(1,k_i), \tag{8}$$

where

$$\phi_t^i = \sum_{\ell=i}^{L} d^\ell(\alpha_{\ell t}^i + 1, \beta_{\ell t}^i) \tag{9}$$

such that $\alpha_{it}^i = t-1$, $\beta_{it}^i = \max\{\tau \ge t-1 : [t,\tau] \subseteq T_i\}$,

$$\alpha_{\ell t}^i = \max\{\tau \ge \alpha_{\ell-1,t}^i : [\alpha_{\ell-1,t}^i + 1, \tau] \subseteq T_\ell\} \text{ and}$$
$$\beta_{\ell t}^i = \max\{\tau \ge \beta_{\ell-1,t}^i : [\beta_{\ell-1,t}^i + 1, \tau] \subseteq T_\ell\} \tag{10}$$

for $\ell \in [i+1, L]$.

Remark 6. In the definition of $\alpha_{\ell t}^i$ and $\beta_{\ell t}^i$, we have $\alpha_{\ell t}^i = \alpha_{\ell-1,t}^i$ when $\alpha_{\ell-1,t}^i + 1 \notin T_\ell$ and $\beta_{\ell t}^i = \beta_{\ell-1,t}^i$ when $\beta_{\ell-1,t}^i + 1 \notin T_\ell$. In general, we have $\beta_{\ell t}^i \le k_\ell\ \forall \ell \in [i,L]$. However, $\beta_{it}^i = \max\{\tau \ge t-1 : [t,\tau] \subseteq T_i\} = k_i$ when $T_i = [1,k_i]$ for some $i \in [1,L]$. In such a case, $\beta_{\ell t}^i = k_\ell\ \forall \ell \in [i+1,L]$ follows because of the iterative definition and the fact that $[k_{\ell-1}+1, k_\ell] \subseteq T_\ell\ \forall \ell \in [i+1,L]$.

If $0 = k_0 = k_1 = \cdots = k_{L-1} \le k_L \le n$, then $T_i = S_i = \emptyset$ $\forall i \in [1,L-1]$ and $S_L \subseteq T_L = [1,k_L]$. It is easy to calculate that $\alpha_{L,t}^L = t-1$ and $\beta_{L,t}^L = k_L$. Inequality (8) then reduces to

$$\sum_{t\in[1,k_L]\setminus S_L} x_t^L + \sum_{t\in S_L} d^L(t,k_L) y_t^L \ge d^L(1,k_L),$$

which is the (ℓ, S) inequality of Barany et al. (1984).

If $0 = k_0 = k_1 = \cdots = k_{L-2} \le k_{L-1} \le k_L \le n$, then $T_i = S_i = \emptyset\ \forall i \in [1,L-2]$, $S_{L-1} \subseteq T_{L-1} = [1,k_{L-1}]$, $S_L \subseteq T_L$, and $[k_{L-1}, k_L] \subseteq T_L \subseteq [1,k_L]$. It follows that $\alpha_{L-1,j}^{L-1} = \alpha_{L,t}^L = t-1$, $\alpha_{L,t}^{L-1} = \beta_{L,t}^L = \max\{\tau \ge t-1 : [t,\tau] \subseteq T_L\}$, $\beta_{L-1,t}^{L-1} = k_{L-1}$, and $\beta_{L,t}^{L-1} = k_L$. Inequality (8) becomes

$$\sum_{i=L-1}^{L}\left(\sum_{t\in T_i\setminus S_i} x_t^i + \sum_{t\in S_i}\phi_t^i y_t^i\right) \ge \sum_{i=L-1}^{L} d^i(1,k_i),$$

where $\phi_t^{L-1} = d^{L-1}(t,k_{L-1}) + d^L(t,k_L) - d^L(t,\beta_{L,t}^L)$ and $\phi_t^L = d^L(t,\beta_{L,t}^L)$, which is the two-echelon inequality of Zhang et al. (2012).

The necessary and sufficient facet conditions for an L-echelon inequality are expected to be complicated (we refer readers to Zhang et al. 2012 on the necessary and sufficient conditions for the two-echelon inequality), and it could be even more difficult to implement the conditions into a branch-and-cut algorithm.

Therefore, we consider only two simple necessary conditions that make an L-echelon inequality facet defining, and they are implemented in our computational study.

Proposition 2. *If the L-echelon inequality* (8) *is facet defining, then we have, for any* $j \in [1, L-1]$,*(a)* $[1, t_j] \cap T_{j+1} = \emptyset$, *where* $t_j = \max\{\tau : [1, \tau] \subseteq \cup_{l \in [1,j]}(T_l \setminus S_l)\}$, *and (b) when* $[\underline{t}_j, \bar{t}_j] \subseteq T_j \setminus S_j$, *there exists* $t \in [\underline{t}_j - 1, \bar{t}_j]$ *such that* $[\underline{t}_j, \bar{t}_j] \cap T_{j+1} = [\underline{t}_j, t]$.

