INFORMS Journal on Computing

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

Lot Sizing with Piecewise Concave Production Costs

Esra Koca, Hande Yaman, M. Selim Aktürk

To cite this article:
Esra Koca, Hande Yaman, M. Selim Aktürk (2014) Lot Sizing with Piecewise Concave Production Costs. INFORMS Journal on Computing 26(4):767-779. https://doi.org/10.1287/ijoc.2014.0597

INFORMS Journal on Computing
Vol. 26, No. 4, Fall 2014, pp. 767–779
ISSN 1091-9856 (print) | ISSN 1526-5528 (online)

http://dx.doi.org/10.1287/ijoc.2014.0597

Lot Sizing with Piecewise Concave Production Costs

Esra Koca, Hande Yaman, M. Selim Aktürk
Department of Industrial Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey
{ekoca@bilkent.edu.tr, hyaman@bilkent.edu.tr, akturk@bilkent.edu.tr}

We study the lot-sizing problem with piecewise concave production costs and concave holding costs. This problem is a generalization of the lot-sizing problem with quantity discounts, minimum order quantities, capacities, overloading, subcontracting or a combination of these. We develop a dynamic programming algorithm to solve this problem and answer an open question in the literature: we show that the problem is polynomially solvable when the breakpoints of the production cost function are time invariant and the number of breakpoints is fixed. For the special cases with capacities and subcontracting, the time complexity of our algorithm is as good as the complexity of algorithms available in the literature. We report the results of a computational experiment where the dynamic programming is able to solve instances that are hard for a mixed-integer programming solver. We enhance the mixed-integer programming formulation with valid inequalities based on mixing sets and use a cut-and-branch algorithm to compute better bounds. We propose a state space reduction–based heuristic algorithm for large instances and show that the solutions are of good quality by comparing them with the bounds obtained from the cut-and-branch.

Keywords: lot sizing; piecewise concave production cost; quantity discounts; subcontracting; dynamic programming
History: Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms; received February 2013; revised August 2013, January 2014; accepted January 2014. Published online in *Articles in Advance* May 20, 2014.

1. Introduction

Lot-sizing problems arise in production, procurement, and transportation systems under different cost and capacity settings. Given a planning horizon, demand, production (or procurement/shipment), and inventory holding costs, the aim of the lot-sizing problem is to propose a minimum cost production plan to satisfy the demand (see, e.g., the seminal works by Wagner and Whitin 1958 and Zangwill 1966 and the book by Pochet and Wolsey 2006). In this paper, we study the lot-sizing problem in which the inventory holding cost function is concave and the production cost function is a piecewise concave function. We call this problem the "lot-sizing problem with piecewise concave production costs" and abbreviate it with LS-PC.

Zangwill (1967) studies piecewise concave functions. A function p is piecewise concave with breakpoints at $b^0 < b^1 < \cdots < b^m$ if p is concave in each of the m intervals $[b^{j-1}, b^j]$ for $j = 1, \ldots, m$. Note that concavity of p in each of the intervals implies that it is lower semicontinuous.

Examples of piecewise concave production costs are depicted in Figures 1 and 2. In Figure 1, the first two functions represent common quantity discounts known as incremental discount and all units discount. Federgruen and Lee (1990) study the lot-sizing problem with these two types of discounts. They assume that the production cost function has two pieces and propose dynamic programming algorithms of complexity $O(n^3)$ and $O(n^2)$ for the problems with all units discount and incremental discount, respectively, where n is the number of periods. Chan et al. (2002) consider the modified all units discount depicted in Figure 1(c). They prove that the lot-sizing problem with this cost structure is NP-hard when either the production cost functions vary from period to period or the number of breakpoints is not bounded by a constant. Li et al. (2012) study the lot-sizing problem with all units discount and resales under the assumptions that the breakpoints of the cost function are time invariant, the number of breakpoints is fixed, and there is no capacity constraint. They develop an $O(n^{m+3})$ time algorithm to solve this problem, where m is the number of breakpoints. Archetti et al. (2011) present polynomial time algorithms to solve special cases of the lot-sizing problem with modified all units discount and incremental discount when the cost functions are time invariant.

Atamtürk and Hochbaum (2001) study the lot-sizing problem with subcontracting where the production and subcontracting costs are concave nondecreasing functions and the inventory holding cost is a linear function. The overall production cost function is depicted in Figure 1(d): The first piece of the function corresponds to regular production and the second piece corresponds to subcontracting or overloading. The authors develop an $O(n^5)$ time dynamic programming (DP) algorithm for the case where the regular production capacities (the breakpoint of the cost function) are the same for all periods.

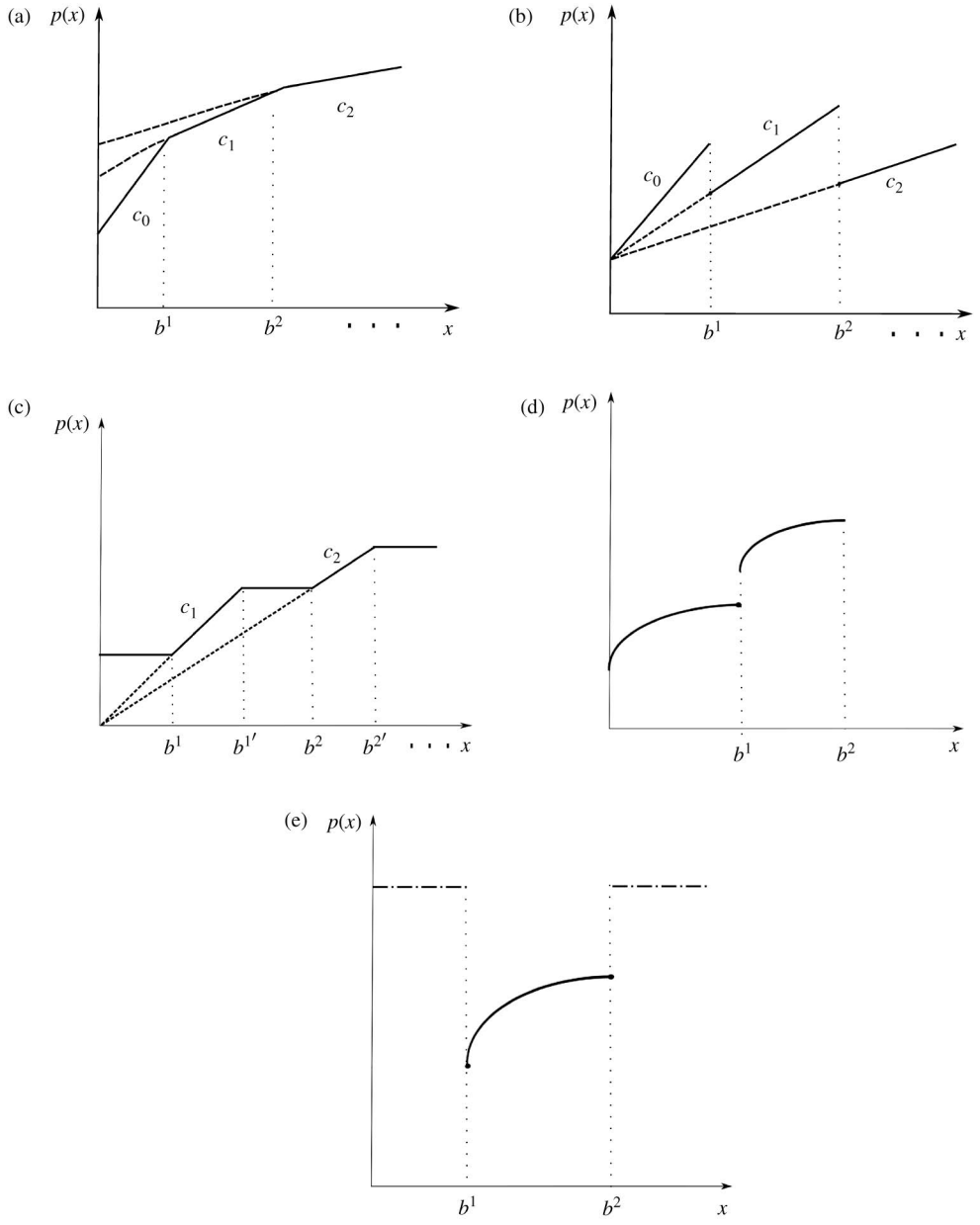

Figure 1 Some Special Cases of Piecewise Concave Functions

The production cost function given in Figure 1(e) models constraints on minimum production (order) quantities, as studied by Hellion et al. (2012). In this setting, if there is a production at a given period, then the production amount should not be less than a minimum level b^1 and should not exceed the capacity b^2. The authors assume that the production and inventory holding cost functions are concave and propose a DP algorithm for this problem. The time complexity reported in Hellion et al. (2012) was corrected and reported as $O(n^6)$ (Hellion et al. 2013). A special case of this problem in which production and inventory holding costs are linear is studied by Okhrin and Richter (2011). They assume that there is no setup cost and unit production and inventory holding costs are constant over the planning horizon. They develop a polynomial time algorithm to solve this problem.

