

The multi-level lot sizing problem with flexible production sequences

JOSEPH BEGNAUD, SAIF BENJAAFAR* and LISA A. MILLER

Graduate Program in Industrial & Systems Engineering, Department of Mechanical Engineering, University of Minnesota, 111 Church St S.E., Minneapolis, MN 55455, USA
E-mail: saif@umn.edu

Received October 2006 and accepted December 2008

This paper considers a multi-level/multi-machine lot sizing problem with flexible production sequences, where the quantity and combination of items required to produce another item need not be unique. The problem is formulated as a mixed-integer linear program and the notion of echelon inventory is used to construct a new class of valid inequalities, which are called echelon cuts. Numerical results show the computational power of the echelon cuts in a branch-and-cut algorithm. These inequalities are compared to known cutting planes from the literature and it is found that, in addition to being strong and valid for the flexible production case, echelon cuts are at least as strong as certain classes of known cuts in the restricted fixed production setting.

Keywords: Production planning, lot sizing, echelon inventory, flexible production sequences, integer programming

1. Introduction

We consider a *lot sizing* problem with multiple items where the objective is to determine when and how much of each item should be produced over a finite planning horizon so as to minimize the sum of production, inventory holding and setup costs. The choice of a production plan is constrained by the need to satisfy all the demand that arises in each period for each item. The problem we consider has multiple levels (or stages), where an item might serve as an input in the production of one or more other items and may itself be the output of production involving one or more other items. In this paper, we consider systems where the production sequences (the series of tasks and the set of input items for each task) used in producing an item can be flexible. Furthermore, we allow for the possibility of multiple flexible machines that may be able to perform one or more tasks. Hence, our treatment of lot sizing with multiple items, multiple levels and multiple machines is general and includes structures such as series, assembly and distribution systems, among others.

In contrast to standard multi-level lot sizing problem formulations, our treatment differentiates between items and production tasks, which allows for the possibility of alternate bills of materials or recipes. This allows us to model potential input substitutions, where it is possible to sub-

stitute one component with another (typically of a higher quality). Component substitution is a common practice in industries, such as computer manufacturing, where it is used as a strategy for mitigating inventory shortages or to take advantage of differences in component costs from period to period. For example, Dell, which manufactures computers in an assemble-to-order fashion, is known to resort to component substitution (e.g., substituting a lower grade chip with one of higher grade) in order to fulfill customer orders within the quoted lead-time. Our formulation also allows us to model settings where there is strong commonality in the input items of various tasks. Such commonality arises in settings, such as electronics manufacturing, where many components are shared among different end products. Finally, our formulation permits the modeling of production processes with flexible machines, a common feature in industries such as metal cutting, printed circuit board assembly and chemical manufacturing.

We formulate the problem as a Mixed-Integer Linear Program (MILP). We use the notion of echelon inventory to construct a set of valid inequalities (cuts) and show via numerical results that they can significantly improve solution time and solution quality. In addition to being strong and valid for the flexible production case, we also show that our inequalities are at least as strong as those proposed by Belvaux and Wolsey (2001) for the multi-level fixed production case.

*Corresponding author

There is an extensive literature on the lot sizing problem. A survey of important results and solution approaches can be found in Pochet and Wolsey (2006); see also Miller and Wolsey (2003). The literature on lot sizing with multiple levels is relatively less extensive. A review of recent advances can be found in Stadtler (2003).

For uncapacitated systems (systems with no constraints on production capacity), Afentakis *et al.* (1984) consider assembly structures and propose a solution approach using decomposition. Afentakis and Gavish (1986) extend these results to systems with general production structures. For capacitated systems, the problem is NP-hard, and even finding a feasible solution can be computationally challenging; see Pochet and Wolsey (2006). Consequently, several researchers have proposed heuristics. Simpson and Erenguc (1998) propose a neighborhood search method to merge multiple smaller lots into larger lots in cases where the reduction in setup cost due to batching is sufficient to justify the increased holding cost. Tempelmeier and Derstroff (1996) decompose the multi-level structure into single-level problems using a Lagrangian relaxation. Harrison and Lewis (1996) propose a heuristic for capacitated serial systems based on repeatedly solving Linear Programming (LP) subproblems and modifying constraint coefficients to implicitly account for capacity consumption. Afentakis (1987) proposes a relaxation which decomposes the problem into T (time horizon) stages, where the set of feasible solutions at stage t ($t \in \{1, \dots, T\}$) is limited by the variables already fixed in stages $1, \dots, t-1$ (for periods $1, \dots, t-1$). This method works well for uncapacitated problems, but can result in infeasibilities for capacitated problems. Katok *et al.* (1998) apply LP-based rounding procedures to multi-level capacitated problems, where setup variables are iteratively fixed based on the output of previous LP-relaxations.

In addition to heuristics, there has been progress in solving some problems to optimality by adding strong valid inequalities to the MILP formulations of these problems. Many of the inequalities, such as those proposed by Pochet and Wolsey (1991), are extensions of inequalities derived for single-level problems, such as the (l, S) inequalities of Barany *et al.* (1984). These single-level inequalities can be extended to the multi-level case via a reformulation of the problem in terms of echelon inventory (see, for example, Afentakis and Gavish (1986), Belvaux and Wolsey (2000), Belvaux and Wolsey (2001) and Wolsey (2002)). Such reformulations have their origin in the seminal paper by Clark and Scarf (1960), who first introduced the notion of echelon inventory.

In problems where production capacity in each period is an integer multiple of some fixed level, Pochet and Wolsey (1993) derive facet-defining (k, l, S, I) inequalities. Constantino (1996) derives extended (k, l, S, I) and supermodular inequalities for a version of the single-item

capacitated case and applies these inequalities as cutting planes for the multi-item case. Miller *et al.* (2003a) derive cover and reverse cover inequalities for the multi-item capacitated case with setup times, again applying polyhedral results obtained for structured problem relaxations in order to strengthen their initial formulation; additional results can be found in Miller *et al.* (2003b). Miller and Wolsey (2003) derive tight problem formulations for various discrete lot sizing problems, many of which result in equivalent integral LP formulations. Belvaux and Wolsey (2000) describe a software implementation, *bc-prod*, designed to solve many common lot sizing problems to optimality.

Although the models and inequalities we develop in this paper have similar intuition and flavor as some of the above literature, the fundamental difference is the inclusion in our work of flexible production sequences, which complicates the calculation of echelon inventory. This added complexity requires special consideration in the development of the model formulation and valid inequalities in this paper, and prevents the direct extension of inequalities developed in previous work. Gaglioppa *et al.* (2008) do consider a problem with flexible production sequences and use a similar notion of echelon inventory to construct valid inequalities, but their setting is different. They treat a “small-bucket” problem where they determine the exact timing of tasks. Furthermore, in their case, tasks are associated with variable production quantities and these quantities are decision variables. Their problem is motivated by applications in process industries while ours is more appropriate for discrete manufacturing where the production quantities associated with individual tasks tend to be fixed.

The rest of the paper is organized as follows. In Section 2, we formulate our problem as a MILP for the case of a system with a single machine. In Section 3, we describe the notion of echelon inventory and use it to derive valid inequalities. In Section 4, we extend our formulation and inequalities to the multi-machine case. In Section 5, we present computational results. In Section 6, we provide a brief summary and suggestions for future research.

2. Description and model formulation: the single-machine case

We shall refer to our multi-level flexible production lot sizing problem as the MLFP. We first consider a system consisting of a single machine and a set of items, R , which are produced via a set of tasks, N . Each task may require multiple items as input, but each task has a single-item output. This assumption, while not necessary for the validity of the problem formulation below, is needed for the validity of the cuts we develop in Section 4 (see also discussion in Section 6). We let $\rho^{i,r}$ denote the number of units of item

r required to initiate one unit of task i , $\sigma^{i,r}$ the number of units of item r produced by one completion of task i and α_i the processing time of task i . We assume no direct product recycling, thus for any task i and item r , $\rho^{i,r}$ and $\sigma^{i,r}$ cannot both be positive. We use this assumption to simplify the proof of Theorem 5, but note that if $\rho^{i,r}$ and $\sigma^{i,r}$ were truly both positive for a manufacturing process, then we could use $\hat{\sigma}^{i,r} = \sigma^{i,r} - \rho^{i,r}$ and $\hat{\rho}^{i,r} = 0$ as the input and output parameters, respectively. Capacity may vary from period to period and we let C_t refer to the available production capacity (in units of time) in period t , where $t = 1, \dots, T$ is the planning horizon. External demand for each item may vary from period to period and we refer to external demand for item r in period t as d_t^r . Each time a task is initiated, we incur a production cost f_t^i . In addition, a setup cost g_t^i is incurred if task i is performed during period t . We assume that a task could be carried out multiple times during a period, but this number is constrained by the available capacity. Although the production cost is incurred each time a task is carried out, the setup cost is incurred only the first time the task is initiated in each period.