Remark 7. Multiechelon inequalities are special cases of the dicut collection inequalities introduced by Rardin and Wolsey (1993). However, dicut collection inequalities are written implicitly as a function of a collection of dicuts in a graph without known combinatorial separation algorithms. To yield a multiechelon inequality, the required dicut collection $\Gamma = \cup_{i \in [1,L]} \{\Gamma_t^i\}_{t \in [1,T]}$ has each Γ_t^i as a singleton $\{Q_t^i\}$ for $t \in [1, n]$ and $i \in [1, L]$. The dicut collection that gives the multiechelon inequality is, for $i \in [1, L]$, $\Gamma_t^i = \emptyset$ if $t \in [k_i + 1, T]$ and $\Gamma_t^i = \{Q_t^i\} = \cup_{l \in [1,i]} \{x_j^l : j \in [1, t] \cap T_l\} \cup \cup_{l \in [1,i]} \{y_j^l : j \in \{\tau : t \in [\alpha_{l\tau}^i + 1, \beta_{l\tau}^i]\} \cap S_l\}$ if $t \in [1, k_i]$. Because a dicut with a smaller size will result in stronger inequalities, Proposition 2 improves the dicut collection. We refer readers to Rardin and Wolsey (1993) for further details on the dicut collection inequalities.

3.2. Separation for L-Echelon Inequality

The exact separation algorithm can be generalized from proposition 4 for 2-ULS in Zhang et al. (2012) with a similar proof. We use the shortest-path network introduced in Section 2.2.2 for the separation network. We have the following proposition.

Proposition 3. *Given a fractional solution of a multiechelon lot-sizing problem, there is an* $O(LT^{2L})$ *algorithm to find the most violated inequality (8), if any.*

As Zhang et al. (2012) point out, even deriving the exact separation for a two-echelon inequality is quite time consuming in practice owing to its $O(T^4)$ time complexity. Now we give an efficient separation heuristic and show the computational results in Section 4.

Note that we can always aggregate adjacent echelons to construct an MULS with fewer than L echelons. The main part of our separation algorithm is to obtain $\mathbf{m} = \{m_1, \ldots, m_{|\mathbf{m}|}\}$ and $\mathbf{k} = \{k_1, \ldots, k_{|\mathbf{m}|}\}$, where a new MULS is derived by aggregating echelons m_1 to $m_2 - 1$ as the first echelon and so on until aggregating echelons $m_{|\mathbf{m}|}$ to L as the last echelon. Vector $\mathbf{k}$ is composed of all the k_i values for each echelon after aggregation. Algorithm 1 provides the details of finding $\mathbf{m}$ and $\mathbf{k}$.

Algorithm 1 Find $\mathbf{m}$ and $\mathbf{k}$

1. $K_i \leftarrow \emptyset \ \forall i \in [1, L]$
2. Aggregate echelons $i, \ldots, L$ to a 1-ULS and apply (ℓ, S) inequalities
3. **if** (ℓ, S) inequalities cut off current linear programming (LP) solution **then**
4. $\quad K_i \leftarrow K_i \cup \{\ell\}$
5. Initiate two vectors $\mathbf{m}$ and $\mathbf{k}$
6. **for** $m_1 = 1$ to $L - 1$ and $m_2 = m_1 + 1$ to L **do**
7. $\quad$ **for** $k_1 \in K_1$ and $k_2 \in K_2$ such that $k_1 \le k_2$ **do**
8. $\quad\quad \mathbf{m}.push\_back(m_1)$, $\mathbf{m}.push\_back(m_2)$
9. $\quad\quad \mathbf{k}.push\_back(k_1)$, $\mathbf{k}.push\_back(k_2)$
10. $\quad\quad$ **for** $i = m_2 + 1$ to L **do**
11. $\quad\quad\quad$ **if** $k_i = \min\{k \in K_i : k \ge$ the last element in $\mathbf{k}\}$ exists **then**
12. $\quad\quad\quad\quad \mathbf{m}.push\_back(i)$; $\mathbf{k}.push\_back(k_i)$

We then obtain $T_i \ \forall i \in \mathbf{m}$ using the separation network described in Algorithm 2 with the given $\mathbf{k}$. Without loss of generality, we present Algorithm 2 by assuming $|\mathbf{m}| = L$ and $\mathbf{m}_i = i$ in order to simplify notation. Algorithm 2 shows that $T_i \ \forall i \in [1, L]$ can be determined by a shortest-path algorithm with complexity $O(LT^2)$. Proposition 2 is applied next to improve the choice of $T_i \ \forall i \in [1, L]$. We let $t \in S_i$ if $x_t^i > \phi_t^i y_t^i$. Finally, the L-echelon inequality (8) is added only if it cuts off the current LP solution.