As seen, piecewise concave functions can be used to represent discounts, subcontracting, capacity acquisition, and overloading, as well as minimum quantity requirements and capacities. In addition, one can represent any combination of these using piecewise concave functions. In Figure 2(a), we model a setting with discounts and overloading. The unit cost, c_0, up to the first breakpoint b^1 can be viewed as the regular unit purchasing cost. Then a quantity discount applies and the unit cost becomes $c_1 < c_0$ up to the second breakpoint b^2, which is the capacity of the supplier. Thereafter, the supplier requires use of overtime (or subcontracting) to fulfill the additional orders, so the unit cost is $c_2 > c_0$.

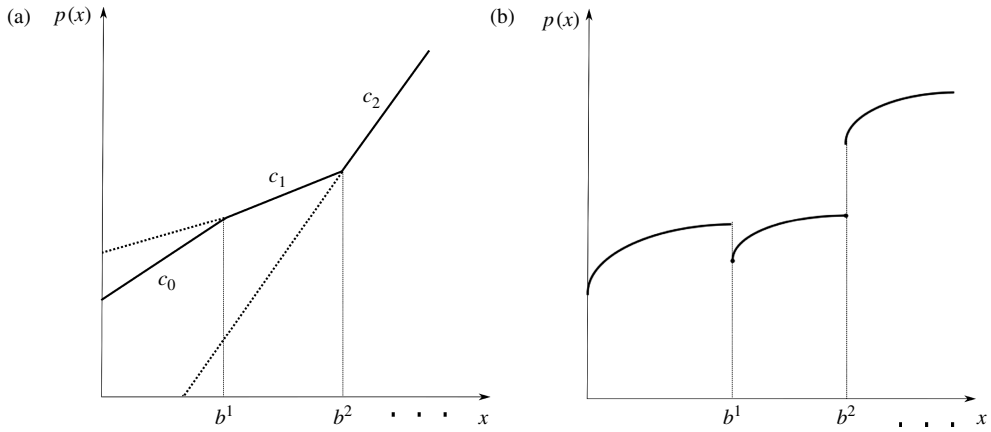

Figure 2 Examples of Piecewise Concave Functions

Note that the resulting cost function is neither convex nor concave.

Now consider the case where several suppliers give offers (possibly with discounts) for a product and the company purchases its products from at most one supplier in each period. Then the production cost is the minimum of the purchasing costs over all suppliers and is a piecewise concave function if the cost function of each supplier is concave. An example is given in Figure 2(b), in which each segment of the cost function represents a supplier. The second supplier offers the most attractive price but has a lower bound for procurement, b^1 units, and a capacity of b^2 units. It is more beneficial to buy from the first supplier up to b^1 units and from the third supplier after b^2 units. Accordingly, decisions on the purchasing amounts in each period will also determine the supplier of each period. Therefore, this problem can be seen as a supplier selection and lot-sizing problem.

As the lot-sizing problem with modified all units discount studied by Chan et al. (2002) is a special case of LS-PC, LS-PC is NP-hard unless the breakpoints are time invariant and the number of breakpoints is bounded above by a constant.

Swoveland (1975) presents characteristics of an optimal solution when inventory holding and production cost functions are piecewise concave functions. He proposes a pseudopolynomial DP algorithm to solve this problem. Shaw and Wagelmans (1998) present an algorithm for the capacitated lot-sizing problem with piecewise linear production costs (not necessarily convex or concave) and general inventory holding costs. Their algorithm is also pseudopolynomial. VanHoesel and Wagelmans (1996) show that if the production cost function is piecewise concave and monotone and the number of pieces is polynomially bounded in the size of the problem, then there exists a fully polynomial approximation scheme.

The special cases of LS-PC with cost functions depicted in Figure 1 are polynomially solvable. However, to the best of our knowledge, there is no polynomial time algorithm to solve the problem with cost functions like those in Figure 2. Indeed, the complexity of the problem is open for the case where the number of breakpoints is fixed and the breakpoints are time invariant. Li et al. (2012) showed that the lot-sizing problem with modified all units cost is polynomially solvable under these assumptions. In this study, we prove that the more general problem, LS-PC, can be solved in polynomial time with a DP algorithm. This algorithm generalizes the algorithm of Florian and Klein (1971) for the constant capacity lot-sizing problem, which corresponds to the special case with one breakpoint. For the special cases with regular production and subcontracting, minimum production quantities, and constant capacities, our DP has the same time complexity as the one of Atamtürk and Hochbaum (2001) and Hellion et al. (2013), respectively. We also conduct a computational study to see if the DP is useful in practice. We derive a mixed-integer programming (MIP) formulation and solve it with an off-the-shelf solver. Our results show that the DP outperforms the MIP approach for some instances even when we strengthen the formulation with valid inequalities. For larger instances, we propose a heuristic method based on state space reduction. Our computational experiments show that the heuristic provides good quality solutions in reasonable computation times when the solver and the exact DP fail.

The rest of the paper is organized as follows. In §2, we formally define the problem LS-PC and state some important properties of an optimal solution to the problem. In §3, we present a polynomial time DP algorithm for solving the problem when the number of breakpoints is fixed and the breakpoints are time invariant; we show that the complexity of the DP is as good as the complexity of algorithms available in the literature for some special cases of the problem. We then report our computational experiments in §4 and propose a state space reduction based heuristic

algorithm for large instances in §5. Finally, in §6 we present some concluding remarks.

2. Problem Definition and Properties of Optimal Solutions

In the lot-sizing problem, we would like to find a minimum cost production plan over a planning horizon of n periods. The demand d_t, the production cost function p_t, and the inventory holding cost function h_t are given for each period t. Let x_t be the amount produced in period t and s_t be the stock on hand at the end of period t. Using these variables, the lot-sizing problem can be modeled as

$$\min \quad \left\{\sum_{t=1}^{n} p_t(x_t) + \sum_{t=0}^{n} h_t(s_t)\right\} \tag{1}$$

$$\text{s.t.} \quad s_{t-1} + x_t = d_t + s_t \quad t = 1, \ldots, n, \tag{2}$$

$$s_0 = 0, \tag{3}$$

$$s, x \geq 0. \tag{4}$$

Constraints (2) are inventory balance constraints. The assumption on the initial inventory being zero is imposed by constraint (3) and is made without loss of generality. Constraints (4) are variable restrictions. The objective function (1) is the sum of production and inventory holding costs.

In LS-PC, the inventory holding cost function $h_t(\cdot)$ is a concave function on $[0, \infty)$ and $p_t(\cdot)$ is a piecewise concave function on $[0, \infty)$ with m_t finite breakpoints $b_t^1, \ldots, b_t^{m_t}$ such that $b_t^0 = 0$ and $b_t^{i-1} < b_t^i$ for $i = 1, \ldots, m_t$.

As is typically done in the lot-sizing literature (see Pochet and Wolsey 2006), we will use the concepts of regeneration intervals and fractional periods to analyze the structure of optimal solutions. An interval $[j, l]$ with $1 \leq j \leq l \leq n$, $s_{j-1} = s_l = 0$ and $s_t > 0$ for $j \leq t < l$ is referred to as a *regeneration interval*, and a period i whose production level is not equal to any of the breakpoints of the production cost function, i.e., $x_i \in [b_i^0, \infty) \setminus \{b_i^0, \ldots, b_i^{m_i}\}$ is referred to as a *fractional period*. We define $b_i^{m_i+1} = \infty$ for all i.

