Single-item lot-sizing with quantity discount and bounded inventory

Douglas G. Down[a], George Karakostas[a,*], Stavros G. Kolliopoulos[b], Somayye Rostami[a]

[a] *Department of Computing and Software, McMaster University, Hamilton ON, Canada*
[b] *Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece*

Abstract

In this paper, an efficient $O(n^2)$ algorithm is proposed to solve a special case of single-item lot-sizing problems (SILSP) in which both the production and holding costs are piecewise linear, there is an all-unit discount with one breakpoint for the production cost, and the inventory is bounded. The algorithm is based on a key structural property that may be of more general interest, that of a just-in-time ordering policy. Finally, we show that when the problem is extended to two items, it is NP-complete.

Keywords: lot-sizing, quantity discount, bounded inventory

1. Introduction

Lot-sizing problems have seen continued research interest for the past six decades. Our focus is on single-item lot-sizing problems (SILSP), for which an extensive survey is provided in [1]. In the basic SILSP, there is a time-varying, but known in advance, demand for a single product over a time horizon of n periods. One wishes to determine the periods in which production (or purchase) of the product will take place, and the quantities that will be produced. The cost function is the summation of production and holding costs, where the production cost may include a fixed setup-cost component. The goal is to find a sequence of orders that satisfies the demands and minimizes the total cost. Several variations of the main problem can be defined, by including constraints such as production capacities, bounded inventory, stochastic demands, backlogging, and lost sales [1]. A typical assumption in the literature is that the cost function is piecewise concave, which provides the flexibility to model scenarios such as discounts, minimum quantity requirements, and capacities [2]. A function with m breakpoints is piecewise concave if it is concave over each interval between two adjacent breakpoints. The case of capacitated production or bounded inventory can be modelled by adding a breakpoint to the production or holding cost function, and defining an infinite cost for the last interval [4].

In this paper, we study the very basic scenario where in every period one can buy the item at two price levels: expensively or cheaply depending on the quantity of units purchased. Demands have to be satisfied on time, without backlogging, and there are no order capacities, or set-up costs. In order to anticipate future demand, one may stock units bought cheaply, but this is curbed in two ways: hoarding incurs a holding cost *and* is limited by the hard constraint of a bound on the size of the inventory. In SILSP terminology, this is an all-unit discount pricing scheme with one breakpoint for the production cost and bounded inventory. *All-unit* means that the discount is applied to the entire volume of a purchase if this volume falls within a specified range. In our case, the range is any quantity at least equal to the single production breakpoint Q. To our knowledge this basic setting has not been considered separately in the literature. We show that an optimal solution can be found in polynomial time, and how exploiting a just-in-time ordering policy achieves significant running-time improvements. We assume that the production cost is a piecewise linear (and, hence, concave) function with two segments. The holding cost function is linear.

Solutions for SILSP problems are typically found using dynamic programming. There are several results in the literature, based on characterizations of the structure of an optimal solution that can be leveraged to develop efficient algorithms [3],[5],[6]. A seminal work of this nature (and for lot sizing in general) is [3]. They consider a problem with fixed unit production cost (the final inventory value is zero and only the setup cost changes) and linear holding cost, for which they derive an algorithm with time complexity $O(n^2)$. In [6], constraints on the production and inventory quantities are studied. For the case of bounded inventory and concave cost function, they propose an algorithm with complexity $O(n^3)$. In [7], an $O(mn^3)$ algorithm is presented for the problem with capacitated production and piecewise linear cost functions with an average of m breakpoints in each time period. In [8], an $O(n^{m+2} \log n)$ algorithm is constructed for the uncapacitated version of

*Corresponding author
Email addresses: dowmd@mcmaster.ca (Douglas G. Down), karakos@mcmaster.ca (George Karakostas), sgk@di.uoa.gr (Stavros G. Kolliopoulos), rostas1@mcmaster.ca (Somayye Rostami)

the problem in [7]. In [4], a model with an all-unit discount with one breakpoint and capacitated production is studied. The complexity of their algorithm is $O(n^4)$. In [9], a cost function with one breakpoint is considered and algorithms for the cases of all-unit and incremental discount with complexity of $O(n^2)$ and $O(n^3)$ respectively, are constructed. The production and holding costs considered in [10] are linear for the case of capacitated production. They study the NP-completeness of the problem based on five cases of general, nonincreasing, nondecreasing, constant and zero-value functions for the setup, production and holding costs, and production capacity. An $O(n^3)$ algorithm is constructed in [11] for the case of concave production cost and linear holding cost with constant capacities. An $O(n^2)$ algorithm is presented in [12] and fixed by [13] for the case of linear costs with a lower and upper-bounded inventory. A fully polynomial approximation scheme is proposed in [14] for an NP-hard case of SILSP. A more efficient approximation algorithm is proposed in [15]. In [16] a fixed cost and variable cost per unit for a change in the inventory size at each time period are also considered.