3.3. Capacitated Case

The most general class of inequalities defined for constant-capacity, single-echelon lot sizing (1-CLS) is the so-called (k, l, S, I) inequalities of Pochet and Wolsey (1993, 2006). The main idea is to construct a mixing set in the form $\{(s, z_j) \in \mathbb{R}_+ \times \mathbb{Z}^n : s + \mathbb{C} z_j \ge b_j\}$, and then the mixing mixed-integer rounding (MIR) inequalities developed by Günlük and Pochet (2001) imply valid inequalities for 1-CLS. Let $\{i_0, \ldots, i_{s-1}\} = S \subseteq [k, l]$ with $k < l \in [1, T]$ and $|S| = s$. We denote $i_j = \max\{\tau \in [k-1, l] : |[k, \tau] \setminus S| = j\}$ for $j = 0, \ldots, s$. It is clear that $i_s = l$. As demonstrated by Pochet and Wolsey (2006), the mixing set with inequalities

$$\sum_{t=1}^{k-1} x_t^1 + \sum_{t \in [k,l] \setminus S} x_t^1 + \mathbb{C} \sum_{t \in [k, i_j] \cap S} y_t^1 \ge d^1(1, i_j) \quad \forall j \in [0, s] \tag{11}$$

can be used to generate the (k, l, S, I) inequality. In this subsection, we only consider valid inequalities for 2-CLS. We give a family of valid inequalities (12) that can be used to construct a mixing set.

Algorithm 2 Find $T_i \ \forall i \in [1, L]$ with given $k_i \ \forall i \in [1, L]$

1. $T_1 \leftarrow [1, k_1]$, and $T_i \leftarrow \emptyset \ \forall i \in [2, L]$
2. $j \leftarrow T$, $T'_{j-1} \leftarrow [1, k_{j-1}]$
3. **repeat**

Procedure: build a separation network with node set $\{0, \ldots, k_{j-1} + 1\} \cup \{1', \ldots, k'_{j-1}\}$

4. $\quad$ Arcs $(0, 1)$ and $(0, 1')$ with cost 0
5. $\quad$ Arcs (u, v') with $1 \le u < v \le k_{j-1}$ (or $(u, k_{j-1} + 1)$).

The shortest path visiting the arc (u, v') implies that to minimize the left-hand side of inequality (8), we let $[u, v-1] \subseteq T_j$ and $u-1, v \notin T_j$. The cost on this arc is

$$\sum_{t=u}^{v-1} \min\left\{x_{\tau}^{j-1}, \phi_t^{j-1} y_t^{j-1}\right\} + \sum_{t=u}^{v-1} \min\left\{x_t^j, \phi_t^j y_t^j\right\},$$

where ϕ_t^{j-1} and ϕ_t^j are defined in (9). Note that $\alpha_{j-1,t}^{j-1} = t-1$, $\beta_{j-1,t}^{j-1} = k_{j-1}$, $\alpha_{j,t}^{j-1} = t-1$, $\beta_{j,t}^{j-1} = k_j$. $\alpha_{j,t}^{j} = t-1$, and $\beta_{j,t}^{j} = t-1$. The rest of αs and βs can be obtained iteratively because T_ℓ, $\forall \ell \in [j+1, L]$, are known.

6. Arc (v', w) with $1 \le v < w \le k_{j-1}+1$. A shortest path visiting each arc (v', w) implies that to minimize the left-hand side of inequality (8), we let $[v, w-1] \cap T_j = \emptyset$ and $v-1, w \in T_j$. The cost on this arc is

$$\sum_{t=v}^{w-1} \min\left\{x_t^{j-1}, \phi_t^{j-1} y_t^{j-1}\right\},$$

where ϕ_t^{j-1} is defined in (9). Note that $\alpha_{j-1,t}^{j-1} = t-1$, $\beta_{j-1,t}^{j-1} = k_{j-1}$, $\alpha_{j,t}^{j-1} = t-1$, and $\beta_{j,t}^{j-1} = k_j$. The rest of the αs and βs can be obtained iteratively because T_ℓ, $\forall \ell \in [j+1, L]$ are known

End Procedure

7. Find a shortest path from node 0 to node $k_{j-1}+1$ in the separation network
8. **for** all arcs (a, b) in the shortest path **do**
9. **if** $(a, b) = (u, v')$ is an arc defined in step 5 **then**
10: Let $[u, v-1] \subseteq T_j$ and $u-1, v \notin T_j$
11. **if** $(a, b) = (v', w)$ is an arc defined in step 6 **then**
12. Let $[v, w-1] \cap T_j = \emptyset$ and $v-1, w \in T_j$
13. $j \leftarrow j-1$
14. **until** $j = 1$
15. **return** T_i $\forall i \in [1, L]$