If the production cost function is not monotone (see Figures 1(e) and 2(b)), we may have a positive ending inventory in all optimal solutions. Therefore, contrary to the case with the classical lot-sizing problems, we cannot say that there exists an optimal solution that is composed of a series of successive regeneration intervals. However, for our problem, there exists an optimal solution that is composed of a series of regeneration intervals that cover the interval $[1, j-1]$ plus an interval $[j, n]$ for some $1 \leq j \leq n+1$. We know the following properties for these intervals.

Theorem 1 (Swoveland 1975). *There exists an optimal solution to the problem LS-PC such that in each regeneration interval $[j, l]$ there exists at most one fractional period.*

Theorem 1 is a generalization of the "fractional period property" for the capacitated lot-sizing problem. Note that if $x_i > b_i^{m_i}$, then period i is a fractional period.

Theorem 2. *Suppose that the ending inventory is positive in all optimal solutions. Then there exists an optimal solution to the problem in which the last interval $[j, n]$ with $s_{j-1} = 0$ and $s_t > 0$ for $j \leq t \leq n$ does not contain any fractional periods. In other words, there exists an optimal solution to the problem that is composed of a series of regeneration intervals that cover the interval $[1, j-1]$ plus an interval $[j, n]$ for some $1 \leq j \leq n$ with no fractional period in the last interval $[j, n]$.*

Proof. Suppose that at all optimal solutions we have $s_n > 0$. Let (x, s) be an optimal solution with the largest j value such that $s_{j-1} = 0$ and $s_t > 0$ for $t = j, \ldots, n$. Suppose there exists a fractional period with $i \in [j, n]$ such that $b_i^k < x_i < b_i^{k+1}$ for some $k \in \{0, \ldots, m_i\}$. Define $\alpha = \min\{\min_{t=i}^{n} s_t, x_i - b_i^k\}$ and $\beta = b_i^{k+1} - x_i$ if b_i^{k+1} is finite and $\beta = \alpha$ otherwise. Clearly, α and β are positive. Now consider the two solutions (x^1, s^1) and (x^2, s^2) that are the same as (x, s) except that $x_i^1 = x_i - \alpha$, $s_t^1 = s_t - \alpha$ for $t = i, \ldots, n$, $x_i^2 = x_i + \beta$, and $s_t^2 = s_t + \beta$ for $t = i, \ldots, n$. Both solutions are feasible. Optimality of (x, s) implies that $p_i(x_i - \alpha) + \sum_{t=i}^{n} h_t(s_t - \alpha) - p_i(x_i) - \sum_{t=i}^{n} h_t(s_t) \geq 0$ and $p_i(x_i + \beta) + \sum_{t=i}^{n} h_t(s_t + \beta) - p_i(x_i) - \sum_{t=i}^{n} h_t(s_t) \geq 0$. Since p_i is concave on $[b_i^k, b_i^{k+1}]$ and h_t is concave on $[0, \infty)$ for each $t = i, \ldots, n$, we also have $(\beta/(\alpha+\beta))p_i(x_i - \alpha) + (\alpha/(\alpha+\beta))p_i(x_i + \beta) \leq p_i(x_i)$ and $(\beta/(\alpha+\beta))h_t(s_t - \alpha) + (\alpha/(\alpha+\beta))h_t(s_t + \beta) \leq h_t(s_t)$ for $t = i, \ldots, n$. Therefore, both (x^1, s^1) and (x^2, s^2) are also optimal. Either b_i^{k+1} is finite and (x^2, s^2) is an optimal solution where the fractional period i is eliminated or $k = m_i$ and, as (x, s) is an optimal solution with the largest j value such that $s_{j-1} = 0$ (implying that $s_t^1 > 0$ for $t = i, \ldots, n$), (x^1, s^1) is an optimal solution in which i is not a fractional period anymore. □

Remark 1. If we assume that the inventory holding cost function, h_t, is also piecewise concave with q_t finite breakpoints $r_t^1, \ldots, r_t^{q_t}$ such that $r_t^i < r_t^{i+1}$ for $i = 1, \ldots, q_t - 1$ and $t = 1, \ldots, n$, then with small modifications Theorems 1 and 2 still remain valid. In this case, an interval $[j, l]$ with $1 \leq j \leq l \leq n$ is called a *regeneration interval* if $s_{j-1} \in \{r_{j-1}^1, \ldots, r_{j-1}^{q_{j-1}}\}$, $s_l \in \{r_l^1, \ldots, r_l^{q_l}\}$, and $s_t \notin \{r_t^1, \ldots, r_t^{q_t}\}$ for $j \leq t < l$ (Swoveland 1975). Theorem 1 still holds true for this definition (Swoveland 1975). However, we need to restate Theorem 2 as the following:

Suppose that the ending inventory is not at a breakpoint level of the inventory holding cost; i.e., $s_n \notin \{r_n^1, \ldots, r_n^{q_n}\}$, in all optimal solutions. Then there exists an optimal solution to the problem in which the last interval $[j, n]$ with $s_{j-1} \in \{r_{j-1}^1, \ldots, r_{j-1}^{q_{j-1}}\}$ and $s_t \notin \{r_t^1, \ldots, r_t^{q_t}\}$ for $j \leq t \leq n$ does not contain any fractional periods.

From Theorem 1, similar to that done in the classical lot-sizing problems, we can find the minimum cost solution for each regeneration interval $[j, l]$ by assuming that it consists of at most one fractional period. However, it is not sufficient for finding a minimum cost solution for the problem, since for the intervals $[j, n]$ we need to consider the case where it is not a regeneration interval. In this case, for the intervals $[j, n]$, from Theorem 2, we can search for a minimum cost solution by assuming that it does not contain any fractional period. Consequently, we can find a minimum cost solution for each interval $[j, n]$ by comparing the cost when it is a regeneration interval with the cost when it is not. In the next section, we develop a DP algorithm for finding an optimal solution for LS-PC by using these results.

3. Dynamic Programming Algorithm

In this section, we propose a DP algorithm for the special case where the breakpoints of the production cost function are time invariant and the number of breakpoints is fixed; i.e., $b_t^i = b^i$ for all $t = 1, \ldots, n$ and $i = 0, \ldots, m$, where $m_t = m$ for all $t = 1, \ldots, n$ and m (≥ 1) is fixed.

This algorithm is a generalization of the algorithm given by Florian and Klein (1971) for the constant capacity lot-sizing problem.

Let e_i be a unit vector of size m in which the ith component is one and the other components are zero for $i = 1, \ldots, m$ and e_0 be a zero vector of size m.

3.1. Minimum Cost for an Interval $[j, l]$ with No Fractional Period

First, we compute the minimum cost for a regeneration interval $[j, l]$ with $1 \leq j \leq l \leq n-1$ and for an interval $[j, n]$ for $1 \leq j \leq n$ when there is no fractional period. To this end, we define the following function. Let $\tau \in \mathbb{Z}_+^m$ and $t \in \{j, \ldots, l\}$. If $l \leq n-1$, let $F_{jl}(t, \tau)$ be the minimum cost for periods j up to t during which τ_i times b^i, for $i = 1, \ldots, m$, units are produced, no fractional production is done, given that $s_{j-1} = s_l = 0$ and $s_u > 0$ for $u \in \{j, \ldots, \min\{t, l-1\}\}$. If $l = n$, then we define the same function by dropping the requirement that $s_l = 0$. For $j \leq t$, we let $d_{jt} = \sum_{i=j}^{t} d_i$.

Note that the amount of production between periods j and t is equal to $\sum_{i=1}^m \tau_i b^i$ and the number of periods in which production takes place is $\sum_{i=1}^m \tau_i$. If $t < l$ and $\sum_{i=1}^m \tau_i b^i \leq d_{jt}$, then we cannot have $s_t > 0$. Also, if $t = l$ and $\sum_{i=1}^m \tau_i b^i \neq d_{jl}$, then $s_l = 0$ is not possible. If $\sum_{i=1}^m \tau_i > t - j + 1$, the production schedule is infeasible.