We assume that periods are sufficiently long so that tasks initiated in a period are always completed in that period. Also, we assume that items produced in one period can be used in the production of other items in the same period. This assumption is a standard one in the “big-bucket” lot sizing literature (see, for example, Pochet and Wolsey (2006)). This assumption is justified when the length of a period is much longer than the processing times of individual tasks (for instance, the length of a period is a week while processing times are minutes or hours). Inventory of an item r carried over from period t to period $t + 1$ incurs a holding cost h_t^r . Note that the holding cost for an item is assumed to be the same regardless of which task produced it and which corresponding input items were used. This is consistent with industry accounting practices where, once produced, items of the same type are not differentiated based on how they were produced. Differences in the costs of different input items are, however, taken into account via the production cost parameters f_t^i . For example, a task that produces the same item as another task but uses more expensive components may have a higher production cost.

The objective of the MLFP is to determine a production plan, specified in terms of the number of tasks of each type to carry out in each period in order to meet demand while minimizing the sum of production setup and inventory holding costs. We define the following decision variables:

- x_t^i = production variable equal to the number of times task i is performed in period t ;
- y_t^i = setup variable equal to one if production of task i occurs in period t and otherwise equal to zero;
- s_t^r = inventory level of item r at the end of period t .

The MLFP can now be formulated as the following MILP:

$$\min \sum_{t \in T} \sum_{i \in N} f_t^i x_t^i + \sum_{t \in T} \sum_{i \in N} g_t^i y_t^i + \sum_{t \in T} \sum_{r \in R} h_t^r s_t^r, \quad (1)$$

subject to

$$s_t^r = s_{t-1}^r + \sum_{i \in N} \sigma^{ir} x_t^i - \sum_{i \in N} \rho^{ir} x_t^i - d_t^r \quad \forall r \in R, t \in T, \quad (2)$$

$$\sum_{i \in N} \alpha_i x_t^i \leq C_t \quad \forall t \in T, \quad (3)$$

$$x_t^i \leq \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor y_t^i \quad \forall i \in N, t \in T, \quad (4)$$

$$x_t^i \in \mathbb{Z}^+, \quad y_t^i \in \{0, 1\} \quad \forall i \in N, t \in T, \quad (5)$$

$$s_t^r \geq 0 \quad \forall r \in R, t \in T. \quad (6)$$

In the above formulation, constraints (2) ensure balance of flow for all items in each period. Constraints (3) enforce limits on machine capacity. Constraints (4) ensure that a setup is incurred if a task is carried out in a particular period.

The unique feature of the MLFP is that the item–task formulation allows for alternate bill of materials or recipes. Just as the same item can be an output from multiple tasks, the same set of items may serve as inputs, in varying quantities and combinations, to multiple tasks. Consequently, the same item may be produced via different production sequences requiring varying combinations and quantities of input items along the way. This is illustrated in Fig. 1 for an example system, with items represented by circles and tasks shown as ellipses. In this example, item 1 can be produced by either task A or B, and so the quantities of input items (2, 3 and 4) required to produce a particular unit of item 1 depend on which task (A or B) produced that unit. Of course, problems with fixed production sequences are special cases of the MLFP. They correspond to problems where, for each item r , we have $\sigma^{i,r} > 0$ for exactly one $i \in N$. In other words, there is a one-to-one correspondence between items and the task that produces them. In this case, the formulation can be greatly simplified by eliminating the notion of tasks (the production and

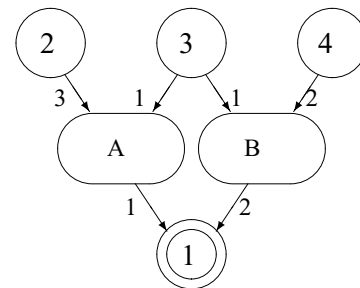


Fig. 1. Example of a multi-level production system with flexible sequences.

setup variables can now be associated with items instead of tasks).

As mentioned in the literature review, instances of our problem, such as the single-stage capacitated lot sizing problem, are known to be NP-hard. Therefore, our problem is also NP-hard. In the next section, we introduce a new set of valid inequalities that can be added to our formulation to yield tighter LP relaxations, which in turn can reduce the computational effort needed to find optimal solutions or improve the quality of suboptimal solutions obtained within a specified time frame.

3. Valid inequalities

We derive two sets of valid inequalities in this section, inspired respectively by the (l, S) inequalities of Barany *et al.* (1984) and the work of Gaglioppa *et al.* (2008); see also Pochet and Wolsey (2006) for more general families of valid inequalities, of which the (l, S) inequalities are special cases.

3.1. Single-level inequalities

We make the following important observation. The local inventory of an item r can only increase in period t if there is production of some task $i \in O(r)$, where $O(r) = \{i \in N \mid r \text{ is an output item of task } i\}$. On the other hand, if there is no production of any tasks $i \in O(r)$ in a time interval $[k, l]$, then the local inventory of r cannot increase in that interval, that is $s_{k-1}^r \geq s_l^r$. Based on this observation, Theorem 1 states that if the local inventory of item r at the beginning of interval $[k, l]$ is insufficient to satisfy external demand for r in that interval, then sufficient production of item r must occur in that interval.

Theorem 1. *The following Single-Level Inequalities ($SLI_{r,k,l}$) are valid for the MLFP:*

$$s_{k-1}^r \geq \sum_{t=k}^l \left[d_t^r \left(1 - \sum_{i \in O(r)} \sum_{u=k}^t y_u^i \right) \right] \quad \forall r \in R, k = 1, \dots, T, l = k, \dots, T. \quad (7)$$

Proof. First, we introduce a new decision variable $s_{k-1,t}^r$ defined as the quantity of inventory of item r available at the end of period $k - 1$ but not used to satisfy external demand for r until period t ($t \geq k$). For a fixed r and interval $[k, l]$:

$$s_{k-1}^r \geq \sum_{t=k}^l s_{k-1,t}^r \quad (8)$$

$$\geq \sum_{t=k}^l \left[d_t^r \left(1 - \sum_{i \in O(r)} \sum_{u=k}^t y_u^i \right) \right]. \quad (9)$$

As s_{k-1}^r is the *total* amount of inventory of r at the end of period $k - 1$, then Equation (8) is valid by the definition of $s_{k-1,t}^r$. Considering a given $s_{k-1,t}^r$ in Equation (8), if $y_u^i = 1$ for at least one $i \in O(r)$, $u \in [k, t]$, then $s_{k-1,t}^r \geq d_t^r (1 - \sum_{i \in O(r)} \sum_{u=k}^t y_u^i)$ is trivially valid. On the other hand, if $y_u^i = 0$ for all $i \in O(r)$, $u \in [k, t]$, thus there is *no* direct production of item r in the interval $[k, t]$, so *all* external demand for r in t must be satisfied by local inventory of r available at the beginning of the interval. Therefore, $s_{k-1,t}^r = d_t^r = d_t^r (1 - \sum_{i \in O(r)} \sum_{u=k}^t y_u^i)$. As either of the above two cases must be true, inequality (9) is valid. ■

Note that if there is no external demand for item r in period $l + 1$ ($d_{l+1}^r = 0$), then inequality $SLI_{r,k,l+1}$ is identical to inequality $SLI_{r,k,l}$ (for $l \geq k$). Therefore, we need only consider those values of l ($l \geq k$) with positive external demand for item r . That is, if we define the set of periods $L(r, k) = \{t \in T \mid t \geq k \text{ and } d_t^r > 0\}$, then the set of *all* single-level inequalities is exactly the set of single-level inequalities over all $r \in R$, $k = 1, \dots, T$, and $l \in L(r, k)$.