Proposition 4. *The inequalities*

$$\sum_{t=1}^{k-1}(x_t^1 - x_t^2) + \sum_{t\in[k,l]\setminus S} x_t^1 + \mathbb{C}\sum_{t\in[k,i_j]\cap S} y_t^1 \ge \max\big(d^1(1,i_j), d^1(1,i_j) + d^2(1,i_j) + d^1(1,k-1) - (k-1)\mathbb{C}\big) \quad \forall j \in [0,s] \tag{12}$$

are valid for MCLS.

Remark 8. We can use inequalities (12) to construct mixing sets, and the valid inequalities for 2-CLS can be derived by applying the mixing MIR procedure. In Example 3, two valid and facet-defining inequalities are obtained using this procedure. Clearly, building a mixing set to generate valid inequalities is crucial. We contributes a method to develop valid inequalities in an *explicit form*, based on the explicit convex hull description for mixing set developed by Günlük and Pochet (2001). More important, the inequalities we developed generalize the (k, l, S, I) inequalities to 2-CLS because inequalities (12) degenerate to (11) for 1-CLS, in which case, $d^1(1, k-1) \le (k-1)\mathbb{C}$.

Owing to the complexity of the capacitated lot-sizing problem, Pochet and Wolsey (1993) show that the (k, l, S, I) inequality is not enough to describe the convex hull of 1-CLS. No combinatorial polynomial exact separation algorithm is known for the (k, l, S, I) inequality (see Pochet and Wolsey 2006). Even more, we do not find any literature on studying (k, l, S, I) inequalities computationally. As evidence in our computational study with a simple separation heuristic, the results show that the (k, l, S, I) inequality is not necessarily more efficient than the (ℓ, S) inequality by Barany et al. (1984). Thus, an algorithm that can find good mixing sets is important on effectively applying mixing-set based inequalities, which we leave as future research.

Example 3. Consider a 2-CLS problem with four periods, $d_t^1 = d_t^2 = 10$ $\forall t \in [1, 4]$ and $\mathbb{C} = 32$. Let $k = 2$, $l = 4$, $S = \{3, 4\}$. Then $i_0 = 2$, $i_1 = 3$, and $i_2 = 4$. We have a mixing set described by inequalities (12) as follows:

$$(x_1^1 - x_1^2) + x_2^1 \ge 20$$
$$(x_1^1 - x_1^2) + x_2^1 + 32y_3^1 \ge 38$$
$$(x_1^1 - x_1^2) + x_2^1 + 32y_3^1 + 32y_4^1 \ge 58.$$

A valid inequality, which is also facet-defining,

$$x_1^1 - x_1^2 + x_2^1 + 18y_3^1 + 6y_4^1 \ge 44$$

can be derived from the mixing set.

A mixing set can be obtained by other valid inequalities. When $d^2(k, n) \le \mathbb{C}$, as in the example, we have valid inequalities (with a similar proof as for Proposition 4)

$$\sum_{t=1}^{k-1}(x_t^1 - x_t^2) + \sum_{t\in[k,l]\setminus S} x_t^1 + \mathbb{C}\left(\sum_{t\in[k,i_j]\cap S} y_t^1 + \sum_{t\in[k,i_p]\cap S} y_t^2\right) \ge \max\big(d^1(1,i_j), d^1(1,i_j) + d^2(1,i_{\max(p,j)}) + d^1(1,k-1) - (k-1)\mathbb{C}\big) \quad \forall j \in [0,s], p \in [0, j+1]. \tag{13}$$

We can get a different mixing set with inequalities

$$(x_1^1 - x_1^2) + x_2^1 \ge 20,$$
$$(x_1^1 - x_1^2) + x_2^1 + 32y_3^2 \ge 28,$$
$$(x_1^1 - x_1^2) + x_2^1 + 32y_3^1 + 32y_3^2 \ge 38,$$
$$(x_1^1 - x_1^2) + x_2^1 + 32y_3^1 + 32y_4^1 + 32y_3^2 \ge 58,$$

where the last three inequalities are in the form of (13) with $p = 1$ and $j = 0, 1, 2$. The mixing set gives another valid and facet-defining inequality

$$x_1^1 - x_1^2 + x_2^1 + 16y_3^1 + 6y_4^1 + 2y_3^2 \ge 44.$$

4. Computational Results

To investigate the computational advantages of the proposed strong inequalities, we implement them as cutting planes in the branch-and-bound process of the state-of-the-art optimization solver CPLEX; that is, as an implementation of the branch-and-cut (B&C) algorithm. In particular, the Concert Technology of CPLEX 12.6.1 is used for adding user-defined cuts, and a traditional branch-and-bound search method with single thread is adopted. All tests are subject to a 3,600-second central processing unit (CPU) time limit. If no optimal solution is obtained within the time limit, then the optimality gap is reported. The computational study was carried out on a node machine at the University of Houston Center for Advanced Computing and Data Science, which contains two Intel Xeon 2.8-GHz processors with 8-GB memories.

