An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities

C.P.M. van Hoesel * A.P.M. Wagelmans †

Revised June 1994

Abstract

We develop an algorithm that solves the constant capacities economic lot-sizing problem with concave production costs and linear holding costs in $O(T^3)$ time. The algorithm is based on the standard dynamic programming approach which requires the computation of the minimal costs for all possible subplans of the production plan. Instead of computing these costs in a straightforward manner, we use structural properties of optimal subplans to arrive at a more efficient implementation. Our algorithm improves upon the $O(T^4)$ running time of an earlier algorithm.

Key words: Economic lot-sizing, complexity.

AMS Subject classification: 90B.

*Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: HOESEL@BS.WIN.TUE.NL; University of Limburg, P.O. Box 616, 6200 MD Maastricht, The Netherlands. E-mail: HOESEL@KE.RULIMBURG.NL

†Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands. E-mail: WAGELMAN@OPRES.FEW.EUR.NL

1 Introduction

In the single-item capacitated economic lot-sizing problem there is demand for a single item in T consecutive periods. The demand in a certain period may be satisfied by production in that period or from inventory that has been produced in earlier periods. It is assumed that there is no inventory at the beginning of period 1 and that no inventory should be left at the end of period T. Furthermore, capacity constraints on the production levels have to be taken into account. The total costs associated with a production plan depend on the production and inventory levels. A fixed set-up cost is incurred in a certain period whenever production takes place. In addition there are production costs which are a function of the production level. Finally, there are holding costs, which are a function of the inventory level at the end of the period. The objective is to find a feasible production plan that minimizes total costs.

In most models that have been studied in the literature, the cost functions are assumed to be concave or linear. Under these assumptions, many uncapacitated models are polynomially solvable. For instance, if all cost functions are linear, then the uncapacitated version of the above problem is solvable in $O(T \log T)$ time (cf. Aggarwal and Park [1], Federgruen and Tzur [5] and Wagelmans et al. [11]). Polynomial algorithms also exist for many other uncapacitated lot-sizing problems with linear costs (cf. Aggarwal and Park [1] and Van Hoesel et al. [8]). The uncapacitated problem with concave production and holding costs is solvable in $O(T^2)$ time (cf. Veinott [10]).

For capacitated problems the situation is quite different. Florian et al. [7] and Bitran and Yanasse [2] have shown that the single item capacitated economic lot-sizing problem is NP-hard, even in many special cases. Bitran and Yanasse also designed a classification scheme for capacitated lot-sizing problems with linear production and holding costs. They use the four field notation $\alpha/\beta/\gamma/\delta$, where α, β, γ and δ represent the set-up cost, unit holding cost, unit production cost and capacity type, respectively. Each of the parameters α, β and γ can take on one of the values *G, C, ND, NI* or Z. G means that the parameter follows an arbitrary pattern over time, whereas *C, ND, NI* and Z indicate constant, non-decreasing, non-increasing and zero parameter values, respectively. δ can take on the values *G, C, ND* or *NI*; in case there are no capacity restrictions, the fourth field is omitted.

A very successful DP approach to solve the most general problem, $G/G/G/G$, has recently been proposed by Chen et al. [3]. We also refer to that paper for a discussion of other work on NP-hard versions of the capacitated economic lot-sizing problem.

With respect to polynomially solvable special cases of the capacitated economic lot-sizing problem, the following results are known. Bitran and Yanasse showed that *ND/Z/ND/NI* and $C/Z/C/G$ can be solved in $O(T)$ respectively $O(T \log T)$ time. Chung and Lin [4] gave an $O(T^2)$ algorithm for *NI/G/NI/ND* and an $O(T^4)$ algorithm for $G/G/G/C$ was presented by Florian and Klein [6]. The latter algorithm also solves the more general constant capacity problem in which the cost functions are concave instead of linear. Pochet and Wolsey [9] consider the related problem in which multiple batches of equal capacity are available, each requiring a set-up cost. They solve this problem in $O(T^3)$ time.

In this paper we will show that when the production costs are concave and the holding costs are linear, it is possible to solve the economic lot-sizing problem with constant capacities in $O(T^3)$ time. Hence, for this case we improve upon the Florian–Klein algorithm.