3.2. Echelon inequalities

The single-level inequalities require production of an item to occur if the local inventory of that item is insufficient to satisfy *external* demand for that item over some time horizon. However, they ignore *induced* demand for intermediate items that may be required as input to certain tasks. To address this limitation, we use the notion of echelon inventory to develop a stronger set of cuts, which consider external demand as well as internal demand for intermediate items.

The *echelon inventory* of item r equals the sum of local inventory of r and the total amount of r contained in the downstream items produced via tasks that required r for production. Echelon inventories can be calculated from local item inventory levels for fixed production structures. This calculation is more difficult for flexible production structures, as the production components of an item that can be made via alternate tasks is not known *a priori*. One option to deal with this difficulty is to introduce additional variables to keep track of how each unit of each item was produced. However, this could make the problem excessively large. Another option, which we pursue here, is to use lower bounds on the exact value of echelon inventory to construct valid cuts. As we show, these lower bounds can be computed without the introduction of additional integer variables.

For ease of exposition, and to help develop intuition, we start by discussing the case where production sequences are fixed. We then show how the same approach, with appropriate modifications, can be extended to the case of flexible sequences. We define item g to be a *successor* of item r if g can be produced by a task with r , or a successor of r , as an input to the task. We denote the set of all successor items of r as $S(r)$. Furthermore, if item g is a successor of item r ,

then we call r a *predecessor* of g . We define the parameter $\delta^{r,g}$, which we call the *coefficient of transformation*, as the amount of item r required per unit of production of item g . In the case where multi-unit production batches are required, we do allow for fractional $\delta^{r,g}$ values, where input items are allocated equally over the size of the production batch. In calculating $\delta^{r,g}$ for an item g produced by multiple tasks, we assign $\delta^{r,g}$ the value associated with the sequences of tasks requiring the *least* amount of r per unit of item g output (we let $\delta^{r,r} = 1$ for all $r \in R$).

Note that, assuming no cycles in production, the items in R can be labeled from $1, \dots, |R|$ so that every item has a larger label than its successor items. Given such an ordering, the $|R| \times |R|$ $\delta^{r,g}$ -matrix can be calculated in $O(|R|^2|N|)$ time, by considering values for item g sequentially from $|R|$ to 1, and item r from $|R|$ to $g + 1$ (the pool of possible predecessor items). In systems with fixed production sequences, for a given choice of distinct items r and g , $\delta^{r,g}$ is given by

$$\delta^{r,g} = \left(\frac{\sum_{h \in R} \delta^{r,h} \rho^{i,h}}{\sigma^{i,g}} \right). \quad (10)$$

The above values are computed recursively, so that computing $\delta^{r,g}$ takes advantage of previously computed $\delta^{r,h}$ for $h > g$.

We define the *echelon inventory* of item r as the sum of local inventory of r and the amount of r contained in the downstream items requiring r for production, where the latter is determined (in the case of fixed production sequences) by $\delta^{r,g}$ for all $g \in S(r)$. That is, echelon inventory of item r at the end of period t (e_t^r) is defined as

$$e_t^r = s_t^r + \sum_{g \in S(r)} \delta^{r,g} s_t^g = \sum_{g \in R} \delta^{r,g} s_t^g, \quad (11)$$

where the second equality holds because $\delta^{r,r} = 1$ and $\delta^{r,g} = 0$ for all $g \in N \setminus (\{r\} \cup S(r))$.

Note that the echelon inventory for item r does not increase when production of a task downstream from r occurs. Therefore, this echelon inventory can only increase when item r is produced directly.

Property 1. *The completion of a task for which an item r is an output is the only event that increases the echelon-inventory of r .*

Property 1 implies that if a task i does not produce item r directly ($i \in N \setminus O(r)$), the change in the echelon inventory of r resulting from a run of task i cannot be positive.

For systems with fixed production sequences, we can now show that the inequalities described in the following theorem are valid.

Theorem 2. *For systems with fixed production sequences, the following Echelon Inequalities ($EI_{r,k,l}$) are valid for the*

MLFP:

$$e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \min \left\{ D_{t,l}^r, \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right\} y_t^i \geq D_{k,l}^r \\ \forall r \in R, k = 1, \dots, T, l = k, \dots, T, \quad (12)$$

where $D_{k,l}^r$ is defined as $D_{k,l}^r = \sum_{t=k}^l \sum_{g \in R} \delta^{r,g} d_t^g$.

The inequalities in Theorem 2 are a special case of the inequalities in Theorem 3, which is introduced and proved later in this section. We clarify the intuition behind them here. The inequalities state that if the echelon inventory of r at the beginning of interval $[k, l]$ is insufficient to satisfy echelon demand for r in that interval, then sufficient production of item r must occur in the interval. For each $i \in O(r)$, the echelon inventory of r increases by $\sigma^{i,r}$ units each time task i is performed. Therefore, the maximum increase in the echelon inventory of r associated with production of task i in period t is

$$\left(\left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right),$$

due to the capacity restriction limiting production to $\lfloor C_t/\alpha_i \rfloor$. On the other hand, the maximum portion of the overall echelon-demand for r over the interval $[k, l]$ that can be satisfied via production of any task in period t ($t \in [k, l]$) is the portion remaining from period t to period l ($D_{t,l}^r$). That is, we cannot satisfy demand for an earlier period ($k, \dots, t-1$) via production in period t . Consequently, the coefficient of y_t^i corresponds to the minimum of these two upper bounds. Note that the echelon inequalities can be restricted to the set of those inequalities with positive $D_{k,l}^r > 0$; i.e., to the set of periods in the set $L^{\text{ech}}(r, k) = \{t \mid \sum_{g \in R} \delta^{r,g} d_t^g > 0, t \geq k\}$.

The use of echelon-based inequalities is not new for systems with fixed production sequences. Belvaux and Wolsey (2001) discuss the transformation of the following local inventory inequalities:

$$s_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} d_{t,l}^r y_t^i \geq d_{k,l}^r \\ \forall r \in R, k = 1, \dots, T, l = k, \dots, T, \quad (13)$$

where $d_{a,b}^r = \sum_{t=a}^b d_t^r$, into ones involving echelon inventory as follows:

$$e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} D_{t,l}^r y_t^i \geq D_{k,l}^r \\ \forall r \in R, k = 1, \dots, T, l = k, \dots, T. \quad (14)$$

Inequalities (12) and (14) are identical in the uncapacitated case. On the other hand, in the capacitated case, if $\lfloor C_t/\alpha_i \rfloor \sigma^{i,r} < D_{t,l}^r$ for some $i \in O(r)$ and $t \in [k, l]$, then inequality (12) dominates inequality (14). Belvaux and Wolsey (2001) also discuss cuts similar to Equation (14), where the y_t^i -variable coefficients $D_{t,l}^r$ are replaced by the

capacity term $\lfloor C_t/\alpha_i \rfloor \sigma^{i,r}$. By similar reasoning, if $D'_{t,l} < \lfloor C_t/\alpha_i \rfloor \sigma^{i,r}$ for some $i \in O(r)$ and $t \in [k, l]$, then inequality (12) would dominate these capacity coefficient inequalities as well.

Next, we show how echelon-inventory-based constraints can be derived for systems with flexible sequences. This extension is not straightforward because, as we mentioned earlier, echelon inventory cannot be easily determined in this case. For example, consider an item g which can be produced via two unique tasks. The combination and amounts of predecessor items contained in a unit of item g will depend upon which task produced that particular unit of g . Therefore, computing the echelon inventory based on item g is impossible without tracking how each unit of item g was produced, requiring a large investment in additional binary variables in the formulation. An alternative approach, which we adopt in this paper, is to seek a lower bound on item echelon inventory.