Our testing data sets consist of instances of the multi-item, multiechelon lot-sizing problem with T periods, L echelons, R items, and mode constraints that allow at most κ orders to be placed in each period at each echelon. On the basis of a hierarchy of formulations established in Zhang et al. (2012) and our initial tests on different formulations, we use the formulation in Equations (14a)–(14g) rather than a multi-item formulation based on Equations (1a)–(1f). Let M_a^i and $\hat{d}_{at}^i$ be the order capacity and demand for item a at echelon i in period t, respectively. The demands, fixed costs, variable costs, and holding cost for each item at each echelon in each period are generated using a discrete uniform distribution in the intervals $[0, 50]$, $[1{,}000, 2{,}000]$, $[0, 20]$, and $[0, 6]$, respectively, except that $\hat{d}_{at}^i = 0\ \forall t \in [1, \lceil R/\kappa \rceil]$ to ensure the feasibility because of the mode constraints. The capacity M_a^i is set to $3\lceil \sum_{\ell=i}^{L} \hat{d}_a^i(1, T)/T \rceil$. We define $\hat{d}_a^i(u, v) = \sum_{j=u}^{v} \hat{d}_{aj}^i$. Let x_{at}^i denote the total order quantity of item a in period t at echelon i. The mixed-integer programming formulation of the capacitated multi-item lot-sizing problem with mode constraints is as follows:

$$\min \quad \sum_{a=1}^{R} \sum_{i=1}^{L} \sum_{t=1}^{T} (c_t^i x_{at}^i + f_t^i y_{at}^i) \tag{14a}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} x_{at}^i = \sum_{\ell=i}^{L} \hat{d}_a^\ell(1, T) \quad \forall i \in [1, L], a \in [1, R], \tag{14b}$$

$$\sum_{j=1}^{t} (x_{aj}^i - x_{aj}^{i+1}) \geq \hat{d}_a^i(1, t) \quad \forall i \in [1, L-1], t \in [1, T], a \in [1, a], \tag{14c}$$

$$\sum_{j=1}^{t} x_{aj}^L \geq \hat{d}_a^L(1, t) \quad \forall t \in [1, T], a \in [1, R], \tag{14d}$$

$$x_{at}^i \leq M_a^i y_{at}^i \quad \forall i \in [1, L], t \in [1, T], a \in [1, R], \tag{14e}$$

Table 4. Sizes of All Instances

L	$T.R.\kappa$	L	$T.R.\kappa$	L	$T.R.\kappa$	L	$T.R.\kappa$
3	15.7.4	5	20.3.2	7	20.2.1	9	30.1.1
	20.7.4		25.3.2		25.2.1		35.1.1
	20.5.3		15.5.3		15.3.2		15.2.1
	25.5.3		20.5.3		20.3.2		20.2.1

$$\sum_{a=1}^{R} y_{at}^i \leq \kappa \quad \forall i \in [1, L], t \in [1, T], \tag{14f}$$

$$x_{at}^i \geq 0, y_{at}^i \in \{0, 1\} \quad \forall i \in [1, L], t \in [1, T], a \in [1, R]. \tag{14g}$$

By varying the values of T, L, R, κ with $\kappa = \lceil R/2 \rceil$, we generate 80 base instances of 16 different sizes (see Table 4) and five instances for each size.

Then, for each instance, we randomly select an echelon $l \in [2, L-1]$ and generate a new instance by letting $\hat{d}_{at}^l = 0\ \forall a \in [1, R], t \in [1, T]$. Hence, in all new instances, there is no demand for any product at one of their echelons and $L_1 = L - 1$. We call those 80 instances *derived instances*. Those random instances are tested with implementation of the following computing methods:

- **CPX** indicates the default CPLEX with traditional B&C in single-thread mode;
- **LS** indicates a B&C algorithm to which the (ℓ, S) inequalities (Barany et al. (1984)) are added at the root node only;
- **C** indicates a B&C algorithm to which the (k, l, S, I) inequalities (Pochet and Wolsey (1993)) are added at the root node only;
- **2LS** indicates a B&C algorithm to which the (ℓ, S) inequalities and the two-echelon inequalities (Zhang et al. (2012)) are added at the root node only; and
- **MLS** indicates a B&C algorithm to which the (ℓ, S) inequalities and the L-echelon inequalities are added at the root node only.