Our proposed problem is a special case of the general problem with piecewise concave production and holding costs defined by Swoveland in [5]. Swoveland showed that there is an optimal solution which is a sequence of time intervals with starting and ending inventories either 0 or equal to the inventory bound, such that within each interval there is at most one order not equal to a production breakpoint level. In [5] the author used this to propose a pseudo-polynomial algorithm for capacitated production, bounded inventory and piecewise concave production and holding costs, which also works for our problem. In order to reduce the running time to polynomial, we essentially prove in Section 3 that there is an optimal solution which is a sequence of regeneration intervals with 0 starting and ending inventories, each comprised by at most two intervals of the type defined in [5]. In fact the structure of our regeneration interval is more specific than that, cf. Lemmas 1-3 for the exact description. We use this property to design in Section 4 an efficient $O(n^2)$ Dynamic Programming (DP) algorithm for the case of bounded inventory, piecewise linear production cost function with one breakpoint, and linear holding cost. We show that by adopting a just-in-time policy. Finally, we show in Section 5 that when the problem is extended to two items with separate budgets, it becomes NP-complete.

2. Problem definition

A buyer has to order a number of units of a product, in order to satisfy n consecutive demand requests $d_1, d_2, \ldots, d_n$. At time period t, the buyer orders x_t units, so that, together with the remaining inventory I_{t-1} at the end of time $t-1$, demand d_t is satisfied, i.e., $x_t + I_{t-1} \geq d_t$. The new remaining inventory is $I_t = x_t + I_{t-1} - d_t$. Initially, the available inventory of the product is I_0. The buyer has an inventory capacity $B(t)$ at time t, which cannot be exceeded at any time, i.e., $I_t \leq B(t),\ t = 1, \ldots, n$.

The product pricing is as follows: Given threshold Q, if $x_t < Q$ then the buyer pays a price $p_1(t)$ per unit, otherwise the buyer pays a price $p_2(t)$ per unit, where $p_1(t) \geq p_2(t)$. We assume that unit prices are non-increasing as a function of time, i.e., $p_1(t) \geq p_1(t'),\ p_2(t) \geq p_2(t'),\ t < t'$. There is also a holding cost $h(t)$ per inventory unit for time t. Given Q, I_0, and $p_1(t),\ p_2(t),\ B(t),\ d_t,\ h(t),\ t = 1, 2, \ldots, n$, we would like to compute orders $x_t,\ t = 1, 2, \ldots, n$ that respect the inventory capacity constraint, and minimize the buyer's total cost.

In what follows, an *expensive* order at time t is an order priced at $p_1(t)$, and a *cheap* order is an order priced at $p_2(t)$. For $i \leq j$, $d_{i,j}$ denotes $\sum_{k=i}^{j} d_k$.

3. Structural properties of an optimal solution

Let OPT be an optimal solution. In this section we show that there is an optimal solution of a special structure, which will allow its fast computation. Similarly to [2], we define a *regeneration interval* (i, j) of OPT (with $i \leq j$) to be a sequence of consecutive time periods $[i, i+1, \ldots, j]$ with inventories $I_k > 0$ for all $i \leq k \leq j-1$, and $I_{i-1} = 0$ or $i = 1$, and $I_j = 0$ or $j = n$. In other words, a regeneration interval is the time between two times i and j which starts and ends with 0 inventory (except for the interval with $i = 1$, or $j = n$), and maintains a non-zero inventory everywhere in between. Note that if $I_k > 0,\ \forall 1 \leq k < n$, then there is only one interval $(1, n)$.

Starting from OPT, we derive an optimal solution with the following structural properties:

Lemma 1. *There is an optimal solution where, for any regeneration interval (i, j), an expensive order can only occur at time j. If x_j is expensive, then $x_j = d_j - I_{j-1}$.*

Proof: Let (i, j) be an interval of OPT with an expensive order x_k at time $i \leq k < j$. Since there are leftover inventories $I_k > 0, I_{k+1} > 0, \ldots, I_{j-1} > 0$, we can reduce order x_k by 1, reduce all these inventories by 1, and increase order x_j by 1, without harming the cost or the feasibility of OPT (x_k was already an expensive order below Q, x_j is increased, prices are non-increasing, and the intermediate inventories are reduced, thus not increasing the holding costs). We continue this process of reducing x_k by 1, until either $x_k = 0$ or $I_l = 0$ for some $k \leq l < j$. In the second case, interval (i, j) is now split into two or more smaller intervals, and we can repeat the process with the new set of intervals, until we obtain the optimal solution claimed by the lemma.

If x_j is expensive, then if $j < n$, by the definition of an interval, we have $I_j = x_j + I_{j-1} - d_j = 0$; if $j = n$, then OPT uses the available inventory I_{n-1} and if $d_n > I_{n-1}$ orders an additional $x_n = d_n - I_{n-1}$ or Q units (whichever is cheaper). □

Lemma 2. *There is an optimal solution satisfying the property of Lemma 1, which satisfies the following: For any regeneration interval (i,j), let k be the total number of cheap orders occurring in it. Then the first $k-1$ orders are for exactly Q units, and the k-th order is for at least Q units.*

Proof: Let OPT be an optimal solution satisfying the property of Lemma 1, and let (i,j) be any of its intervals.