In particular, we define the parameter $\underline{\delta}^{r,g}$, which we call the *minimum coefficient of transformation*, as the minimum amount of item r required per unit of production of item g . For a pair of distinct items r and g , $\underline{\delta}^{r,g}$ is given by

$$\underline{\delta}^{r,g} = \min_{i \in O(g)} \left(\frac{\sum_{h \in R} \underline{\delta}^{r,h} \rho^{i,h}}{\sigma^{i,g}} \right), \quad (15)$$

where the parameters $\underline{\delta}^{r,g}$ are computed recursively. Note that for systems with fixed production sequences, we have $\underline{\delta}^{r,g} = \delta^{r,g}$.

Example 1. Consider the production system shown in Fig. 1. In calculating the value of $\delta^{3,1}$, the algorithm assigns a value according to which task (A or B) requires the least amount of item 3 as input per unit of item 1 output. For task A, we get

$$\begin{aligned} \left(\frac{\sum_{h \in \{2,3\}} \underline{\delta}^{3,h} \rho^{A,h}}{\sigma^{A,1}} \right) &= \left(\frac{\underline{\delta}^{3,2} \rho^{A,2} + \underline{\delta}^{3,3} \rho^{A,3}}{\sigma^{A,1}} \right) \\ &= \left(\frac{0 \times 3 + 1 \times 1}{1} \right) = 1. \end{aligned}$$

Whereas for task B, we get

$$\begin{aligned} \left(\frac{\sum_{h \in \{3,4\}} \underline{\delta}^{3,h} \rho^{B,h}}{\sigma^{B,1}} \right) &= \left(\frac{\underline{\delta}^{3,3} \rho^{B,3} + \underline{\delta}^{3,4} \rho^{B,4}}{\sigma^{B,1}} \right) \\ &= \left(\frac{1 \times 1 + 0 \times 2}{2} \right) = \frac{1}{2}. \end{aligned}$$

This leads to $\underline{\delta}^{3,1} = 1/2$.

We define the *lower-bound-on-echelon-inventory* of item r as the sum of local inventory of r and the lower bound on the amount of r contained in the downstream items requiring r for production, where the lower bound is determined by $\underline{\delta}^{r,g}$ for all $g \in S(r)$. That is, the lower-bound-on-echelon-inventory of item r at the end of period t (\underline{e}'_t) is

defined as

$$\underline{e}'_t = s'_t + \sum_{g \in S(r)} \underline{\delta}^{r,g} s'_t{}^g = \sum_{g \in R} \underline{\delta}^{r,g} s'_t{}^g. \quad (16)$$

Obviously, for systems with fixed production sequences, $\underline{e}'_t = e'_t$. Similar to the exact echelon inventory, the echelon inventory lower bound for item r does not increase when production of a task downstream from r occurs. The lower bound can only increase when item r is produced directly.

Property 2. *The completion of a task for which an item r is an output is the only event that increases the lower-bound-on-echelon-inventory of r .*

Proof. Consider an item $r \in R$ and task $i \in N \setminus O(r)$. By definition (15) of $\underline{\delta}^{r,g}$, for materials $r, g \in R$ and task $i \in N \setminus O(r)$:

$$\underline{\delta}^{r,g} \sigma^{i,g} \leq \sum_{h \in R} \underline{\delta}^{r,h} \rho^{i,h}. \quad (17)$$

Therefore

$$\sum_{\{g \in R \mid \sigma^{i,g} > 0\}} \underline{\delta}^{r,g} \sigma^{i,g} \leq \sum_{\{g \in R \mid \sigma^{i,g} > 0\}} \sum_{h \in R} \underline{\delta}^{r,h} \rho^{i,h}. \quad (18)$$

By the assumption that every task has only one item type as output, this implies:

$$\sum_{g \in R} \underline{\delta}^{r,g} \sigma^{i,g} \leq \sum_{h \in R} \underline{\delta}^{r,h} \rho^{i,h}. \quad (19)$$

Therefore, completion of any task $i \in N \setminus O(r)$ cannot increase the lower-bound-on-echelon-inventory of r . ■

We are now ready to state the main result of this paper, which generalizes Theorem 2 to the case of systems with flexible production sequences.

Theorem 3. *The following Echelon Inequalities ($EI_{r,k,l}$) are valid for the MLFP:*

$$\begin{aligned} \underline{e}'_{k-1} + \sum_{t=k}^l \sum_{i \in O(r)} \min \left\{ \underline{D}'_{t,l}, \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right\} y_t^i &\geq \underline{D}'_{k,l} \\ \forall r \in R, k = 1, \dots, T, l = k, \dots, T, \end{aligned} \quad (20)$$

where $\underline{D}'_{k,l} = \sum_{t=k}^l \sum_{g \in R} \underline{\delta}^{r,g} d_t^g$.

Proof. First, we show that the following $|R|$ echelon-based single-item relaxations of MLFP are valid for all $r \in R$:

$$\underline{e}'_{k-1} + \sum_{t=k}^l \sum_{i \in O(r)} x_t^i \sigma^{i,r} \geq \underline{D}'_{k,l} \quad \forall k \in [1, T], l \in [k, T], \quad (21)$$

$$x_t^i \leq \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor y_t^i \quad \forall i \in N, t \in T, \quad (22)$$

$$x_t^i \in \mathbb{Z}^+, \quad y_t^i \in \{0, 1\} \quad \forall i \in N, t \in T, \quad (23)$$

$$\underline{e}'_t \geq 0 \quad \forall r \in R, t \in T. \quad (24)$$

Constraints (22) and (23) are valid from the formulation of MLFP, whereas constraints (24) are valid from the definition of the variables e_t^r . To show that inequalities (21) are valid, consider the following. By combining Equations (2) and (16), we obtain:

$$e_{k-1}^r = e_k^r + \sum_{i \in N} \sum_{g \in R} \underline{\delta}^{r,g} (\rho^{i,g} - \sigma^{i,g}) x_k^i + \underline{D}_{k,k}^r. \quad (25)$$

Using recursion on Equation (25) for $l \geq k$ leads to

$$e_{k-1}^r = e_l^r + \sum_{t=k}^l \sum_{i \in N} \sum_{g \in R} \underline{\delta}^{r,g} (\rho^{i,g} - \sigma^{i,g}) x_t^i + \underline{D}_{k,l}^r, \quad (26)$$

which may be rewritten as

$$\begin{aligned} e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \sigma^{i,r} x_t^i &= e_l^r + \sum_{t=k}^l \sum_{i \in N \setminus O(r)} \sum_{g \in R} \underline{\delta}^{r,g} \\ &\times (\rho^{i,g} - \sigma^{i,g}) x_t^i + \underline{D}_{k,l}^r, \end{aligned} \quad (27)$$

by separating tasks in the set $O(r)$ from those in the set $i \in N \setminus O(r)$ and noting that r is the only item output from tasks in the set $O(r)$. Using inequalities (19) for tasks in the set $N \setminus O(r)$ yields inequalities (21).

We now prove the validity of the Echelon Inequalities of Theorem 3 by proving their validity for the single-item relaxations of MLFP above. Consider fixed choices for $r \in R$, $k \in [1..T]$, and $l \in [k..T]$, defining a particular echelon inequality $EI_{r,k,l}$.

Case 1. Suppose there is no y_u^i -variable ($i \in O(r)$, $u \in [k, l]$) equal to one such that $\underline{D}_{u,l}^r < \lfloor C_u / \alpha_i \rfloor \sigma^{i,r}$. Then

$$\begin{aligned} e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \min \left\{ \underline{D}_{t,l}^r, \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right\} y_t^i \\ &= e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} y_t^i \\ &\geq e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \sigma^{i,r} x_t^i \quad (28) \\ &\geq \underline{D}_{k,l}^r. \quad (29) \end{aligned}$$

Inequalities (28) and (29) follow from the relaxation constraints (22) and (21), respectively.