Observing that adding cuts at every node of the branch-and-bound tree is rather ineffective, our implementations mainly focus on cut generation at the root node. We implement **LS** and **2LS** algorithms as in Zhang et al. (2012). The **C** algorithm is implemented by adding (k, l, S, I) inequalities where the set S is generated in the same way as for the (ℓ, S) inequality, and we set $I = S$. Then, we enumerate all possible values of parameters k and l to obtain violated (k, l, S, I) inequalities. The detailed computational results of each algorithm are presented in Tables EC.1 and EC.2 in e-companion. Tables 5–8 summarize the algorithms' overall performance.

In Table 5, for both base and derived instances, we report the number of instances solved to optimality (column "Solved"), the number of unsolved instances

Table 5. Summarized Results of All Instances

	Base instances				Derived instances			
	Solved	Unsolved	Average gap (unsolved) (%)	Average cuts	Solved	Unsolved	Average gap (unsolved) (%)	Average. cuts
CPX	20	60	0.36	0.0	43	37	0.32	0.0
LS	37	43	0.30	161.4	53	27	0.28	150.1
C	40	40	0.34	141.6	55	25	0.29	132.1
2LS	44	36	0.29	87.9	56	24	0.25	108.6
MLS	55	25	0.28	608.2	62	18	0.23	547.9

Table 6. Summarized Results of All Instances

Base instances						Derived instances					
	Solved by						Solved by				
Unsolved by	**CPX**	**LS**	**C**	**2LS**	**MLS**	Unsolved by	**CPX**	**LS**	**C**	**2LS**	**MLS**
CPX	—	18	20	24	35	**CPX**	—	10	13	13	19
LS	1	—	6[a]	8	18	**LS**	0	—	3	3	9
C	0	3	—	6	15	**C**	1	1	—	2	7
2LS	0	1	2	—	11	**2LS**	0	0	1	—	6
MLS	0	0	0	0	—	**MLS**	0	0	0	0	—

[a]Indicates that **C** can solve six base instances that **LS** cannot.

(column "Unsolved"), the average gap before termination among unsolved instances (column "Average gap (unsolved)"), and the average number of user-defined cuts added before termination (column "Average cuts"). Although the (ℓ, S) inequalities are added to both **2LS** and **MLS** algorithms, we only count the number of two-echelon inequalities and L-echelon inequalities added. Note that the number of (ℓ, S) inequalities added to algorithms **LS**, **2LS**, and **MLS** are very close.

In Table 6, we compare the algorithms' ability to solve instances to optimality in 1 hour. Each entry in the table shows the number of instances unsolved by one algorithm but solved by another. For example, as shown in the table footnote, **C** can solve six base instances that **LS** cannot. According to our experiments, **MLS** is the most robust algorithm because all instances that **MLS** cannot prove to optimality in 1 hour are unsolved by other algorithms.

In Tables 7 and 8, we benchmark B&C algorithms' performances against CPLEX on 19 base instances and 42 derived instances that can all be solved to make the comparison fair. Additionally, we benchmark **2LS** and **MLS** against **LS** on 36 base instances and 53 derived instances that all three can solve. We report the algorithms' data on those instances with the average CPU seconds (column "Average time"), the ratio of the average CPU seconds between CPLEX (or **LS**) and other B&C algorithms (column "$\frac{\text{CPXTime}}{\text{B\&CTime}}$" or, similarly, column "$\frac{\text{LSTime}}{\text{B\&C Time}}$ for **LS**"), the average number of nodes explored (column "Average nodes"), and the ratio of the average number of nodes explored between CPLEX (or **LS**) and other B&C algorithms (column "$\frac{\text{CPXNodes}}{\text{B\&C Nodes}}$" or, similarly, column "$\frac{\text{LSNodes}}{\text{B\&C Nodes}}$ for **LS**").

On the basis of Tables 5–8, a few observations can be made. Algorithm **C** is not necessarily more effective than **LS**, even though the tested instances have

Table 7. Summarized Results of Base Instances

	19 instances solved by all algorithms				36 instances solved by **LS**, **2LS**, and **MLS**			
	Average time	CPX time / B&C time	Average nodes (×1,000)	CPX nodes / B&C nodes	Average time	LS time / B&C time	Average nodes (×1,000)	LS nodes / B&C nodes
CPX	1,331.2	1.00	312.8	1.00				
LS	635.8	2.09	131.0	2.39	1,222.0	1.00	226.1	1.00
C	645.8	2.06	126.4	2.48				
2LS	515.4	2.58	85.3	3.67	1,039.5	1.18	161.0	1.40
MLS	333.9	3.99	37.9	8.25	643.2	1.90	72.7	3.11