For $i = 0, \ldots, m$, we let

$$F_{jl}(j, e_i) = \begin{cases} p_j(b^i) + h_j(b^i - d_j) & \text{if } d_j < b^i \text{ and } (j < l \text{ or } l = n), \\ p_j(b^i) & \text{if } d_j = b^i \text{ and } j = l, \\ \infty & \text{otherwise,} \end{cases}$$

and $F_{jl}(j, \tau) = \infty$ if $\sum_{i=1}^m \tau_i \geq 2$.

Let $t \in \{j+1, \ldots, l\}$ and $\tau \in \mathbb{Z}_+^m$. If we produce b^i units for some $i \in \{0, \ldots, m\}$ in period t, then the minimum cost for periods j to $t-1$ is $F_{jl}(t-1, \tau - e_i)$. Therefore, we compute $F_{jl}(t, \tau)$ as

$$F_{jl}(t, \tau) = \begin{cases} \infty & \text{if } \sum_{i=1}^m \tau_i > t - j + 1 \\ & \text{or } \left(\sum_{i=1}^m \tau_i b^i \leq d_{jt} \text{ and } t < l \right) \\ & \text{or } \left(\sum_{i=1}^m \tau_i b^i \neq d_{jl} \text{ and } t = l \text{ and } l < n \right) \\ & \text{or } \left(\sum_{i=1}^m \tau_i b^i < d_{jl} \text{ and } t = l = n \right); \\ \min\limits_{i=0,\ldots,m:\, \tau \geq e_i} \Big\{ F_{jl}(t-1, \tau - e_i) + p_t(b^i) & \\ \qquad + h_t\left(\sum_{i=1}^m \tau_i b^i - d_{jt} \right) \Big\} & \text{otherwise.} \end{cases}$$

We evaluate the recursion for increasing values of t and all possible values of τ. For given t and τ, $F_{jl}(t, \tau)$ can be computed in constant time since we assume that m is fixed. As $\tau_i \leq n$ for $i = 1, \ldots, m$, we have $O(n^m)$ possible τ vectors. As a result, the function F_{jl} can be evaluated in $O(n^{m+1})$ time for a given interval $[j, l]$.

3.2. Minimum Cost for an Interval $[j, l]$ with a Fractional Period

Next, we compute the minimum cost for a regeneration interval $[j, l]$ with $1 \leq j \leq n$ when the interval contains a fractional period. Note that for an interval $[j, n]$ that is part of an optimal solution, when the interval contains a fractional period, there exists an optimal solution with $s_n = 0$. Hence, we only consider regeneration intervals in this computation.

The minimum cost when a fractional period exists is computed for two separate cases:

Case a. The fractional production amount is less than b^m.

As we are interested in solutions with one fractional period, we know that there is no production greater than b^m.

Let $\tau \in \mathbb{Z}_+^m$, $\pi \in \mathbb{Z}_+^{m-1}$, and $t \in \{j, \ldots, l\}$. If τ_i times b^i, for $i = 1, \ldots, m$, units are produced in periods j up to $t-1$ and π_i times b^i, for $i = 1, \ldots, m-1$, and $\lfloor (d_{jl} - \sum_{i=1}^m \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i)/b^m \rfloor$ times b^m units are produced in periods $t+1$ to l, then the production amount in period t is equal to

$$\rho_{jl}(\tau, \pi) = d_{jl} - \sum_{i=1}^m \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i - \left\lfloor \frac{d_{jl} - \sum_{i=1}^m \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor b^m.$$

Let $G_{jl}(t, \tau, \pi)$ be the minimum cost for periods j up to t, during which τ_i times b^i units for $i = 1, \ldots, m$, are produced and a fractional production is done once, given that π_i times b^i, for $i = 1, \ldots, m-1$, and

$\lfloor (d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i)/b^m \rfloor$ times b^m units are produced after period t, $s_{j-1} = s_l = 0$, and $s_u > 0$ for $u \in \{j, \dots, \min\{t, l-1\}\}$.

Let $\tau \in \mathbb{Z}_+^m$ and $\pi \in \mathbb{Z}_+^{m-1}$. If $\sum_{i=1}^{m} \tau_i \geq 1$ or $d_{jl} \leq \sum_{i=1}^{m-1} \pi_i b^i$ or $\sum_{i=1}^{m-1} \pi_i + \lfloor (d_{jl} - \sum_{i=1}^{m-1} \pi_i b^i)/b^m \rfloor > l - j$ or $\rho_{jl}(e_0, \pi) \in \{0, b^1, \dots, b^m\} \cup (b^m, \infty)$, we set $G_{jl}(j, \tau, \pi) = \infty$. For other values, we compute

$$G_{jl}(j, e_0, \pi) = \begin{cases} p_j(\rho_{jl}(e_0, \pi)) + h_j(\rho_{jl}(e_0, \pi) - d_j) & \text{if } \rho_{jl}(e_0, \pi) > d_j \text{ and } j < l, \\ p_j(\rho_{jl}(e_0, \pi)) & \text{if } \rho_{jl}(e_0, \pi) = d_j \text{ and } j = l, \\ \infty & \text{otherwise.} \end{cases}$$

Now let $t \in \{j+1, \dots, l\}$, $\tau \in \mathbb{Z}_+^m$, and $\pi \in \mathbb{Z}_+^{m-1}$. If $\sum_{i=1}^{m} \tau_i > t - j$ or $\sum_{i=1}^{m-1} \pi_i + \lfloor (d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i)/b^m \rfloor > l - t$, then we set $G_{jl}(t, \tau, \pi) = \infty$. If $\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}(\tau, \pi) \leq d_{jt}$ and $t < l$, then $s_t \leq 0$, and if $\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}(\tau, \pi) \neq d_{jl}$ and $t = l$, then $s_l \neq 0$. If $d_{jl} < \sum_{i=1}^{m} \tau_i b^i + \sum_{i=1}^{m-1} \pi_i b^i$, then s_l cannot be zero. Moreover, we do not want to have $\rho_{jl}(\tau, \pi) \in \{0, b^1, \dots, b^m\} \cup (b^m, \infty)$. Hence, we set $G_{jl}(t, \tau, \pi) = \infty$ in these cases. For the remaining values, we compute

$$\begin{aligned} &G_{jl}(t, \tau, \pi) \\ &\quad = h_t\left(\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}(\tau, \pi) - d_{jt}\right) \\ &\qquad + \min\Big\{ F_{jl}(t-1, \tau) + p_t(\rho_{jl}(\tau, \pi)), \\ &\qquad\qquad \min_{i=0,\dots,m:\, \tau \geq e_i} \{G_{jl}(t-1, \tau - e_i, \pi + \bar{e}_i) + p_t(b^i)\} \Big\}, \end{aligned}$$

where $\bar{e}_i$ is the restriction of e_i to the first $m-1$ entries. Here, we first add the inventory holding cost. If the fractional production takes place at period t, then the production cost is $p_t(\rho_{jl}(\tau, \pi))$ and the minimum cost for periods j to $t-1$ is $F_{jl}(t-1, \tau)$. If we produce b^i units in period t for some $i \in \{0, \dots, m\}$, then the production cost is $p_t(b^i)$ and the minimum cost for periods j to $t-1$ is $G_{jl}(t-1, \tau - e_i, \pi + \bar{e}_i)$ since the fractional period is before period t.

For given t, τ, and π, $G_{jl}(t, \tau, \pi)$ can be computed in constant time. Hence G_{jl} can be evaluated in $O(n^{2m})$ time.

Case b. The fractional production amount is greater than b^m.

Let $\tau \in \mathbb{Z}_+^m$, $\hat{\pi} \in \mathbb{Z}_+^m$, $t \in \{j, \dots, l\}$, and $\hat{G}_{jl}(t, \tau, \hat{\pi})$ be the minimum cost for periods j up to t during which τ_i times b^i units, for $i = 1, \dots, m$, are produced and a fractional production $\hat{\rho}_{jl}(\tau, \hat{\pi}) = d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m} \hat{\pi}_i b^i > b^m$ is done once, given that $\hat{\pi}_i$ times b^i, for $i = 1, \dots, m$, units are produced after period t, $s_{j-1} = s_l = 0$, and $s_u > 0$ for $u \in \{j, \dots, \min\{t, l-1\}\}$. The function $\hat{G}_{jl}$ can be computed in a similar way to G_{jl}. As the dimension of the vector $\hat{\pi}$ is one more than the one of π, computing $\hat{G}_{jl}$ requires $O(n^{2m+1})$ time.