Case 2. Let $\hat{u} \in [k, l]$ be the earliest period for which there exists an $\hat{i} \in O(r)$ with $y_{\hat{u}}^{\hat{i}} = 1$ and $\underline{D}_{\hat{u},l}^r <$

$\lfloor C_{\hat{u}} / \alpha_{\hat{i}} \rfloor \sigma^{\hat{i},r}$. Then

$$\begin{aligned} e_{k-1}^r + \sum_{t=k}^l \sum_{i \in O(r)} \min \left\{ \underline{D}_{t,l}^r, \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right\} y_t^i \\ &= e_{k-1}^r + \sum_{t=k}^{\hat{u}-1} \sum_{i \in O(r)} \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} y_t^i \\ &\quad + \sum_{t=\hat{u}}^l \sum_{i \in O(r)} \min \left\{ \underline{D}_{t,l}^r, \left\lfloor \frac{C_t}{\alpha_i} \right\rfloor \sigma^{i,r} \right\} y_t^i \\ &\geq e_{k-1}^r + \sum_{t=k}^{\hat{u}-1} \sum_{i \in O(r)} \sigma^{i,r} x_t^i + \underline{D}_{\hat{u},l}^r \quad (30) \end{aligned}$$

$$\begin{aligned} &\geq \underline{D}_{k,\hat{u}-1}^r + \underline{D}_{\hat{u},l}^r \\ &= \underline{D}_{k,l}^r. \quad (31) \end{aligned}$$

Inequality (30) is valid from constraints (22) and the assumption regarding $y_{\hat{u}}^{\hat{i}}$ for this case. Inequality (31) is valid from constraints (21).

The Echelon Inequalities for an item $r \in R$ are valid for the echelon-based single-item relaxation of MLFP for r , and, therefore, they are also valid for MLFP. ■

Note that the interpretation and intuition of the echelon inequalities in Theorem 3 are similar to those of the inequalities presented in Theorem 2. In fact, the above inequalities reduce to those in Theorem 2 when the production sequences are fixed. Note here too that the echelon inequalities can be restricted to the set of those inequalities with positive $\underline{D}_{k,l}^r > 0$; i.e., to the set of periods in the set $L_{\text{ech}}(r, k) = \{t \mid \sum_{g \in R} \underline{\delta}^{r,g} d_t^g > 0, t \geq k\}$.

4. Multi-machine MLFP

In the *multi-machine* case, there are multiple processing units capable of performing the various tasks. These machines need not be identical, in that the set of production items, processing times, and production and setup costs can vary from machine to machine.

4.1. MLFP- m

We denote the multi-machine version of the MLFP as MLFP- m . Let M represent the set of machines, with N_m being the set of tasks available to be processed on machine m . While task production costs ($f_t^{i,m}$), setup costs ($g_t^{i,m}$), processing times ($\alpha_t^{i,m}$) and production capacities ($C_t^{i,m}$) may vary from one machine to the next, the combination and amounts of task inputs ($\rho^{i,r}$) and outputs ($\sigma^{i,r}$) are independent of which machine performs that task. Production-related variables $x_t^{i,m}$ and $y_t^{i,m}$ are defined as before, but now include information on which machine each task is performed.

The MILP model for the MLFP-m can be formulated as follows:

$$\begin{aligned} \min & \sum_{t \in T} \sum_{m \in M} \sum_{i \in N_m} f_t^{i,m} x_t^{i,m} + \sum_{t \in T} \sum_{m \in M} \sum_{i \in N_m} g_t^{i,m} y_t^{i,m} \\ & + \sum_{t \in T} \sum_{r \in R} h_t^r s_t^r, \end{aligned} \quad (32)$$

subject to

$$s_t^r = s_{t-1}^r + \sum_{m \in M} \sum_{i \in N_m} \sigma^{ir} x_t^{i,m} - \sum_{m \in M} \sum_{i \in N_m} \rho^{ir} x_t^{i,m} - d_t^r \quad \forall r \in R, t \in T, \quad (33)$$

$$\sum_{i \in N_m} \alpha_i^m x_t^{i,m} \leq C_{m,t} \quad \forall m \in M, t \in T, \quad (34)$$

$$x_t^{i,m} \leq \left\lfloor \frac{C_{m,t}}{\alpha_i^m} \right\rfloor y_t^{i,m} \quad \forall m \in M, i \in N_m, t \in T, \quad (35)$$

$$x_t^{i,m} \in \mathbb{Z}^+, y_t^{i,m} \in \{0, 1\} \quad \forall m \in M, i \in N_m, t \in T, \quad (36)$$

$$s_t^r \geq 0 \quad \forall r \in R, t \in T. \quad (37)$$

The interpretation of the objective function and constraints of the MLFP-m is analogous to the single-machine setting.

4.2. Valid inequalities

The following single-level inequalities are valid for the MLFP-m:

$$s_{k-1}^r \geq \sum_{t=k}^l \left[d_t^r \left(1 - \sum_{m \in M} \sum_{i \in O(r) \cap N_m} \sum_{u=k}^t y_u^{i,m} \right) \right] \quad \forall r \in R, k = 1, \dots, T, l = k, \dots, T. \quad (38)$$

Likewise, the following echelon inequalities are valid for the MLFP-m:

$$\begin{aligned} e_{k-1}^r + \sum_{t=k}^l \sum_{m \in M} \sum_{i \in O(r) \cap N_m} \min \left\{ \underline{D}_{t,l}^r, \left\lfloor \frac{C_{m,t}}{\alpha_i^m} \right\rfloor \sigma^{i,r} \right\} y_t^{i,m} \\ \geq \underline{D}_{k,l}^r \quad \forall r \in R, k = 1, \dots, T, l = k, \dots, T, \end{aligned} \quad (39)$$

where e_{k-1}^r and $\underline{D}_{k,l}^r$ are defined as in the single-machine case.

The proofs on inequalities (38) and (39) follow from the same arguments as the corresponding single-machine proofs in Theorems 1 and 3. As in the single-machine case, the single-level inequalities can be restricted to periods l with positive external demand for item r . Similarly, the echelon inequalities can be restricted to those with a positive lower-bound-on-echelon-demand for r in period l ($\underline{D}_{l,l}^r > 0$).

5. Computational results

We tested the performance of our echelon inequalities on a set of both fixed and flexible production lot sizing structures. Table 1 provides details about each problem instance, namely the number of items, the number of *flexible production items* (i.e., the number of items that may be produced by multiple tasks), the number of tasks, the number of levels (i.e., the largest number of consecutive tasks required to produce an end item), the number of machines, and the product structure (i.e., serial, assembly, or general network).

Table 1. Lot sizing problem instances

Problem	Items	Flexible items	Tasks	Levels	Machines	Product structure
1	5	0	4	4	{1,3}	Serial
2	10	0	8	4	{1,2}	Serial
3	12	0	9	3	{1,2}	Serial
4	12	0	6	3	{1,3}	Assembly
5	12	2	7	3	{1,3}	Assembly
6	12	2	8	3	{1,3}	Assembly
7	10	1	8	4	{1,2}	General
8	11	2	11	4	{1,2}	General
9	10	1	9	3	{1,2}	General
10	12	1	10	4	{1,2,3}	General
11	10	2	9	3	{1,2}	General
12	13	2	10	3	{1,3}	General
13	11	2	11	3	{1,2}	General
14	14	3	12	3	{1,2}	General
15	9	1	8	4	{1,2}	General
16	9	2	9	4	{1,2}	General
17	10	3	10	3	{1,3}	General
18	9	3	10	4	{1,3}	General
19	10	3	11	3	{1,3}	General
AG86	15	0	12	5	{1}	General

Problems were chosen with varying structural characteristics. *Serial* structures are defined such that each item serves as an input to at most one task and as an output from at most one task (fixed production), and each task has a single input. *Assembly* structures require that each item is input to at most one task, but relaxes the restrictions on task inputs and flexible production. Lastly, *general* structures allow for items to be input to multiple tasks. The general structure of problem AG86 is taken from Afentakis and Gavish (1986).