Table 8. Summarized Results of Derived Instances

	42 instances solved by all algorithms				53 instances solved by **LS**, **2LS**, and **MLS**			
	Average time	CPX time / B&C time	Average nodes (×1,000)	CPX nodes / B&C nodes	Average time	LS time / B&C time	Average nodes (×1,000)	LS nodes / B&C nodes
CPX	909.7	1.00	144.1	1.00	–	–	–	–
LS	431.9	2.11	66.5	2.17	686.3	1.00	102.8	1.00
C	453.5	2.01	71.8	2.01	–		–	–
2LS	360.2	2.53	42.4	3.40	549.7	1.25	68.5	1.50
MLS	227.6	4.00	20.1	7.19	356.9	1.92	31.2	3.30

constant capacities at each echelon. It may indicate that a more efficient way of generating strong mixing sets needs to be considered in future research. As we expected, the derived instances are much easier to solve than the base instances. Overall, algorithm **MLS** with multiechelon inequality (8) is the most effective one in obtaining optimal solutions or closing the optimality gap. Part of the reason for this could be that the average number of user cuts added by **MLS** is approximately five times that of other algorithms.

5. Conclusions

We studied the multiechelon serial lot-sizing problem with intermediate demands (MLS). Many existing studies have provided polynomial algorithms on MLS with demand occurring at the final echelon only or have shown that the multiple sources network flow problem is NP-hard. However, the complexity of MLS with a fixed-charge cost structure, a classic single-source network flow problem, remains unknown. As one of many contributions, this paper proves that MLS with a fixed-charge cost structure is NP-hard, which closes the theoretical gap.

We investigated both uncapacitated and capacitated MLS with different types of cost functions, such as general concave costs, fixed-charge costs, stepwise and nonconcave transportation cost, and fixed-charge transportation cost with no speculative motives. By considering intermediate demands, our results (see Table 1) generalize the findings of Zangwill (1969), Lee et al. (2003), van Hoesel et al. (2005), and Zhang et al. (2012), which are special cases of the problems we studied. In addition to developing efficient algorithms for solving both MULS and MCLS, we show that, in terms of complexities, our algorithms outperform many of the best-known algorithms in the literature (see Table 2).

Besides improving computational complexities, we studied the polyhedral structure of MLS in order to provide an efficient means to solve large instances of the multi-item, multiechelon lot-sizing problems. Again, our inequalities generalize many known inequalities, including the (ℓ, S), two-echelon, and (k, l, S, I) inequalities. Their efficacy is demonstrated in a comprehensive computational study. By implementing them in a B&C algorithm, we show that our algorithm is four times faster than CPLEX and approximately two times faster than the best-known algorithm in the literature.

There are a number of avenues that can be pursued in future. Currently, our development of dynamic programming algorithms for MULS and MCLS is of theoretical interest. It is worth studying heuristic algorithms by modifying proposed exact algorithms to obtain feasible solutions efficiently. There are many practical variations at the retailer level in a supply chain, such as the multi-item case with a big bucket capacity constraint, the backlogging case, and inventory bounds. Although our developed cutting planes are valid for some variations, it is more interesting to strengthen the cutting planes. In this paper, we assume a constant capacity at the first echelon. The more general case with varying capacity at the first as well as transportation echelons also needs to be studied. In addition, the complete convex hull of 2-ULS is still unknown. To improve practical applicability, we should consider algorithms to solve MLS in assembly systems.

Acknowledgments

The authors thank the area editor, the associate editor, and the referees for their constructive comments and suggestions on improving this manuscript.

References

Aggarwal A, Park J (1993) Improved algorithms for economic lot size problems. *Oper. Res.* 41(3):549–571.

Ahmed S, He Q, Li S, Nemhauser G (2016) On the computational complexity of minimum-concave-cost flow in a two–dimensional grid. *SIAM J. Optim.* 26(4):2059–2079.

Akartunalı K, Fragkos I, Miller AJ, Wu T (2016) Local cuts and two-period convex hull closures for big-bucket lot-sizing problems. *INFORMS J. Comput.* 28(4):766–780.

Atamtürk A, Küçükyavuz S (2005) Lot sizing with inventory bounds and fixed costs: Polyhedral study and computation. *Oper. Res.* 53(4):711–730.

Atamtürk A, Küçükyavuz S (2008) An $\mathcal{O}(n^2)$ algorithm for lot sizing with inventory bounds and fixed costs. *Oper. Res. Lett.* 36(3): 297–299.

Atamtürk A, Muñoz JC (2004) A study of the lot–sizing polytope. *Math. Programming* 99(3):43–65.

Barany I, van Roy T, Wolsey L (1984) Uncapacitated lot-sizing: The convex hull of solutions. *Math. Programming Study* 22:32–43.