3.3. Time Complexity

Overall, we can find the minimum cost for interval $[j, l]$ as

$$\mu_{jl} = \min_{\tau \in \{0,\dots,n\}^m} \{F_{jl}(l, \tau), G_{jl}(l, \tau, \bar{e}_0), \hat{G}_{jl}(l, \tau, e_0)\}.$$

THEOREM 3. *The LS-PC is polynomially solvable when the breakpoints of the production cost function are time invariant and when the number of breakpoints is fixed.*

PROOF. For an interval $[j, l]$ with $1 \leq j \leq l \leq n$, as evaluating the functions F_{jl}, G_{jl}, and $\hat{G}_{jl}$ take $O(n^{m+1})$, $O(n^{2m})$, and $O(n^{2m+1})$ time, respectively, the minimum cost μ_{jl} can be computed in $O(n^{2m+1})$ time. Once these costs are computed, we can solve the problem by solving a shortest path problem, as is done for the classical lot-sizing problem. Let $G = (V, A)$ be a directed graph for $V = \{1, \dots, n+1\}$ and $A = \{(j, l+1)\colon 1 \leq j \leq l \leq n\}$. The shortest path problem from node 1 to node $n+1$ in the graph G with cost μ_{jl} on arc $(j, l+1)$ with $d_{jl} > 0$ and cost 0 on arc $(j, l+1)$ with $d_{jl} = 0$, solves our problem. As μ_{jl} can be computed in $O(n^{2m+1})$ time and there are $O(n^2)$ intervals, we require $O(n^{2m+3})$ time to construct the graph. This dominates the time to compute a shortest path. Therefore, the overall complexity is $O(n^{2m+3})$ and is polynomial for fixed m. □

3.4. Special Cases

Now we discuss some special cases. Suppose that the production amount in any period cannot exceed a given capacity C. This can be modeled by setting $b^m = C$ and $p_t(x) = \infty$ for $x \in (b^m, \infty)$ and $t = 1, \dots, n$. In this case $\hat{G}_{jl} = \infty$ for all intervals $[j, l]$. Then the overall complexity of the algorithm decreases to $O(n^{2m+2})$. The constant capacity lot-sizing problem is the special case with $m = 1$. For this special case our algorithm runs in $O(n^4)$ time and hence has the same time complexity as the one of Florian and Klein (1971).

Hellion et al. (2012) study the capacitated lot-sizing problem with concave costs, minimum order quantities (L), and constant capacities (C). To model this special case, we let $p_t(x) = \infty$ if $x \in (0, L) \cup (C, \infty)$, so we assume that $m = 2$. In this case, again, $\hat{G}_{jl} = \infty$ for all intervals $[j, l]$. Therefore, our DP algorithm can solve this special case of the problem in $O(n^6)$ time, which is equal to the computational complexity of the algorithm of Hellion et al. (2013).

Atamtürk and Hochbaum (2001) propose an $O(n^5)$ algorithm for the special case where the production cost function has two pieces: the first piece corresponds

to regular work and the second piece represents subcontracting. As $m=1$, our DP algorithm can also solve this problem in $O(n^5)$ time.

If we assume that backordering is allowed, we can redefine $h_t(s_t)$ as the cost of holding s_t units of inventory during period t if $s_t > 0$ and the cost of backordering s_t units during period t if $s_t < 0$. We assume that $h_t(\cdot)$ is a concave function on both $(-\infty, 0]$ and $[0, \infty)$; consequently $h_t(\cdot)$ is a piecewise concave function on $\mathbb{R}$. If we change the condition $s_t > 0$ to $s_t \neq 0$ in the definition of regeneration intervals, Theorems 1 and 2 still hold true in the case of backlogging. We can use the DP given in this section to solve the problem with some small modifications without changing the computational complexity.

In conclusion, for the cases discussed above, our algorithm's performance is as good as the performance of the algorithms in the literature.

Finally, note that in the case when the inventory holding cost is a piecewise concave function, it can be handled similarly if we assume that the breakpoints of the holding cost function are also time invariant and the number of breakpoints is fixed; i.e., $r_t^i = r^i$ for all $t = 1, \ldots, n$ and $i = 1, \ldots, q$, where $q_t = q$ for all $t = 1, \ldots, n$ and $q(\geq 1)$ is fixed. According to the redefinition of regeneration interval given in Remark 1, for each regeneration interval $[j, l]$ now we need to know the starting and ending inventories, $s_{j-1}, s_l \in \{r^1, \ldots, r^q\}$. Therefore, for each function defined for (regeneration) interval $[j, l]$ in the DP algorithm, additional initial (and final) inventory levels should be appended. For example, $F_{jl\alpha\beta}(t, \tau)$ will give the minimum cost for periods j up to t in a regeneration interval $[j, l]$ with initial inventory $s_{j-1} = r^\alpha$ and final inventory $s_l = r^\beta$, given that τ_i times b^i, for $i = 1, \ldots, m$, units are produced and no fractional production is done until period t. Moreover, for the last interval $[j, n]$ (which may not be a regeneration interval), as the final inventory level may not be equal to a breakpoint level, we need to define $\hat{F}_{j\alpha}(t, \tau)$ as the minimum cost for periods j up to t in the interval $[j, n]$ with $s_{j-1} = r^\alpha$ and $s_t \notin \{r^1, \ldots, r^q\}$ for $t = j, \ldots, n$, given that τ_i times b^i, for $i = 1, \ldots, m$, units are produced and no fractional production is done until period t. Similarly, functions $G_{jl\alpha\beta}$ and $\hat{G}_{jl\alpha\beta}$ will give the minimum costs for the regeneration interval $[j, l]$ with initial inventory $s_{j-1} = r^\alpha$ and final inventory $s_l = r^\beta$ when there exists exactly one fractional period.

4. Computational Results

In this section, we examine the computational efficiency of our algorithm. Although our algorithm can solve the lot-sizing problem with any piecewise concave function, to compare the algorithm's performance with an MIP solver, we use piecewise linear production cost functions and linear holding costs in our computational study.

We tested three well-known linearizations of piecewise linear functions: multiple choice (MC), incremental formulations, and convex combination formulations (see, e.g., Croxton et al. 2003). Our preliminary tests showed that the multiple choice linearization outperformed the other two linearizations. For the capacitated lot-sizing problem, this linearization is as follows:

(MC)

$$\min \left\{ \sum_{t=1}^{n} \sum_{j=1}^{m} (f_t^j y_t^j + c_t^j x_t^j) + \sum_{t=1}^{n} h_t s_t \right\} \tag{5}$$

s.t.

$$s_{t-1} + \sum_{j=1}^{m} x_t^j = d_t + s_t, \quad t = 1, \ldots, n, \tag{6}$$

$$b^{j-1} y_t^j \leq x_t^j \leq b^j y_t^j, \quad t = 1, \ldots, n,\ j = 1, \ldots, m, \tag{7}$$

$$\sum_{j=1}^{m} y_t^j \leq 1, \quad t = 1, \ldots, n, \tag{8}$$

$$s_0 = 0, \tag{9}$$

$$s, x \geq 0, \quad y \text{ binary}. \tag{10}$$

In this formulation, if the production amount is in the jth piece of the cost function, then there is a fixed cost f_t^j and a variable cost c_t^j (see Figure 3). We assume that the production cost function is lower semicontinuous. The inventory holding cost function is a linear function and h_t is the cost of holding one unit of inventory during period t. The variable y_t^j is equal to one if the production amount in period t lies in the segment $[b^{j-1}, b^j]$. Constraints (8) ensure that at most one of the y_t^j variables is one in period t. Consequently, constraints (7) guarantee that x_t^j should be in the segment $[b^{j-1}, b^j]$ if $y_t^j = 1$, and at most one of the production variables x_t^j will be nonzero for t.