Lemma 1 implies that the only expensive order can occur at time j, so the previous k orders are all cheap. Let $x_{l_1}, x_{l_2}, \ldots, x_{l_k}$ be these orders $l_k \leq j$, occurring at times $l_1, l_2, \ldots, l_k$, respectively. Let $l_m < l_k$ be the last of these times before the last, for which $x_{l_m} > Q$ (i.e., $x_{l_{m+1}} = \cdots = x_{l_{k-1}} = Q$). Then, since $I_{l_m} > 0, I_{l_m+1} > 0, \ldots, I_{l_k-1} > 0$, we can reduce x_{l_m} by 1 without changing its pricing, reduce inventories $I_{l_m}, I_{l_m+1}, \ldots, I_{l_k-1}$ by 1, and increase x_{l_k} by 1, without violating the feasibility of the solution or increasing its cost (both x_{l_m} and x_{l_k} remain cheap, the prices are non-increasing, and the holding cost is always non-increasing when postponing orders for later). We continue repeating this process, until either $x_{l_m} = Q$, or at least one of the inventories $I_{l_m}, I_{l_m+1}, \ldots, I_{l_k-1}$ becomes 0. In the first case, we repeat this process, but now concentrating on a cheap order bigger than Q that is earlier than l_m; in the second case, interval (i,j) is split into two or more (smaller) intervals, and we can repeat the process with the new set of intervals. □

In order to show that there is an optimal solution making cheap orders just-in-time, we will need the following definition.

Definition 1. *Let (i,j) be a regeneration interval of an optimal solution adhering to the structure of Lemma 2, with its last cheap order at time $i \leq t < j$. A* feasible time *t' is any time $t < t' \leq j$ where a cheap just-in-time order can be placed, without increasing I_j, and without violating inventory bounds, after setting $x_t := Q$. More specifically, given inventory I_{t-1} and $x_t := Q$, let $t < k \leq j$ be the time when both*

$$I_{t-1} + Q \geq d_{t,k-1} \text{ and } I_{t-1} + Q < d_{t,k}$$

hold. Then, if a cheap order can be placed at k without increasing I_j, and all inventory bounds $B(l)$, $k \leq l \leq j$, are respected, we set $t' := k$.

Note that in Definition 1, it may be the case that the latest possible time k one can afford to wait until forced to place a cheap order just-in-time, may not be a feasible time, e.g., if $Q - d_k > B(k)$ and, therefore, there is not enough inventory space to accommodate a cheap order at k. If this is the case, then placing a cheap order for Q units earlier than k will still violate the inventory constraint at k, i.e., if k cannot be a feasible time, no time $t < t' < k$ can be a feasible time.

Lemma 3. *There is an optimal solution satisfying the properties of Lemmas 1 and 2, which, for any regeneration interval (i,j), also satisfies the following: If there are cheap orders, then every cheap order occurs at the furthest possible time from the previous one (or from time i, for the first cheap order); also, for $j < n$, there is no feasible time for a new cheap order between the last cheap order and j.*

Proof: Let OPT be an optimal solution complying with Lemmas 1 and 2.

First we prove the lemma for intervals (i,j) with $i > 1$. For simplicity, we assume that $d_i > 0$. Since $I_{i-1} = 0$, $x_i > 0$ in OPT. If this order is the expensive one, or the only cheap one, then there is nothing more to prove. Otherwise, since there is more than one cheap order, so $x_i = Q$ (Lemma 2). Let $k > i$ be the latest time when there must be a cheap order, otherwise some demands will not be satisfiable, i.e., k is the latest possible time in the sense of Definition 1 (since there is more than one cheap order, k is well-defined). If there are no other cheap orders by OPT in the times between i and k, then notice that OPT has to order cheaply at k, and we repeat our arguments here with k playing the role of i. Otherwise, let $x_{l_1} = x_{l_2} = \ldots = x_{l_m} \geq Q$ be these cheap orders with $i < l_1 < \ldots < l_m < k$. Then we can place all these orders (together with any preexisting order $x_k > 0$ in OPT) cheaply at time k, since the inventory I_{k-1} is now reduced by $\sum_{g=1}^{m} x_{l_g}$, and the other intermediate inventories can only decrease. By the definition of k, no demand is left uncovered by this transfer of orders. If after the transfer inventory I_t drops to zero for some $i < t \leq k-1$ we split (i,j) into smaller regeneration intervals, like in Lemma 1. Moreover, similarly to the previous lemma the cost does not increase. By this process, we get another optimal solution OPT', with its first two cheap orders at times i, k, with $x_i = Q$ and $x_k \geq Q$, and which can be brought into the format of Lemma 2. Then we repeat the argument above (with k now playing the role of i) repeatedly, until we obtain an optimal solution satisfying the property of the lemma.

For the first interval $(1,j)$, we can define time k exactly as before, while it may be the case $x_1 = 0$ in OPT (OPT uses the initial inventory I_0 to satisfy the initial demands). Then the argument proceeds exactly as before.

For the second property, assume that (i,j) (with $j < n$) is a regeneration interval of an optimal solution that satisfies the first property of the lemma. Note that if there is a feasible time t' after the last cheap order x_t at time $t < t'$, then we can set $x'_t := Q$ and $x'_{t'} := x_t - Q$, without increasing the cost of the solution; this is because $x'_{t'}$ is cheap (by the feasibility of t'), and the inventories $I_t, I_{t+1}, \ldots, I_{t'-1}$ can only decrease, ensuring that holding costs do not increase. If there is more than one feasible time, let t' be the feasible time furthest from t. Then the new optimal solution also satisfies the first property of the lemma. As before, if the new inventory $I'_l = 0$ for some $t \leq l \leq t'-1$, then we split (i,j) into two new intervals $(i,l), (l+1,j)$

and repeat the process with the new set of intervals. Otherwise, we can repeat this process until there is no feasible time after the last cheap order in our final optimal solution. □

4. A Dynamic Programming algorithm

We use the properties of Lemmas 1-3 to compute an optimal solution, using Dynamic Programming (DP). Let $OPT(i)$ be the optimal cost for interval (i, n). Then

$$OPT(i) = \begin{cases} \min_{i \le k \le n}\{cost(i,k) + OPT(k+1)\}, & i \le n \\ 0, & i = n+1 \end{cases} \quad (1)$$

where $cost(i, j)$ is the minimum feasible cost in a regeneration interval (i, j).