Each of these problem instances is tested over a 12-period time horizon. All other problem parameters were generated randomly (while ensuring moderate capacity utilization) as follows: Item/task inputs ($\rho^{i,r}$) and outputs ($\sigma^{i,r}$) were generated randomly in $\{1, 2, 3\}$, with varying fixed probabilities associated with each of these three quantities. Task production costs ($J_i^{i,m}$) were generated uniformly in $[100, 500]$, task setup costs ($g_i^{i,m}$) were generated uniformly in $[5000, 10\,000]$ and item holding costs were generated uniformly in $[1, 100]$. Task processing times ($\alpha_{i,m}$) were generated randomly in $\{1, 2, 3\}$ with uniform probability. Demand for end items was randomly generated over the interval $[0, 20]$ or $[0, 40]$, depending on product structure, in periods 2, ..., T . Finally, capacities were assigned according to the demand and processing times generated above, so that capacity utilization was approximately 70% for each problem. Note that determining exact capacity utilization *a priori* is impossible due to flexibility in the production sequences and the multi-machine nature of the problems, and so averaging item production over flexible production tasks and machines was necessary to calculate capacities. Capacity was constant over time periods, but varied over machines. Detailed problem instance data is available upon request from the authors.

We solved each problem instance, using CPLEX 8.1, in three ways. First, we solved the initial problem formulation, MLFP-m MIP, under the solver's default settings and with all standard cuts turned on. Second, we generated all single-level inequalities for that problem and added them to CPLEX's cutpool. The problem was then solved via a branch-and-cut approach using the single-level cuts and CPLEX's standard cuts. Third, we added the lower-bound-on-echelon-inventory variable (e_i^l) to the formulation of the MLFP. From there, we generated all echelon inequalities for that problem and added them to the cutpool. The problem was then solved via a branch-and-cut approach using the echelon cuts and CPLEX's standard cuts. In each case, CPLEX managed the single-level and/or echelon cuts in the cut pool using its default settings. The stopping criteria is either reaching optimality or a predetermined time cut-off (10 000 seconds). Imposing such a time limit is consistent with treatments elsewhere in the literature; it is perhaps also consistent with industry practice, where solutions must typically be obtained within a certain time window.

Table 2 shows the CPU time required to solve each problem by each of the three solution techniques. For problems that do not reach optimality within 10 000 seconds, the table also shows the corresponding *optimality gap*. To simplify notation, we denote the two-machine version of problem 3, for example, as problem 3-m2. The *CPU reduction vs MLFP MIP* column refers to the relative performance of the MLFP + echelon cuts (MLFP + single-level) solution method compared against that of the initial formulation, MLFP MIP. This ratio is calculated as

$$\frac{\text{CPU}_{\text{MLFP}} - \text{CPU}_{\text{MLFP+Ech}}}{\text{CPU}_{\text{MLFP}}} \times 100\%$$

for the echelon cuts case, where CPU_{MLFP} ($\text{CPU}_{\text{MLFP+Ech}}$) denotes the time required to solve to optimality via the original formulation of the MLFP (MLFP + echelon cuts). In the MLFP + single-level case, we replace $\text{CPU}_{\text{MLFP+Ech}}$ with $\text{CPU}_{\text{MLFP+SL}}$, where $\text{CPU}_{\text{MLFP+SL}}$ is the solution time associated with the MLFP + single-level solution method.

We can see in Table 2 that the addition of the echelon cuts to the initial formulation of the MLFP can result in significant reduction in solution times. Of the 26 problem instances for which an optimal solution was found by either the MLFP MIP or MLFP + echelon cuts solution method, only three (1-m3, 4-m3 and 5-m3) resulted in increased computational times by adding the echelon cuts, where 1-m3 and 5-m3 were very small problems, each solving in under 70 seconds via both methods. The third instance, 4-m3, resulted in a rather marginal computational increase of 13.6%. The remaining 23 problem instances experienced reduced CPU times as a result of the echelon cuts, many in the range of reduction of 90% and above. Also, note that the echelon cuts outperformed the single-level cuts in 20 out of 26 problem instances that were solved to optimality (with both solution methods performing essentially equally for 5-m1 and 5-m3). Furthermore, the single-level cuts resulted in increased computational time, compared with the MLFP MIP, in three cases (1-m1, 1-m3 and 5-m3), performing no better than the echelon cuts in this regard. Lastly, there were 13 problem instances for which no optimal solution was found by any of the three solution procedures after 10 000 seconds. In addition to these 13, there were six problem instances not solvable by MLFP MIP within the time limit, with only one additional instance (11-m1) not solvable using the echelon cuts after 10 000 seconds.

In Table 3 we consider the 13 problems for which no optimal solution was found by any of the three solution procedures within our time limit. To gain insight into which solution method is performing better in these cases, we examine the optimality gap for each method after our time limit is reached. Table 3 shows the objective value and optimality gap corresponding to the best integer solution found after 10 000 seconds, for each of the three solution methods.

Table 2. Reduction in computational time

Problem	MLFP MIP CPU time (seconds)	MLFP + single level cuts		MLFP + echelon cuts	
		CPU time (seconds)	CPU reduction vs MLFP MIP (%)	CPU time (seconds)	CPU reduction vs MLFP MIP (%)
1-m1	0.8	5.1	-537.5	0.5	37.5
1-m3	44.8	78.3	-74.8	60.4	-34.8
2-m1	*10 000	*10 000	—	*10 000	—
2-m2	9480.4	766.2	91.9	3051.9	67.8
3-m1	*10 000	1505.3	≥84.9	141.3	≥98.6
3-m2	*10 000	*10 000	—	*10 000	—
4-m1	1002.8	736.4	26.6	163.0	83.7
4-m3	537.1	305.1	43.9	610.1	-13.6
5-m1	1.3	0.7	46.2	0.9	30.7
5-m3	51.7	68.3	-32.1	69.7	-34.8
6-m1	64.6	40.4	37.5	21.5	66.7
6-m3	1709.5	205.9	87.9	197.1	88.5
7-m1	1562.3	1477.1	5.5	68.2	93.7
7-m2	*10 000	*10 000	—	*10 000	—
8-m1	492.0	189.7	61.4	14.8	97.0
8-m2	*10 000	*10 000	—	*10 000	—
9-m1	*10 000	*10 000	—	*10 000	—
9-m2	*10 000	*10 000	—	*10 000	—
10-m1	46.3	2.9	93.7	1.0	97.8
10-m2	6167.5	145.4	97.6	16.6	99.7
10-m3	*10 000	3590.7	≥64.1	101.1	≥99.0
11-m1	*10 000	826.0	≥91.7	*10 000	—
11-m2	*10 000	7244.7	≥27.5	4321.1	≥56.8
12-m1	*10 000	*10 000	—	*10 000	—
12-m3	*10 000	*10 000	—	*10 000	—
13-m1	*10 000	1038.3	≥89.6	52.4	≥99.5
13-m2	*10 000	*10 000	—	*10 000	—
14-m1	24.2	11.1	54.1	2.6	89.2
14-m2	4154.8	573.4	86.2	249.5	94.0
15-m1	6617.3	859.5	87.0	617.0	90.7
15-m2	*10 000	*10 000	—	*10 000	—
16-m1	196.7	42.4	78.4	28.0	85.7
16-m2	3727.1	1326.1	64.4	3619.0	2.9
17-m1	3.06	0.56	81.7	0.51	83.3
17-m3	1215.1	88.2	92.7	7.8	99.4
18-m1	682.6	251.3	63.2	393.0	42.4
18-m3	*10 000	*10 000	—	*10 000	—
19-m1	*10 000	875.2	91.2	262.2	97.4
19-m3	*10 000	*10 000	—	*10 000	—
AG86-m1	*10 000	*10 000	—	*10 000	—

*Problem not solved to optimality within 10 000 seconds.

The optimality gap for each solution method, as calculated by CPLEX, is defined by the ratio

$$\frac{Z_{INT} - Z_{LB}}{Z_{INT}} \times 100\%,$$

where Z_{INT} (Z_{LB}) refers to the objective value of the best integer solution (lower bound) found after 10 000 seconds. The optimality gap is a good measure of *how close* a solution method is to solving a problem to optimality.