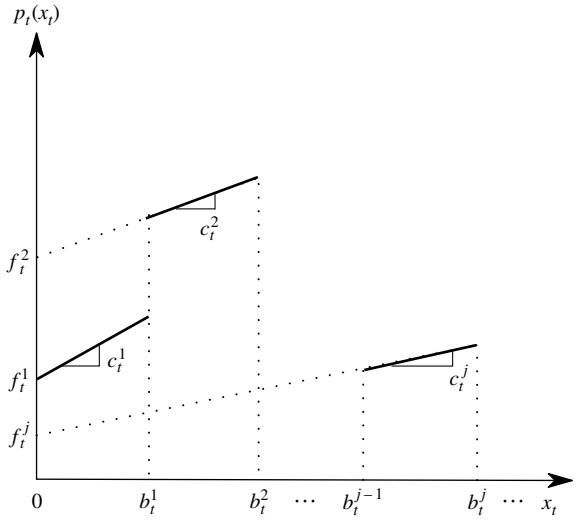

Figure 3 Production Cost Function for MC

Table 1 **Experimental Factors when $m = 2$**

Factors	No. of levels	Experimental settings 1	2	3	4
Fixed costs (f^1, f^2)	3	(3,000, 6,000)	(3,000, 4,000)	(3,000, 7,500)	
Variable costs (c^1, c^2)	4	(0, 0)	(0.5, 1)	(1, 0.5)	(1, 1)
Breakpoints (b^1, b^2)	3	(800, 1,600)	(900, 1,800)	(1,000, 2,000)	

Constraints (6) are inventory balance constraints and the objective function (5) is the sum of production and inventory holding costs. By constraints (9), we impose the requirement that the initial inventory is zero.

We implemented the formulation MC in Xpress 1.22 and the DP in Java (JDK 7) and ran them on a 2.53 GHz Intel Core 2 Duo machine with a 4 GB memory running Windows 7. We let the solver run for 1,000 seconds.

In our computational study, we only consider the capacitated problem and ignore the last piece of the cost function, since we assume that it has a very large cost. We first analyze two cost segment instances, i.e., $m = 2$, and create randomly generated problems with different cost parameters, all time invariant, as summarized in Table 1. Furthermore, for 40 and 50 period cases we assume that the demand has the same distribution, and the holding cost is the same such that the inventory holding cost is 0.05 and the demand is an integer drawn from a uniform distribution, $U[400, 500]$. Consequently, for each case there are 36 randomly generated test problems. We also generated instances as described in Hellion et al. (2012). However, we do not report the results for these instances, as all were solved in less than one second by a commercial solver for $n = 40$.

For 20 periods and three cost segment instances, we consider different cost structures, as summarized in Table 2. For example, increasing unit costs (1.3, 1.5, 1.8) may represent a system with subcontracting, or decreasing unit costs (1.8, 1.5, 1.3) may represent quantity discounts. Also, note that unit costs (1.5, 1.3, 1.8) can be seen as a combination of these two systems (Figure 2(a)). We now generate 42 problems randomly, for which we assume that the inventory holding cost is 0.05 and the demand is an integer drawn from a uniform distribution, $U[500, 600]$.

We also consider instances with 15 periods and four cost segments. We generate 24 instances where the inventory holding cost is 0.05 and the demand is an integer from a uniform distribution, $U[400, 500]$. Other experimental settings for these instances are given in Table 3.

To improve the bounds obtained from the formulation MC, we use the valid inequalities recently developed by Sanjeevi and Kianfar (2012) for the multimodule lot-sizing problem. These inequalities are based on mixing set relaxations. We briefly describe these inequalities in the online supplement (available as supplemental material at http://dx.doi.org/10.1287/ijoc.2014.0597).

In Table 4, we present the results for $n = 50$ and $m = 2$. The results for other n and m values are given in the online supplement. In these tables, we report the results for the formulation MC, the formulation MC with valid inequalities (MC-CUTS), and our DP algorithm. Columns BUB, LPGap, and FGap correspond to the best upper bound obtained by the solver within the time limit, the percentage gap between the optimal value of the LP relaxation and the optimal value of the integer problem, and the percentage gap between the best lower and upper bounds attained at the end of the time limit, respectively. Some instances are solved to optimality by MC or MC-CUTS; in this case we report the time spent to solve the formulation in parentheses in column (Time). Columns OPT and Time under DP correspond to the optimal value of the problem and the solution time of the DP algorithm.

Table 2 **Experimental Factors when $m = 3$**

Factors	No. of levels	Experimental settings 1	2	3	4	5	6	7
(f^1, f^2, f^3)	3	(3,000, 6,000, 9,000)	(3,000, 5,000, 6,500)	(3,000, 3,500, 5,000)				
(b^1, b^2, b^3)	2	(500, 1,000, 1,500)	(600, 1,200, 1,800)					
(c^1, c^2, c^3)	7	(0, 0, 0)	(1.3, 1.5, 1.8)	(1.3, 1.8, 1.5)	(1.5, 1.3, 1.8)	(1.5, 1.8, 1.3)	(1.8, 1.3, 1.5)	(1.8, 1.5, 1.3)

Table 3 **Experimental Factors when $m = 4$**

Factors	No. of levels	Experimental settings 1	2	3	4
$f = (f^1, f^2, f^3, f^4)$	2	(3,000, 6,000, 9,000, 12,000)	(3,000, 5,500, 8,000, 10,000)		
$b = (b^1, b^2, b^3, b^4)$	3	(450, 900, 1,350, 1,800)	(600, 1,200, 1,800, 2,400)	(750, 1,500, 2,250, 3,000)	
$c = (c^1, c^2, c^3, c^4)$	4	(0.6, 0.8, 1.0, 1.3)	(0.8, 0.6, 1.3, 1.0)	(1.0, 0.6, 0.8, 1.3)	(1.3, 0.6, 0.8, 1.0)

Table 4 **Results for $n = 50$ and $m = 2$**

Instance			MC			MC-CUTS			DP	
(f^1, f^2)	(c^1, c^2)	(b^1, b^2)	BUB	LPGap	FGap	BUB	LPGap	FGap (time)	OPT	Time
1	1	1	87,849.5	3.97	3.17	87,831.3	2.77	2.69	87,727.0	719.77
		2	76,621.0	1.88	0.99	76,313.5	0.24	(6)	76,313.5	677.49
		3	70,052.2	3.65	2.20	70,300.4	1.64	1.74	69,943.9	653.19
	2	1	99,074.2	3.52	2.76	99,044.9	2.40	1.26	98,959.0	744.83
		2	87,718.0	1.64	0.50	87,545.5	0.19	(2)	87,545.5	689.00
		3	81,292.8	3.14	1.86	81,254.1	1.40	0.95	81,175.9	658.82
	3	1	100,021.0	4.52	2.57	100,401.1	2.73	2.39	99,990.3	741.92
		2	89,027.8	3.27	0.42	89,026.0	0.96	(10)	89,026.0	696.00
		3	82,740.4	4.92	1.76	82,695.7	2.21	0.42	82,695.1	658.40
	1	1	110,270.0	3.16	2.49	110,245.6	2.21	2.04	110,191.0	720.37
		2	99,265.0	1.45	0.98	98,777.5	0.19	(11)	98,777.5	685.54
		3	92,473.8	2.76	1.56	92,574.75	1.24	1.06	92,407.9	653.36
2	1	1	60,591.9	7.23	4.01	60,598.8	2.80	1.16	60,537.6	737.28
		2	53,386.0	6.43	1.88	53,348.5	1.49	(5)	53,348.5	688.72
		3	49,067.8	8.38	3.02	49,035.6	2.81	(114)	49,035.6	660.97
	2	1	82,665.6	4.82	2.46	82,684.0	2.31	1.31	82,601.6	735.74
		2	75,490.0	3.95	0.76	75,362.5	1.08	(8)	75,362.5	682.21
		3	71,027.8	5.08	1.37	70,999.6	2.13	(735)	70,999.6	648.44
	3	1	72,014.6	6.39	3.61	71,990.3	2.45	(523)	71,990.3	744.03
		2	64,885.8	5.72	1.94	64,862.9	1.26	(7)	64,862.9	671.57
		3	60,748.9	7.47	3.05	60,695.1	2.49	(12)	60,695.1	642.07
	4	1	83,054.9	5.27	2.92	83,001.6	2.04	0.77	83,001.6	737.10
		2	75,850.0	4.52	1.29	75,812.5	1.05	(8)	75,812.5	674.39
		3	71,511.3	5.74	1.98	71,499.6	1.88	(84)	71,499.6	650.80
3	1	1	87,801.8	3.97	3.04	87,781.6	2.68	0.59	87,727.0	727.36
		2	76,440.4	1.88	0.48	76,313.5	0.22	(1)	76,313.5	676.49
		3	70,020.4	3.65	2.06	70,044.4	1.63	1.07	69,943.8	642.60
	2	1	98,989.1	3.52	2.62	98,959.0	2.37	(24)	98,959.0	732.44
		2	87,725.5	1.64	0.52	87,545.5	0.19	(2)	87,545.5	680.33
		3	81,269.5	3.14	1.76	81,391.1	1.40	1.07	81,175.9	636.95
	3	1	110,247.0	3.16	2.41	111,020.9	2.16	1.99	110,191.0	720.73
		2	98,905.0	1.45	0.40	98,777.5	0.17	(2)	98,777.5	681.56
		3	92,543.1	2.76	1.66	92,529.1	1.23	0.84	92,407.9	639.65
	4	1	110,341.0	3.16	2.46	111,865.6	2.13	2.02	110,191.0	725.70
		2	98,867.5	1.45	0.35	98,777.5	0.17	(2)	98,777.5	677.62
		3	92,483.4	2.76	1.55	92,588.8	1.23	0.94	92,407.9	645.63