The crucial idea for the algorithm is to notice that the cheap orders at level exactly Q made to achieve $cost(i, j)$ remain valid for the sequence of orders that achieve $cost(i, j+1)$, due to Lemma 3. This means that the addition of d_j affects only the last cheap order of $cost(i, j)$ in two possible ways: (i) either it also remains the last cheap order for $cost(i, j)$, but with a different number of units (and with a potential corresponding expensive order x_{j+1}), or (ii) there are new feasible time(s) created, so it breaks into one or more Q just-in-time cheap orders, according again to Lemma 3.

In what follows, we denote by $x^{(i,j)}(t)$ the order placed at time t for the solution that achieves $cost(i, j)$. A sequence of orders is *feasible for* (i, j) if it respects the inventory bounds, $I_j = 0$ and $I_k > 0$, $i \le k \le j-1$, and moreover complies with Lemma 3.

Preprocessing: The algorithm starts by calculating the following quantities:

- $d_{i,j} = \sum_{k=i}^{j} d_k$, $h_{i,j} = \sum_{k=i}^{j} h(k)$, and $H(i, j) = \sum_{k=i}^{j} h(k) d_{i,k}$ for all $1 \le i \le j \le n$. These quantities are needed in calculations performed by the algorithm, and they are done in $O(n^2)$ time.

- For all $1 \le i \le j \le n$, let $A(i, j)$ be the maximum order size at time i, that does not violate any inventory bounds $B(l)$, $i \le l \le j$ when we assume that $I_{i-1} = 0$ and no other orders happen between $i+1$ and j (inclusive). Then

$$A(i,i) = B(i) + d_i$$
$$A(i, j+1) = \begin{cases} A(i,j), & \text{if } A(i,j) - d_{i,j+1} \le B(j+1) \\ d_{i,j+1} + B(j+1), & \text{otherwise.} \end{cases}$$

Note that if there is inventory $I_{i-1} > 0$ coming into i, the value $A(i, j) - I_{i-1}$ gives an upper bound for order x_i, which ensures that x_i will not violate any inventory bounds in time interval (i, j). All values $A(i, j)$ can be calculated in total $O(n^2)$ time.

- For every i, the algorithm calculates a sequence $\mathcal{T}^i = \{t_1, t_2, \ldots, t_l\}$ of potential just-in-time order times, as well as inventory values $\hat{I}_{t_1-1}, \hat{I}_{t_2-1}, \ldots, \hat{I}_{t_l-1}$ as follows: The initial condition is that $t_1 = i$ and $\hat{I}_{t_1-1} = 0$. If $x_i = Q$ is infeasible, the sequence $\mathcal{T}^i$ is empty. To calculate t_s, $s \ge 2$ given inventory $\hat{I}_{t_{s-1}-1}$, we set $x_{t_{s-1}} := Q$ and scan times $t_{s-1}+1, t_{s-1}+2, \ldots$ until we reach the furthest time t_s when a cheap order can be placed with no demand being unsatisfied, and no inventory bound violated. The former can be checked by checking that $Q + \hat{I}_{t_{s-1}-1} \ge d_{t_{s-1},f}$, and the latter can be checked by checking that $Q \le A(t_{s-1}, f) - \hat{I}_{t_{s-1}-1}$, for all times $f = t_{s-1}+1, t_{s-1}+2, \ldots$ we scan. If there can be no cheap order at t_s, then t_{s-1} is the last element of sequence $\mathcal{T}^i$. Otherwise, we also calculate inventory value $\hat{I}_{t_s-1} = \hat{I}_{t_{s-1}-1} + Q - d_{t_{s-1},t_s-1}$, and continue to discover the next sequence element t_{s+1}. Clearly sequence $\mathcal{T}^i$ and values $\hat{I}$ can be calculated in $O(n-i)$ time, or $O(n^2)$ for all i. Inventory values $\hat{I}$ will be used in the calculation of functions f_1, f_2 below. In what follows i is fixed so we use $\mathcal{T}$ in place of $\mathcal{T}^i$.
Remark: The sequence $\mathcal{T}^i$ is the sequence of possible cheap order times that is promised by Lemma 3, i.e., for period $(i, j+1)$ all cheap orders happen at times that are a prefix of $\mathcal{T}^i$, all of them except possibly the last are exactly Q, and there is potentially one more order at $j+1$.

- For each s'th element of sequence $\mathcal{T}^i$, we calculate cost $C^i(t_s) = \sum_{w=1}^{s-1}\left(Qp_2(t_w) + \sum_{k=t_w}^{t_{w+1}-1} h(k)(\hat{I}_{t_w-1} + Q - d_{t_w,k})\right)$. This is the ordering and holding cost incurred in time period $(t_1, t_s - 1)$ when all orders at times $t_1, t_2, \ldots, t_{s-1} \in \mathcal{T}^i$ are exactly Q.