Note that the optimality gap corresponding to the MLFP + echelon cuts method is smaller than the gap corresponding to the MLFP MIP method in all 13 instances, ranging from moderate (18-m3: 0.64%) to significant (7-m2: 8.81%) reductions in the optimality gap. Furthermore, note that the echelon cuts outperform the single-level cuts, with regard to lower optimality gaps, in 12 out of 13 problems instances, performing slightly worse for problem 19-m3 (0.51%). On the other hand, moderate (12-m3: 0.63%) to significant (9-m1: 5.50%) reductions in the optimality gap result by

Table 3. Optimality gaps

Problem	MLFP MIP		MLFP + single level cuts		MLFP + echelon cuts	
	Best integer solution	Optimality gap (%)	Best integer solution	Optimality gap (%)	Best integer solution	Optimality gap (%)
2-m1	348 069	6.53	346 112	5.44	346 969	4.36
3-m2	399 678	8.38	400 305	5.97	398 718	4.28
7-m2	325 939	10.08	314 075	3.73	313 256	1.27
8-m2	344 854	11.53	345 465	9.00	343 541	6.24
9-m1	373 831	12.14	378 171	8.90	367 002	3.40
9-m2	350 896	12.21	335 828	6.85	337 168	5.27
12-m1	566 084	5.94	566 115	5.35	558 852	0.87
12-m3	497 039	9.87	492 621	7.42	494 523	6.79
13-m2	655 072	11.31	646 490	8.31	640 386	4.34
15-m2	412 167	5.43	410 731	3.11	409 334	2.21
18-m3	951 247	6.77	952 081	6.79	943 168	6.13
19-m3	451 241	7.18	450 685	5.05	452 742	5.56
AG86-m1	1115 806	9.76	1119 049	6.56	1100 485	1.36

Best integer solution and **smallest optimality gap** amongst the three solution approaches.

abandoning the single-level cuts in favor of the echelon cuts in the other 12 cases. Also, note that while the MLFP + single-level method outperforms the MLFP + echelon cuts method with respect to the best integer solution found in 10 000 seconds in four of 13 problem instances, the ability of the MLFP + echelon cuts method to consistently provide stronger *lower bound* solutions results in lower optimality gaps for three of these four instances, which in turn suggests faster optimal solution times for the MLFP + echelon cuts method. Finally, regarding the best integer solution obtained, note that the MLFP MIP method outperforms the single-level cuts in six of 13 problem instances. However, there is only one instance (19-m3) in which the MLFP MIP outperforms the echelon cuts method.

Next, we examine a specific problem, problem 12-m1, and show how various upper and lower bounds on the optimal cost behave as CPU run time increases. In Fig. 2, we display the cost of the best integer solution and the corresponding lower bound as a function of CPU run time for each of the three solution approaches, MLFP MIP, MLFP + single level cuts and MLFP + echelon cuts. While Table 3 identifies the MLFP + echelon cuts method as providing the best integer solution and tightest optimality gap after 10 000 seconds for this problem, it is also interesting to note the relative performance of the three solution approaches at earlier points in time in the optimization. For 12-m1, the MLFP + echelon cuts method yields the best integer solution in more than 98% of the 10 000 second run time, the best lower bound in 100% of the run time and the tightest optimality gap in 100% of the run time as well. Similar results were observed for other problems; we do not enclose those here for the sake of brevity.

Because the echelon inequality coefficients depend directly on capacity levels, we investigated the impact of capacity on the performance of the echelon cuts. Note that

while increased utilization (decreased capacity) typically results in larger solution times, the echelon inequalities can actually become stronger by decreasing capacity, because of how the y -variable coefficients are derived. We performed a sensitivity analysis with respect to varying capacity levels on four problem instances. We tested the effect of varying capacity levels on the solution times for the MLFP and MLFP + echelon cuts solution methods. The four test problems were chosen because they solved to optimality (within 10 000 seconds) by both solution approaches and had initially performed well via the echelon cuts solution approach (our objective is to examine the performance sensitivity of this approach to capacity utilization). Capacity utilization for each problem instance was varied by scaling capacity to result in utilization levels of 50, 60, 70, 80 and

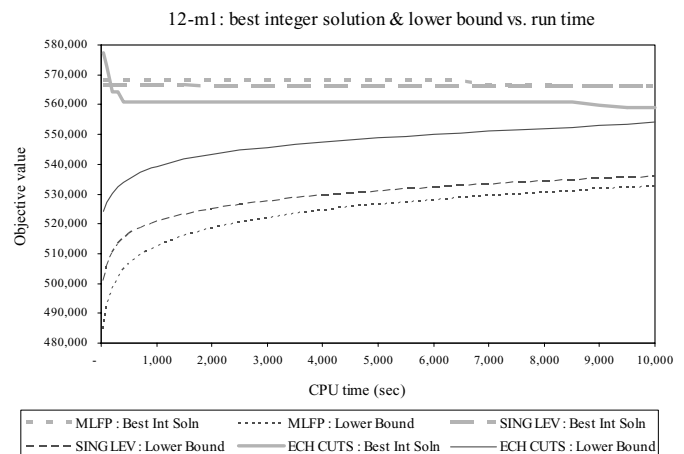


Fig. 2. Problem 12-m1 best integer solution and lower bound as a function of run time.

Table 4. Capacity sensitivity analysis

Problem	Capacity utilization (%)	MLFP MIP		MLFP + echelon cuts		Percentage of reduction in CPU time (%)
		CPU time (seconds)	Number of nodes	CPU time (seconds)	Number of nodes	
8-m1	50	8.35	11 848	1.70	1210	79.6
	60	303.68	422 016	67.48	63 707	77.8
	70	492.0	782 835	14.8	11 043	97.0
	80	2 154.64	2752 393	523.73	378 824	75.7
	90	*10 000	13 944 067	*10 000	7 053 619	—
10-m2	50	3897.41	4264 417	3.42	1474	99.9
	60	6943.74	7897 354	12.38	7492	99.8
	70	6167.5	5469 795	16.6	9819	99.7
	80	*10 000	8927 623	120.01	63 933	≥98.8
	90	*10 000	8802 654	228.86	96 475	≥97.7
14-m1	50	126.92	190 018	2.37	1505	98.1
	60	334.54	429 988	2.82	1866	99.2
	70	24.2	28 162	2.6	1579	89.2
	80	373.58	404 439	21.29	11 830	94.3
	90	*10 000	11 559 439	1625.68	854 451	≥83.7
16-m1	50	10.46	14 688	2.95	2413	71.8
	60	8.77	11 882	4.35	3912	50.4
	70	196.70	268 409	28.02	23 587	85.7
	80	201.51	265 969	114.34	101 856	43.3
	90	4415.55	5092 264	118.98	105 912	97.3

Table 5. Time horizon sensitivity analysis

Problem	Number of periods	MLFP MIP		MLFP + echelon cuts		Percentage of reduction in CPU time (%)
		CPU time (seconds)	Optimality gap (%)	CPU time (seconds)	Optimality gap (%)	
8-m1	8	128.7	—	2.3	—	98.2
	12	492.0	—	14.8	—	97.0
	16	*20 000	5.60	2886.4	—	≥85.6
	20	*20 000	12.87	*20 000	9.61	—
	24	*20 000	15.27	*20 000	8.01	—
10-m2	8	640.4	—	6.6	—	99.0
	12	6167.5	—	16.6	—	99.7
	16	*20 000	3.69	40.8	—	≥99.8
	20	*20 000	13.08	1307.5	—	≥93.5
	24	*20 000	15.49	9156.2	—	≥54.2
14-m1	8	1.4	—	0.3	—	78.6
	12	24.2	—	2.6	—	89.2
	16	6841.6	—	33.7	—	99.5
	20	*20 000	3.22	686.4	—	≥96.6
	24	*20 000	6.97	1548.0	—	≥92.2
16-m1	8	0.9	—	0.7	—	22.2
	12	196.70	—	28.02	—	85.7
	16	10 856.9	—	698.3	—	93.6
	20	*20 000	2.24	*20 000	2.03	—
	24	*20 000	7.67	*20 000	6.30	—

*Problem not solved to optimality within 20 000 seconds.

90% (capacity was set in previous experiments to result in approximately 70% utilization).

Table 4 displays the results for each of the four test problems over each of the five utilization levels. The MLFP + echelon cuts method significantly outperforms the MLFP MIP method in all instances for which an optimal solution was found. The percentage of reduction in CPU time remains consistently strong for all test problems, regardless of how high or low the utilization levels are set. Furthermore, the reduction percentages remain surprisingly steady for all test problems except 16-m1, which is the smallest of the five instances and had relatively short computational times in all cases. Note also that solution time reductions in the range of 90% and above are very common, with each of the four test problems experiencing such a reduction for at least one of the utilization levels. And so, the MLFP + echelon cuts method drastically outperforms the MLFP MIP method under varying capacity utilization levels, providing significant, and often steady, percentage reductions in CPU time.