We observe that none of the instances are solved to optimality using MC in 1,000 seconds for 40 and 50 periods and two piece instances and only 11 of the 42 instances of the 20 periods and three piece instances are solved to optimality. As expected, the performance varies from one instance to another: the LPGap between 1% and 10% and the final gap between 0% and 5%. MC-CUTS can solve some instances in a second, whereas for others the final gap can be as large as 3%–4%. Clearly, the DP has a stable solution time. Moreover, the proposed DP can handle all of these different cost functions and solves the problems

Table 5 **Summary of the Results**

	MC						MC-CUTS						DP		
	LPGap			FGap			LPGap			FGap			Time		
(n, m)	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
(40, 2)	2.07	5.08	10.74	1.12	2.94	5.06	0.78	2.31	4.02	0.00	1.43	3.90	144.9	154.2	162.5
(50, 2)	1.45	4.04	8.38	0.35	1.91	4.01	0.17	1.60	2.81	0.00	0.68	2.69	637.0	687.8	744.8
(20, 3)	2.41	4.08	6.23	0.00	1.27	3.40	1.43	2.70	3.89	0.00	1.10	3.42	149.5	162.4	176.8
(15, 4)	5.41	6.88	9.26	0.00	2.76	5.60	5.01	6.39	8.41	0.00	2.09	5.02	400.3	420.5	446.4

Table 6 **Experimental Factors for the Heuristic Solution Approach when $m = 3$**

Factors	No. of levels	Experimental settings 1	2	3
(f^1, f^2, f^3)	2	(3,000, 6,000, 9,000)	(3,000, 5,000, 6,000)	
(b^1, b^2, b^3)	2	(800, 1,600, 2,400)	(1,000, 2,000, 3,000)	
(c^1, c^2, c^3)	3	(0, 0, 0)	(1, 0.5, 0.7)	(1, 0.5, 1)

to optimality, whereas the MC formulation in Xpress may end up with an optimality gap of 3% at the end of the time limit of 1,000 seconds.

A summary of the results are given in Table 5. In Table 5, columns named Max, Avg, and Min show the maximum, average, and minimum values of the corresponding columns. As can be observed from Table 5, when n or m increases, as expected, the solution time of DP gets larger. In contrast, the DP solves all of the instances in less than 1,000 seconds, whereas Xpress may end up with positive optimality gaps even for the strengthened formulation.

We conclude that for the small or medium-sized instances, the DP outperforms the MIP approach. Furthermore, for solving larger instances of the problem we can easily modify the DP to get good quality solutions in reasonable computation times, as discussed below.

5. Heuristic for Solving Larger Instances

The computational complexity of our DP algorithm strongly depends on the number of different τ, π, and $\hat{\pi}$ vectors since we need to evaluate the functions F_{jl}, G_{jl}, and $\hat{G}_{jl}$ for all possible τ, π, and $\hat{\pi}$ vectors. As there are $O(n^m)$ possible τ, $\hat{\pi}$, and $O(n^{m-1})$ π vectors, for larger n and m it may not be a good choice to use the DP directly. Moreover, as Xpress could not solve some medium-sized instances in our experiments, we expect its performance to get worse for larger instances.

To get a good solution for larger instances in a reasonable time, we develop a heuristic method based on our DP algorithm. We heuristically restrict the length of any regeneration interval (and also the final interval, which may not be a regeneration interval) of a solution. Let ν $(1 \le \nu \le n)$ be a given upper bound on the length of any regeneration interval. We consider the interval $[j, l]$, $1 \le j \le l \le n$ and find the minimum cost μ_{jl} if $l - j + 1 \le \nu$. Consequently, we reduce the number

Table 7 **Results of the Heuristic for $m = 2$**

	Instance			MC-CUTS 1,000 seconds		MC-CUTS 2,000 seconds		DP-HEUR			
n	(f^1, f^2)	(c^1, c^2)	(b^1, b^2)	BUB	Gap	BUB	Gap	ν	BUB	Gap	Time
80	1	1	3	112,908.7	2.20	112,908.7	2.19	10	118,080.8	6.48	0.7
				(609)	(643)		(1,350)	12	112,492.6	1.84	1.5
								14	**112,488.3**	**1.83**	**3.2**
								16	112,488.3	1.83	6.2
								18	112,488.3	1.83	11.4
								20	112,488.3	1.83	20.2
								22	112,488.3	1.83	31.9
	1	4	1	175,495.1	0.36	175,495.1	0.35	10	175,376.2	0.29	0.7
				(931)	(931)		(1,226)	12	175,351.3	0.28	1.6
								14	175,338.6	0.27	3.3
								16	175,338.6	0.27	6.6
								18	175,338.6	0.27	12.6
								20	**175,323.8**	**0.26**	**21.0**
								22	175,323.8	0.26	34.5
100	1	1	2	154,876	1.22	154,671.7	1.08	10	157,515.1	2.87	0.9
				(44)	(735)	(1,313)	(1,313)	12	157,297.5	2.74	1.9
								14	157,277.4	2.73	4.1
								16	157,250.5	2.71	8.2
								18	157,250.5	2.71	15.6
								20	154,558.1	1.02	26.7
								22	**154,498.1**	**0.98**	**44.1**
	1	4	1	219,162.7	1.05	218,739.3	0.85	10	220,696.6	1.73	0.9
				(781)	(781)	(1,641)	(1,641)	12	220,694.2	1.73	2.0
								14	220,686.7	1.73	4.2
								16	220,686.7	1.73	8.4
								18	220,686.7	1.73	15.6
								20	**217,918.7**	**0.48**	**27.2**
								22	217,912.5	0.48	46.2

of intervals to be considered to $O(\nu n)$. Moreover, for a given interval $[j, l]$ the number of possible τ, π, and $\hat{\pi}$ vectors becomes $O(\nu^m)$, $O(\nu^{m-1})$, and $O(\nu^m)$, respectively. Therefore, with this restriction we reduce the state space and, consequently, the time complexity of the DP.

Note that when $\nu = n$, the restriction becomes redundant and the heuristic is the same as the exact DP. If $\nu = 1$, then the (trivial) solution is to produce in every period as much as the demand of that period. Moreover, if we know the maximum regeneration interval length in an optimal solution, say ν^*, then we can set $\nu = \nu^*$ and obtain an optimal solution to the problem with the heuristic. The performance of this heuristic depends on ν; we may obtain a better quality solution with larger ν but in longer computation time.