In the special case of $i = j$ with no capability of placing a cheap order at time i ($d_i < Q$), we set $x_i^{(i,i)} = d_i$, $cost(i, i) = p_1(i) \cdot d_i$, and $cost(i, i+k) = \infty$, $\forall 1 \le k \le n-i$; otherwise we set $x_i^{(i,i)} = d_i$, $cost(i, i) = p_2(i) \cdot d_i$.

Assuming that $cost(i, j) \ne \infty$ has been calculated, $cost(i, j+1)$ is calculated as follows: Let $t_{s-1}, t_s \in \mathcal{T}$ be the last two times in $\mathcal{T}$ before $j+1$ in $(i, j+1)$ (since $t_1 = i$, there is always at least one such time; if there is only t_1, then what follows is adjusted accordingly). There are orders of exactly Q units up to time t_{s-2}, and there are two possibilities for ordering after t_{s-2}: (i) Set $x_{t_{s-1}} := Q$ and order at t_s and at $j+1$, or (ii) order at t_{s-1} and at $j+1$. We can calculate (in constant time) the optimal orders $x_{t_s}^{(i,j+1)}, x_{j+1}^{(i,j+1)}$ and cost for (i) as follows (the calculation for (ii) is exactly the same):

We define the following functions

$$f_1(x_{t_s}, x_{j+1}) =$$
$$= p_2(t_s)x_{t_s} + p_1(j+1)x_{j+1} + \sum_{k=t_s}^{j} h(k)(\hat{I}_{t_s-1} + x_{t_s} - d_{t_s,k})$$
$$= (p_2(t_s)+h_{t_s,j})x_{t_s} + p_1(j+1)x_{j+1} + \hat{I}_{t_s-1}h_{t_s,j} - H(t_s, j) \tag{2}$$

and

$$f_2(x_{t_s}, x_{j+1}) =$$
$$= p_2(t_s)x_{t_s} + p_2(j+1)x_{j+1} + \sum_{k=t_s}^{j} h(k)(\hat{I}_{t_s-1} + x_{t_s} - d_{t_s,k})$$
$$= (p_2(t_s)+h_{t_s,j})x_{t_s} + p_2(j+1)x_{j+1} + \hat{I}_{t_s-1}h_{t_s,j} - H(t_s, j). \tag{3}$$

Given orders x_{t_s} and x_{j+1}, the functions f_1, f_2 give the cost incurred when x_{t_s} is cheap and x_{j+1} is cheap or expensive, respectively. Note that these functions can be computed in $O(1)$ time, since values $h_{t_s,j}, H(t_s, j)$, and $\hat{I}_{t_s-1}$ have been precomputed.

We solve the following two linear integer programs:

$$
\begin{aligned}
c_1 = \min f_1(x_{t_s}, x_{j+1}) + h(j+1)I_{j+1} + C^i(t_s) \text{ s.t.}\\
x_{t_s} + \hat{I}_{t_s-1} - d_{t_s,j} \geq 1\\
x_{t_s} + \hat{I}_{t_s-1} + x_{j+1} - I_{j+1} = d_{t_s,j+1}\\
I_{j+1} \begin{cases} = 0, & \text{if } j \leq n-2 \\ \leq B(j+1), & \text{if } j = n-1 \end{cases}\\
x_{t_s} \leq A(t_s, j) - \hat{I}_{t_s-1}\\
x_{t_s} \geq Q\\
0 \leq x_{j+1} \leq Q-1
\end{aligned}
$$

and

$$
\begin{aligned}
c_2 = \min f_2(x_{t_s}, x_{j+1}) + h(j+1)I_{j+1} + C^i(t_s) \text{ s.t.}\\
x_{t_s} + \hat{I}_{t_s-1} - d_{t_s,j} \geq 1\\
x_{t_s} + \hat{I}_{t_s-1} + x_{j+1} - I_{j+1} = d_{t_s,j+1}\\
I_{j+1} \begin{cases} = 0, & \text{if } j \leq n-2 \\ \leq B(j+1), & \text{if } j = n-1 \end{cases}\\
x_{t_s} \leq A(t_s, j) - \hat{I}_{t_s-1}\\
x_{t_s} = Q\\
x_{j+1} \geq Q
\end{aligned}
$$

The objective is the cost f_1 or f_2, plus the holding cost for time $j+1$ and the costs incurred before t_s. The first constraint ensures that x_{t_s} satisfies all demands until time $j+1$ while always leaving an inventory (otherwise this cannot be a feasible regeneration period), the second that the order values as well as inventory I_{j+1} are consistent and satisfy all demands, and the remaining constraints ensure that the final inventory and the order values satisfy their inventory restrictions, as well as the cheap/expensive definitions. Note that both integer programs have variables $x_{t_s}, x_{j+1}, I_{j+1}$, and can be solved in $O(1)$ time. If c_1 is infeasible, then we set $c_1 = \infty$, and the same for c_2. Let $cost_1 = \min\{c_1, c_2\}$.

We repeat the same process for $x_{t_{s-1}}^{(i,j+1)}$ and $x_{j+1}^{(i,j+1)}$, and let $cost_2 = \min\{c'_1, c'_2\}$, where c'_1, c'_2 are the analogues of c_1 and c_2. If $cost_1 < cost_2$, then we return the calculated $x_{t_s}^{(i,j+1)}, x_{j+1}^{(i,j+1)}$, and $cost(i, j+1) = cost_1$. Otherwise, if $cost_2 = \infty$ return $cost(i, j+1) = \infty$, else return the calculated $x_{t_{s-1}}^{(i,j+1)}, x_{j+1}^{(i,j+1)}$, and $cost(i, j+1) = cost_2$.