Bitran GR, Yanasse HH (1982) Computational complexity of the capacitated lot size problem. *Management Sci.* 28(10):1174–1186.

Federgruen A, Tzur M (1991) A simple forward algorithm to solve general dynamic lot sizing models with n periods in $\mathcal{O}(n \log n)$ or $\mathcal{O}(n)$ time. *Management Sci.* 37(8):909–925.

Florian M, Klein M (1971) Deterministic production planning with concave costs and capacity constraints. *Management Sci.* 18(1):12–20.

Florian M, Lenstra JK, Rinnooy Kan AHG (1980) Deterministic production planning: Algorithms and complexity. *Management Sci.* 26(7):669–679.

Fragkos I, Degraeve Z, Reyck BD (2016) A horizon decomposition approach for the capacitated lot-sizing problem with setup times. *INFORMS J. Comput.* 28(3):465–482.

Gaglioppa F, Miller LA, Benjaafar S (2008) Multitask and multistage production planning and scheduling for process industries. *Oper. Res.* 56(4):1010–1025.

Günlük O, Pochet Y (2001) Mixing mixed-integer inequalities. *Math. Programming* 90(3):429–457.

He Q, Ahmed S, Nemhauser G (2015) Minimum concave cost flow over a grid network. *Math. Programming* 150(1):79–98.

Hwang H (2010) Economic lot-sizing for integrated production and transportation. *Oper. Res.* 58(2):428–444.

Hwang H, Ahn H, Kaminsky P (2013) Basis paths and a polynomial algorithm for the multistage production-capacitated lot-sizing problem. *Oper. Res.* 61(2):469–482.

Kaminsky P, Simchi-Levi D (2003) Production and distribution lot sizing in a two stage supply chain. *IIE Trans.* 35(11):1065–1075.

Krarup K, Bilde O, Collatz L, ed. (1977) Plant location, set covering and economic lot-sizes: an $O(mn)$ algorithm for structured problems. *Optimierung bei Graphentheoretischen und Ganzzahligen Probleme* (Birkhauser, Basel, Switzerland), 155–179.

Küçükyavuz S, Pochet Y (2009) Uncapacitated lot sizing with backlogging: the convex hull. *Math. Programming* 118(1):151–175.

Lee C, Çetinkaya S, Jaruphongsa W (2003) A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse. *Oper. Res.* 51(5):735–747.

Love S (1972) A facilities in series inventory model with nested schedules. *Management Sci.* 18(5):327–338.

Melo R, Wolsey L (2010) Uncapacitated two-level lot-sizing. *Oper. Res. Lett.* 38:241–245.

Pochet Y, Wolsey L (2006) *Production Planning by Mixed Integer Programming* (Springer, New York).

Pochet Y, Wolsey LA (1993) Lot-sizing with constant batches: Formulation and valid inequalities. *Math. Oper. Res.* 18(4): 767–785.

Rardin R, Wolsey L (1993) Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *Eur. J. Oper. Res.* 71(1):95–109.

van Hoesel C, Wagelmans A (1996) An $\mathcal{O}(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Sci.* 42(1):142–150.

van Hoesel S, Romeijn HE, Morales DR, Wagelmans A (2005) Integrated lot sizing in serial supply chains with production capacities. *Management Sci.* 51(11):1706–1719.

Van Vyve M (2007) Algorithms for single-item lot-sizing problems with constant batch size. *Math. Oper. Res.* 32(3):594–613.

Van Vyve M, Wolsey LA, Yaman H (2014) Relaxations for two-level multi-item lot-sizing problems. *Math. Programming* 146(1–2): 495–523.

Wagelmans A, Van Hoesel S, Kolen A (1992) Economic lot sizing: An $\mathcal{O}(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Oper. Res.* 40(1):145–156.

Wagner H, Whitin T (1958) Dynamic version of the economic lot size problem. *Management Sci.* 5(1):89–96.

Zangwill W (1969) A backlogging model and a multiechelon model of a dynamic economic lot size production system—A network approach. *Management Sci.* 15(9):506–527.

Zhang M, Küçükyavuz S, Yaman H (2012) A polyhedral study of multiechelon lot sizing with intermediate demands. *Oper. Res.* 60(4):918–935.

Ming Zhao is an assistant professor of operations management in the Department of Business Administration at the University of Delaware. His research interests include supply chain risk management, healthcare operations, optimal learning, stochastic and mixed-integer optimization.

Minjiao Zhang is an associate professor of quantitative analysis in the Department of Economics, Finance and Quantitative Analysis at Kennesaw State University. Her research interests are in the areas of mixed-integer optimization and stochastic programming with applications in production planning, healthcare, and disaster planning.