In addition, because each time interval $[k, l] \subseteq [1, T]$ corresponds to a unique echelon inequality, we performed a sensitivity analysis on the length of our time horizon, T . Using the same four problems as in the capacity sensitivity analysis, we tested the effect of varying time horizons on the solution times for the MLFP MIP and MLFP + echelon cuts solution methods. We ran each problem over an 8, 12, 16, 20 and 24-period time horizon, where production capacity was fixed over all periods and external item demand was generated randomly to ensure approximately 70% capacity utilization. We extended our imposed time limit to 20 000 seconds for this analysis.

Table 5 displays the results for each of the four test problems over each of the five time horizons. Optimality gaps are given for tests which did not reach optimality within the time limit of 20 000 seconds. The MLFP + echelon cuts method outperforms the MLFP MIP method in all 16 instances for which an optimal solution was found, with typical solution time reductions in the range of 90% and above. Again, the reduction percentages remained both strong and steady over the varying time horizon lengths, and the actual reduction may be even stronger than the bounds indicated for those instances which were not solved to optimality. The MLFP + echelon cuts solution method also results in smaller optimality gaps than the MLFP MIP method in the four instances for which no optimal solution was found. Therefore, the MLFP + echelon cuts method consistently outperforms the MLFP MIP method under varying time horizon lengths, resulting in significant solution time and optimality gap reductions.

6. Conclusions and future extensions

We considered a multi-level lot sizing problem with flexible production sequences (the MLFP), and formulated it

as a MILP. We derived valid echelon inequalities for our problem based on the notion of lower-bound-on-echelon-inventory, and extended our results to the multi-machine case. Our computational results showed that the echelon inequalities can significantly reduce solution times and improve solution quality. This work provides new tools for manufacturing settings with flexible production sequences. Previous methodology for multi-level production settings has not considered this characteristic, focusing primarily on systems with fixed sequences.

There are several possible extensions to our research. For instance, it is likely that an appropriate separation algorithm, used to identify and generate only the *best* echelon cuts, would limit the number of cuts generated, thus reducing LP-bounding times and overall solution times. Recall that the echelon cuts outperformed the single-level cuts in 20 out of 26 problem instances with regard to optimal solution times. While this is encouraging, the fact that the single level cuts, a *subset* of the set of echelon inequalities, actually outperform the echelon cuts in six out of 26 instances, suggests that too many unnecessary echelon cuts are being generated, thus slowing down solution times. Another interesting extension would be to investigate conditions under which the echelon cuts are facet defining. Begnaud (2006) showed that a similar set of echelon-based inequalities are facet-defining for systems with fixed production and no capacity constraints. It would also be interesting to extend the approach proposed here to incorporate tasks with multiple outputs. The difficulty here is that the notion of echelon inventory breaks down (given the current MLFP formulation). Additional parameters would be needed to determine how input items, or what fraction of input items, are allocated to multiple output items. Other extensions include problems with small buckets, backorders or sequence-dependent changeover costs.

Acknowledgements

We would like to thank two anonymous referees and an associate editor for their suggestions, which greatly improved the readability and relevance of the paper. In particular, we are deeply grateful to the referee who suggested a much clearer proof of the main result using single-item relaxations.

References

- Afentakis, P. (1987) A parallel heuristic for lot-sizing in multistage production systems. *IIE Transactions*, **19**, 34–42.
- Afentakis, P. and Gavish, B. (1986) Optimal lot-sizing algorithms for complex product structures. *Operations Research*, **34**(2), 237–249.
- Afentakis, P., Gavish, B. and Karmarkar, U. (1984) Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Operations Research*, **30**(2), 222–239.

- Barany, I., Roy, T.J.V. and Wolsey, L.A. (1984) Strong formulations for multi-item capacitated lot sizing. *Management Science*, **30**(10), 1255–1261.
- Begnaud, J. (2006) The multilevel lot sizing problem with flexible production sequences. Master's thesis, University of Minnesota, Minneapolis, MN 55455, USA.
- Belvaux, G. and Wolsey, L.A. (2000) bc-prod: a specialized branch-and-cut system for lot-sizing problems. *Management Science*, **46**(5), 724–738.
- Belvaux, G. and Wolsey, L.A. (2001) Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, **47**(7), 993–1007.
- Clark, A.J. and Scarf, H. (1960) Optimal policies for a multi-echelon inventory problem. *Management Science*, **6**(4), 475–490.
- Constantino, M. (1996) A cutting plane approach to capacitated lot-sizing with start-up cost. *Mathematical Programming*, **75**, 353–376.
- Gaglioppa, F., Miller, L.A. and Benjaafar, S. (2008) Multitask and multistage production planning and scheduling for process industries. *Operations Research*, **56**(4), 1010–1025.
- Harrison, T.P. and Lewis, H.S. (1996) Lot sizing in serial assembly systems with multiple constrained resources. *Management Science*, **42**(1), 19–36.
- Katok, E., Lewis, H.S. and Harrison, T.P. (1998) Lot-sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. *Management Science*, **44**(6), 859–877.
- Miller, A.J., Nemhauser, G.L. and Savelsbergh, M.W.P. (2003a) A multi-item production planning model with setup times: algorithms, reformulations, and polyhedral characterizations for a special case. *Mathematical Programming*, **95**(1), 71–90.
- Miller, A.J., Nemhauser, G.L. and Savelsbergh, M.W.P. (2003b) On the polyhedral structure of a multi-item production planning model. *Mathematical Programming*, **94**(2/3), 375–405.
- Miller, A.J. and Wolsey, L.A. (2003) Tight MIP formulations for multi-item discrete lot sizing problems. *Operations Research*, **51**(4), 557–605.
- Pochet, Y. and Wolsey, L.A. (1991) Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, **37**(1), 53–67.
- Pochet, Y. and Wolsey, L.A. (1993) Lot-sizing with constant batches: formulation and valid inequalities. *Management Science*, **18**(4), 767–785.
- Pochet, Y. and Wolsey, L.A. (2006) *Production Planning by Mixed Integer Programming*. Springer, New York, NY.
- Simpson, N. and Erenguc, S. (1998) Improved heuristic methods for multiple stage production planning. *Computers & Operations Research*, **25**(7/8), 611–623.
- Stadtler, H. (2003) Multilevel lot-sizing with setup times and multiple constrained resources: internally rolling schedules with lot-sizing windows. *Operations Research*, **51**, 487–502.
- Tempelmeier, H. and Derstroff, M. (1996) A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. *Management Science*, **42**(5), 738–757.
- Wolsey, L.A. (2002) Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation. *Management Science*, **48**(12), 1587–1602.

Biographies

Joe Begnaud is a supply chain project leader at 3M Company, where his emphasis is in the area of supply chain optimization modeling. He holds an MS from the University of Minnesota-Twin Cities, and a BA from Saint John's University (Collegeville, MN). He completed this research while working toward his MS degree.

Saif Benjaafar is a Professor of Industrial & Systems Engineering at the University of Minnesota where he is also Director of the Industrial & Systems Engineering Program, Director of the Center for Supply Chain Research and a Faculty Scholar with the Center for Transportation Studies. He holds Ph.D. and MS degrees from Purdue University and a BS degree from the University of Texas at Austin. His research is in the areas of manufacturing and service operations, production and inventory systems, and supply chain management. He serves on the editorial board of several journals including *MSOM*, *IIE Transactions*, *NRL* and *POM*. His papers have been published in various journals including *Management Science*, *Operations Research*, *MSOM* and *IIE Transactions*.

Lisa A. Miller is a Lead Optimization Analyst at Target Corporation. She holds Ph.D. and BS degrees from the Georgia Institute of Technology. Her research interests are in optimization and its applications to production systems and retail planning. She completed this work while an Assistant Professor at the University of Minnesota.

Copyright of IIE Transactions is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.