Correctness: We show the following

Theorem 1. *The DP algorithm above computes an optimal solution.*

Proof: To prove that the algorithm is correct, it is enough to prove that it produces an optimal solution of the structure guaranteed by Lemma 3, i.e., it is enough to show that the $cost(i, j+1)$ calculated for regeneration period $(i, j+1)$ (and has the structure of Lemma 3) is optimal.

Lemma 3 implies that all cheap orders (except possibly x_{j+1}) in $(i, j+1)$ are done on the times of sequence $\mathcal{T}^i$, by the definition of the latter as the just-in-time sequence of potential cheap order times. Let t_{s-2}, t_{s-1}, t_s be the last members of $\mathcal{T}^i$ before $j+1$. We observe that, in $(i, j+1)$, the cheap orders up to time $t_{s-2} \in \mathcal{T}^i$ (inclusive) are orders of exactly Q units. This is due to the fact that if t_{s-2} were the last cheap order time for $(i, j+1)$, then $x_{t_{s-2}} \geq 2Q$ to cover d_{t_{s-2},t_s}, since orders of size Q had to be placed at t_{s-2}, t_{s-1} to cover these demands by just-in-time sequence $\mathcal{T}^i$. As a result, we can set $x_{t_{s-2}} := Q$ and move the rest of the units to a cheap order at t_{s-1}, without increasing the cost. Hence, it is enough for the algorithm to check the pairs of times $t_{s-1}, j+1$ and $t_s, j+1$ for the calculation of the best last two orders, which will also result in the optimal $cost(i, j+1)$. □

Complexity: First observe that the calculation of sequence $\mathcal{T}^i$ can be done in time $O(n-i)$, since all its orders can all be calculated by scanning interval (i, n) once. In every interval (i, j) we need to calculate orders for 3 times, namely j and the last 2 times of $\mathcal{T}^i$ contained in (i, j), and this can be done in $O(1)$ time. All other times of $\mathcal{T}^i$ before j get an order of exactly Q. Also note that there is no need to record separately all orders placed in all intervals $(i, j), i \leq j \leq n$, since all we need to record for each one is the last time of $\mathcal{T}^i$ utilized by (i, j), the last cheap order time and size, and $x^{(i,j)}(j)$. With this information, we can reconstruct the orders that incur $cost(i, j)$. Hence, the overall running time and space needed for all i is $O(n^2)$.

Collecting all running times and space needed by the different algorithm components we get

Theorem 2. *The DP algorithm takes $O(n^2)$ time and space.*

5. The case of two items

The two-items decision problem we examine is defined as follows: We have two items, *black* and *red,* which can be stored in an inventory of total capacity I. The pricing scheme for black is given by fixed unit prices $b_1 > b_2$, and threshold Q_B above which price b_2 applies. The pricing scheme for red is given by fixed unit prices $r_1 > r_2$, and threshold Q_R. Given budget targets C_B, C_R for black and red, as well as a sequence of n black and red demands, the question is whether they can be fulfilled, within the corresponding budgets, and without ever exceeding the inventory capacity I. There can also be initial inventories of the two items, but here we prove the NP-completeness of the problem, even in the case where these initial inventories are 0. More formally, the problem is defined as follows:

2-Bulk Ordering
Input: Sets $F = \{f_1, \ldots, f_n\} \subseteq \mathbb{Z}^+$ of black demand values and $G = \{g_1, \ldots, g_n\} \subseteq \mathbb{Z}^+$ of red demand values. Pricing schemes (b_1, b_2, Q_B) and (r_1, r_2, Q_R) for the black and red item respectively. Inventory size I. Nonnegative target costs C_B and C_R.
Question: Are there two n-vectors of orders for the black and the red item so that all demands are satisfied, the inventory capacity is never exceeded, the cost of black orders is at most C_B and the cost of red orders is at most C_R?

A pair of n-vectors of black and red orders is an *admissible sequence* for a 2-Bulk Ordering instance if the units purchased can be consumed at the time they were bought or stored so that all the demands at each time are covered and the inventory capacity I is never exceeded. We reduce from the Equipartition problem.

Equipartition
Input: Set $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{Z}^+$ with $\sum_{i=1}^n a_i = 2B$.
Question: Is there an index set $A' \subseteq [n]$, $|A'| = \frac{n}{2}$, such that $\sum_{i \in A'} a_i = B$?

It is well-known that Equipartition is NP-complete [17]. Given an Equipartition instance A, we construct an instance $\phi(A)$ of the 2-Bulk Ordering problem, that has an affirmative answer iff the Equipartition instance does. We now proceed to define $\phi(A)$. We set $b_1 = r_1 = p_1$ and $b_2 = r_2 = p_2$ for two positive values satisfying $p_1 > p_2$, i.e., both items have the same prices $p_1 > p_2$ (whose exact values will not matter in the reduction). We set

$$I = 5 \max_{i \in [n]} a_i \tag{4}$$

$$Q_B = Q_R = Q = 2I + 3B + 1 \tag{5}$$

$$C_B = C_R = p_1(\frac{n}{2}Q + 2B) + p_2(\frac{n}{2}Q + 2B). \tag{6}$$

The time horizon of $\phi(A)$ consists of n consecutive intervals $S_1, \ldots, S_n$, with each interval consisting of four time periods with respective demands $Q - (I - 2a_i), 0, I, 0$ for black, and $0, Q - (I - 2a_i), 0, I$ for red (cf. Figure 1). When a specific interval is implied from the context its time periods are numbered from 1 to 4.