To test this solution method, we consider different ν values and compare the total cost of the solution obtained by this method with the lower bound obtained from MC-CUTS. We use larger instances that are created the same way as the instances used in the previous section. We have selected a representative set of instances to test the solution quality of the proposed heuristic. The experimental factors are listed in Table 6. For all

Table 8 **Results of the Heuristic for $m = 3$**

				MC-CUTS							
	Instance			1,000 seconds		2,000 seconds		DP-HEUR			
n	(f^1, f^2, f^3)	(c^1, c^2, c^3)	(b^1, b^2, b^3)	BUB	Gap	BUB	Gap	ν	BUB	Gap	Time
50	1	1	2	70,146	2.33	70,146	2.31	10	72,742.1	5.81	8.6
				(355)	(971)		(1,742)	12	69,948.1	2.05	27.3
								14	**69,943.9**	**2.04**	**73.8**
								16	69,943.9	2.04	175.5
								18	69,943.9	2.04	379.6
								20	69,943.9	2.04	763.1
								22	69,943.9	2.04	1,448.1
	2	3	1	83,142.8	1.73	83,126.2	1.68	10	83,433.3	2.07	8.7
				(310)	(885)	(1,495)	(1,737)	12	83,433.3	2.07	27.6
								14	83,430.8	2.07	75.0
								16	**83,041.3**	**1.61**	**179.9**
								18	83,041.3	1.61	394.3
								20	83,041.3	1.61	806.2
								22	83,041.3	1.61	1,507.7
80	1	1	2	105,500	1.12	105,472.8	1.06	10	110,780.9	5.83	14.2
				(253)	(903)	(1,475)	(1,779)	12	108,169.7	3.56	46.1
								14	105,432.6	1.06	124.8
								16	105,432.6	1.06	306.2
								18	**105,429.5**	**1.05**	**676.0**
								20	105,429.5	1.05	1,396.7
								22	105,429.5	1.05	2,700.8
	2	2	2	112,939.7	2.67	112,939.7	2.66	10	118,080.8	6.90	14.0
				(670)	(903)		(1,032)	12	112,492.6	2.28	48.6
								14	**112,488.3**	**2.27**	**124.6**
								16	112,488.3	2.27	307.4
								18	112,488.3	2.27	682.2
								20	112,488.3	2.27	1,402.4
								22	112,488.3	2.27	2,729.4
100	1	1	1	173,543.3	1.43	173,543.3	1.42	10	175,485.6	2.52	19.3
				(424)	(424)		(1,620)	12	175,483.2	2.52	59.6
								14	175,475.7	2.51	165.7
								16	175,475.7	2.51	414.0
								18	175,475.7	2.51	918.9
								20	**172,707.7**	**0.95**	**1,915.9**
								22	172,701.5	0.95	3,759.9
	2	2	2	131,319.6	0.93	131,319.6	0.91	10	137,834.9	5.61	17.6
				(442)	(892)		(1,595)	12	135,443.9	3.95	58.1
								14	132,401.3	1.74	160.4
								16	132,401.3	1.74	394.0
								18	131,275.1	0.90	883.4
								20	**131,234.7**	**0.87**	**1,827.2**
								22	131,234.7	0.87	3,575.9

instances, we assume that the inventory holding cost is 0.05 and the demand is an integer drawn from a uniform distribution, $U[400, 500]$, for all periods.

Tables 7 and 8 summarize the results of this experiment for $m = 2$ and $m = 3$, respectively. Columns under MC-CUTS represent the results for the formulation MC with valid inequalities, and the columns under DP-HEUR represent the results of our heuristic method. For each instance, we consider different ν values to see the trade-off between the solution quality and the solution time, and we indicate the rows with the minimum optimality gap in bold. With MC-CUTS, we let Xpress run 1,000 and 2,000 seconds and calculate the gap of the heuristic solution using the best lower bound obtained in 1,000 seconds. We also report the CPU times at which the best upper bound and the best optimality gap are attained, in parentheses under their corresponding values.

As can be seen from Tables 7 and 8, letting Xpress run for an additional 1,000 seconds results in very little improvement in the final gaps. When the cost function has two pieces (Table 7), in all of the test instances, the heuristic finds better solutions than MC-CUTS in less than 50 seconds. Moreover, as can be seen in the table, when ν increases, the computation time increases (as expected) but the increase is not very fast. Therefore, the user can select a higher ν value and may obtain better solutions in reasonable computation times.

In Table 8, we report the results for the instances with three pieces. For 50 and 80 periods, the heuristic finds better solutions than MC-CUTS in very short computation times. For 100 periods, we again find better solutions using the heuristic algorithm but the computation time of the algorithm is about 2,000 seconds. Note that for the second instance of 100 periods, the solution found for $\nu = 18$ (in less than 1,000 seconds) is also a better solution than that of MC-CUTS. Moreover, we believe that by letting Xpress run for more than 2,000 seconds we can only obtain slightly better optimality gaps. Thus, when $m = 3$, the heuristic algorithm still reports better solutions than the MIP approach in less computation time. Furthermore, according to Tables 7 and 8, similar to the exact DP, for given n, m, and ν values, the computation time of the heuristic algorithm is stable.

6. Conclusion

In this paper, we studied the LS-PC. A piecewise concave function can represent economies of scale, discounts, subcontracting, overloading, minimum order quantities, and capacities. The computational complexity of this problem was an open question in the literature. We developed a DP algorithm and showed that the problem is polynomially solvable when the number of breakpoints of the production cost function are fixed and time invariant. The algorithm performs well for small and medium-sized instances and can easily be modified to be used as a heuristic for larger instances.

As expected, it can be observed from our computational study, that when m increases the solution time of the DP increases rapidly. For example, when $m = 5$, the solution time of the (exact) DP is about 100 seconds for $n = 4$, 350 seconds for $n = 6$, and 1,000 seconds for $n = 8$. Therefore, a different approach is required to solve problems with more breakpoints.

It may also be interesting to consider the problem when one of the pieces of the production cost function is convex (but not linear), which means that the function is not piecewise concave. A convex function can indicate increasing marginal costs; therefore, the convex part of this function may represent overloading or cost of extra use of a resource.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ijoc.2014.0597.

Acknowledgments

This research is partially supported by TUBITAK [Grant 112M220]. The research of the second author is supported by the Turkish Academy of Sciences.

References

Archetti C, Bertazzi L, Speranza MG (2011) Polynomial cases of the economic lot sizing problem with quantity discounts. Technical report, Department of Quantitative Methods, University of Brescia, Brescia, Italy.

Atamtürk A, Hochbaum DS (2001) Capacity acquisition, subcontracting, and lot sizing. *Management Sci.* 47:1081–1100.

Chan LMA, Muriel A, Shen Z-J, Simchi-Levi D (2002) On the effectiveness of zero-inventory-ordering policies for economic lot sizing model with a class of piecewise linear cost structures. *Oper. Res.* 50:1058–1067.

Croxton KL, Gendron B, Magnanti TL (2003) A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Sci.* 49:1268–1273.

Federgruen A, Lee C-Y (1990) The dynamic lot size model with quantity discount. *Naval Res. Logist.* 37:707–713.

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

Hellion B, Mangione F, Penz B (2012) A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs. *Eur. J. Oper. Res.* 222:10–16.

Hellion B, Mangione F, Penz B (2013) Corrigendum to "a polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs." *Eur. J. Oper. Res.* 229:279.

Li C-L, Ou J, Hsu VN (2012) Dynamic lot sizing with all-units discount and resales. *Naval Res. Logist.* 59:230–243.

Okhrin I, Richter K (2011) An $O(T^3)$ algorithm for the capacitated lot sizing problem with minimum order quantities. *Eur. J. Oper. Res.* 211:507–514.

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

Sanjeevi S, Kianfar K (2012) Mixed n-step MIR inequalities: Facets for the n-mixing set. *Discrete Optim.* 9:216–235.

Shaw DX, Wagelmans APM (1998) An algorithm for single-item capacitated economic lot-sizing problem with piecewise linear production costs and general holding costs. *Management Sci.* 44:831–838.

Swoveland C (1975) A deterministic multi-period production planning model with piecewise concave production and holding-backorder costs. *Management Sci.* 21:1007–1013.

VanHoesel CPM, Wagelmans APM (1996) An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Sci.* 42:142–150.

Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. *Management Sci.* 5:89–96.

Zangwill IW (1966) A deterministic multi-period production scheduling model with backlogging. *Management Sci.* 13:105–119.

Zangwill IW (1967) The piecewise concave function. *Management Sci.* 13:900–912.