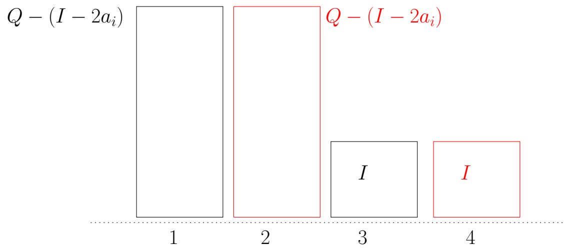

Figure 1: Demands for the four times of interval S_i, $i \in [n]$.

Lemma 4. *In any admissible sequence for the instance $\phi(A)$ at most one of black, red makes a cheap purchase in interval S_i, $i \in [n]$. This purchase happens at time 1 for black or at time 2 for red.*

Proof: Because of (5) a cheap purchase is possible only at time 1 or 2. In interval S_i, $i \in [n]$, each of the items requires $I - 2a_i$ space in the inventory for a cheap purchase at time 1 (for black) or 2 (for red). By (4) $2a_i < I - 2a_i$, hence there is not enough free inventory space for cheap purchases in both times 1 and 2. □

Lemma 5. *In any admissible sequence for $\phi(A)$ where the cost of black orders is at most C_B at least $n/2$ cheap orders for the black item are placed.*

Proof: Assume that in the admissible sequence $n/2 - x$ cheap orders have been placed for black for some $x \geq 1$. Let N_0 be the set of indices of the intervals in which there are no cheap black purchases, and $N_1 = [n] \setminus N_0$. Lemma 4 implies that $|N_1| = \frac{n}{2} - x$, and, therefore, $|N_0| = \frac{n}{2} + x$. Let ℓ_i be the total net amount of black item that can be fetched from the inventory to cover demand during interval S_i. This amount must have been purchased at some previous time, and at a unit price of at least p_2. Hence, the total cost $cost_B$ incurred for the black item is lower-bounded as follows:

$$\begin{aligned} cost_B \geq\ & p_1[(\frac{n}{2} + x)Q + \sum_{i \in N_0} 2a_i] - p_1 \sum_{i \in N_0} \ell_i + \\ & + p_2[(\frac{n}{2} - x)Q + \sum_{i \in N_1} 2a_i] + p_2 \sum_{i \in N_0} \ell_i \\ \geq\ & p_1(\frac{n}{2} + x)Q - p_1 \sum_{i \in N_0} \ell_i + p_2[(\frac{n}{2} - x)Q] + \\ & + 4Bp_2 + p_2 \sum_{i \in N_0} \ell_i \\ =\ & C_B + (p_1 - p_2)(xQ - 2B - \sum_{i \in N_0} \ell_i). \end{aligned} \tag{7}$$

Note that, for any admissible sequence, we can assume that items are never purchased expensively, in order to be stored in the inventory. Let the *excess* of a cheap purchase $x_t \geq d_t$ at some time t be the amount $x_t - d_t$ which will be stored in the inventory. Given an admissible sequence the associated red and black costs are uniquely determined by the order vectors and do not depend on when a stored unit is actually consumed. For the purposes of cost accounting we may allocate the excess units to satisfy demand at any time of our choice as long as all demand has been satisfied by the end of the time horizon and all the excess has been consumed. The *maximum* amount of a cheap purchase at time 1 of interval S_i, $i \in N_1$, is $Q-(I-2a_i)+I = Q+2a_i$, i.e., the excess is at most $2a_i$. The calculation in (7) allocated $Q+2a_i$ units to cover the black demand in interval $i \in N_1$. This way all potential excess has been consumed and it follows that $\sum_{i\in N_0} \ell_i = 0$. But in that case, (5) and (7) imply that $cost_B > C_B$, a contradiction to the assumption that the black cost is at most C_B. Therefore $x = 0$. □

Similarly we can prove:

Lemma 6. *In any admissible sequence for $\phi(A)$ where the cost of red orders is at most C_R at least $n/2$ cheap orders for the red item are placed.*

Lemma 7. *The* Equipartition *instance A is a* `Yes`*-instance iff the* 2-Bulk Ordering *instance $\phi(A)$ is a* `Yes`*-instance.*

Proof: Let the Equipartition instance have a solution given by $A' \subseteq [n]$, $|A'| = n/2$ such that $\sum_{i\in A'} a_i = B$. There is a corresponding admissible sequence for the 2-Bulk Ordering instance, in which a cheap red purchase of $Q + 2a_i$ units takes place at every interval S_i, $i \in A'$, and a cheap black purchase at the amount of $Q+2a_i$ takes place for every S_i, $i \in [n] \setminus A'$. The sequence meets the cost targets for both items.

Conversely, assume that the 2-Bulk Ordering instance is a `Yes`-instance. Let T_B (T_R) be the indices of the intervals in which a black (red) cheap purchase occurs. By Lemmas 4, 5 and 6 $|T_B| = |T_R| = n/2$. The excess of a cheap purchase (either black at time 1 or red at time 2) is at most I. We can lower bound the cost by assuming that it is always I.

We allocate all potential excess of a cheap purchase at interval $S_i \in T_B$ ($S_i \in T_R$) to cover the demand at time 3 (resp. 4). Therefore the total black item amount bought expensively is at least $\frac{n}{2}Q + \sum_{i\in T_R} 2a_i$, and the total red item amount bought expensively is at least $\frac{n}{2}Q + \sum_{i\in T_B} 2a_i$. It holds that $\sum_{i\in T_R\cup T_B} 2a_i = \sum_{i\in[n]} 2a_i = 4B$. If $\sum_{i\in T_R} 2a_i = 2B + y$ for some positive y, the cost paid for the black item is $C_B + (p_1 - p_2)y > C_B$, a contradiction, and similarly for the red. Therefore $\sum_{i\in T_R} 2a_i \leq 2B$ and $\sum_{i\in T_B} 2a_i \leq 2B$, which imply that $\sum_{i\in T_R} 2a_i = \sum_{i\in T_B} 2a_i = 2B$, i.e., the Equipartition instance has a feasible solution. □

Lemma 7, together with the easy fact that 2-Bulk Ordering is in NP, imply the following

Theorem 3. *The* 2-Bulk Ordering *problem is* NP*-complete even when both items have the same pricing scheme (p_1, p_2, Q).*

A pair of values (x_1, x_2) *dominates* another pair (y_1, y_2) if $x_1 \leq y_1$ and $x_2 \leq y_2$. The *cost pair* of an admissible sequence σ for a 2-Bulk Ordering instance is the pair of black and red costs incurred by σ. The *trade-off* or *Pareto curve* for 2-Bulk Ordering is the set of all admissible sequences for which the corresponding cost pair is not dominated by the cost pair of any other (admissible) sequence. Observe that in the proof of Lemma 7 we actually established that the Equipartition instance is a `Yes`-instance iff there is an admissible sequence for the corresponding 2-Bulk Ordering instance where the cost for black is exactly C_B and for red is exactly C_R. The following corollary is immediate and suggests the intractability of identifying points on the Pareto curve.

Corollary 1. *Given an instance of* 2-Bulk Ordering *and a pair of cost targets (C_1, C_2) it is* NP*-complete to determine whether (i) there is an admissible sequence whose cost pair dominates (C_1, C_2) and (ii) there is an admissible sequence on the Pareto curve with cost pair (C_1, C_2).*

Acknowledgement. The authors thank an anonymous reviewer for comments that helped to substantially improve the paper.

[1] N. Brahimi, N. Absi, S. Dauzére-Pérés, A. Nordli, Single-item dynamic lot-sizing problems: An updated survey, European Journal of Operational Research, 263 (3) (2017) pp. 838-863.

[2] E. Koca, H. Yaman, M. Selim Aktürk, Lot sizing with piecewise concave production costs, INFORMS Journal on Computing 26 (4) (2014) pp. 767-779.

[3] H. M. Wagner, T. M. Whitin, Dynamic version of the economic lot size model, Management Science, 5 (1) (1958) pp. 1007-1013.

[4] Y. Malekian, S. Hamid Mirmohammadi, M. Bijari, Polynomial-time algorithms to solve the single-item capacitated lot sizing problem with a 1-breakpoint all-units quantity discount, Computers & Operations Research, 134 (2021).

[5] C. Swoveland, A deterministic multi-period production planning model with piecewise concave production and holding-backorder costs, Management Science, 21 (9) (1975) pp. 89-96.

[6] S. F. Love, Bounded production and inventory models with piecewise concave costs, Management Science, 20 (3) (1973) pp. 313-318.

[7] J. Ou, A polynomial time algorithm to the economic lot sizing problem with constant capacity and piecewise linear concave costs, Operations Research Letters, 45 (5) (2017) pp. 493-497.

[8] J. Ou, Improved exact algorithms to economic lot-sizing with piecewise linear production costs, European Journal of Operational Research, 256 (3) (2017) pp. 777-784.

[9] A. Federgruen, C. Lee, The dynamic lot size model with quantity discount, Naval Research Logistics (NRL), 37 (5) (1990) pp. 707-713.

[10] G. R. Bitran, H. H. Yanasse, Computational complexity of the capacitated lot size problem, Management Science, 28 (10) (1982) pp. 1174-1186.

[11] C. P. M. van Hoesel, A. P. M. Wagelmans, An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities, Management Science, 42 (1) (1996) pp. 142-150.

[12] T. Liu, Economic lot sizing problem with inventory bounds, European Journal of Operational Research, 185 (1) (2008) pp. 204-215.

[13] M. Önal, W. van den Heuvel, T. Liu, A note on "The economic lot sizing problem with inventory bounds," European Journal of Operational Research, 223 (1) (2012) pp. 290-294.

[14] C. P. M. van Hoesel, A. P. M. Wagelmans, Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems, Mathematics of Operations Research, 26 (2) (2001) pp. 339-357.

[15] C. T. Ng, Mikhail Y. Kovalyov, T. C .E. Cheng, A simple FPTAS for a single-item capacitated economic lot-sizing problem with a monotone cost structure, European Journal of Operational Research, 200 (2) (2010) pp. 621-624.

[16] J. Fan, G. Wang, Joint optimization of dynamic lot and warehouse sizing problems, European Journal of Operational Research, 267 (3) (2018) pp. 849-854.

[17] M. R. Garey, D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman and Company, (1979).