This paper is organized as follows. In Section 2 we introduce some notation. Section 3 contains a description of a greedy algorithm for solving a basic subproblem. In Section 4 the actual algorithm is described. Section 5 contains conclusions and some remarks.

2 Preliminaries

We will use the following notation:

T : the length of the planning horizon;
C : the production capacity in each period.

Furthermore, for each period $t \in \{1, \ldots, T\}$:

d_t: the demand in t;
x_t: the production level in t;
I_t: the inventory level at the end of t ($I_0 = 0$);
f_t: the set-up cost in t;
h_t: the unit holding cost in t;
$p_t(x_t)$: the production costs in t, a concave function of x_t.

The cumulative demand of a set of consecutive periods $\{s, \ldots, t\}$ ($1 \leq s < t \leq T$) will be denoted by $d_{s,t} = \sum_{\tau=s}^{t} d_\tau$.

Without loss of generality we may assume:

(a) For each period $t : d_t \leq C$. If this is not the case, we can move the excess demand in t to the preceding period $t-1$ without changing the set of feasible solutions.

(b) The unit holding costs are all equal to zero. If this is not the case, then an equivalent problem is obtained by omitting the holding costs and redefining the production costs as $\tilde{p}_t(x_t) = p_t(x_t) + \sum_{i=t}^{T} h_i x_t$ (cf. Wagelmans et al. [11]). Note that we can achieve this only when the original holding costs are linear.

For notational convenience, we let $cf(t)$ denote the cost of producing at full capacity in period t, i.e., $cf(t) = f_t + \tilde{p}_t(C)$ ($t \in \{1, \ldots, T\}$).

We call production in a period t *fractional* if it is between 0 and C, i.e., $0 < x_t < C$. Florian and Klein [6] have shown that there exists an optimal schedule such that between any pair of fractional production periods there is at least one period with zero inventory. This property is often referred to as the *fractional production property*. It also holds in case of general capacities. For any feasible solution, we define a *subplan* (t_1, t_2) ($1 \leq t_1 \leq t_2 \leq T$) as a set of consecutive periods, starting with t_1 and ending with t_2, such that at most one period has fractional production and $I_{t_1-1} = I_{t_2} = 0$. (Note that our definition of subplan is more general than the usual definition in which inventories of intermediate periods $t_1, \ldots, t_2 - 1$ are required to be strictly positive.) It follows from the fractional production property that we only need to consider feasible solutions that can be subdivided into subplans. This suggests an approach in which we first determine optimal solutions for all subplans and then choose the best combination of subplans which constitute a complete solution. In the next section we will present an algorithm for finding an optimal solution for a given subplan.

3 Greedy algorithm

Consider a fixed subplan (t_1, t_2) for which we want to find a minimum cost solution. In case $d_{t_1,t_2} = KC$ for some integer K, any feasible solution has only full production periods, namely exactly K. Hence, finding a minimum cost solution for the subplan boils down to determining in which K periods full production should take place. In case cumulative demand is not a multiple of C, i.e., if $d_{t_1,t_2} = f + KC$ for some integer K and f such that $0 < f < C$, then any feasible solution has K full production periods and a fractional period in which the production level equals f. Suppose that we fix the fractional production period, then the problem is again to determine an optimal set of full production periods. In the remainder of this paper we will focus on the case in which the subplan contains a fractional period, because this problem is clearly as least as hard as the problem without a fractional period.

We can restrict the fractional production f to periods t with $d_{t,t_2} \geq f$, since fractional production in later periods will lead to positive ending inventory in period t_2, contradicting the definition of a subplan. Therefore, we define t_{max} to be the latest period t such that $d_{t,t_2} \geq f$. Similarly, there is an earliest possible fractional period. If $d_{t_1,t} > (t-1)C + f$, then the periods t_1 through t must be full capacity production periods. Therefore, we define t_{min} as the first period t for which $d_{t_1,t} \leq (t-1)C + f$.

Suppose the fractional period is fixed to $t \in \{t_{min}, \ldots, t_{max}\}$ and let $P(t)$ denote the corresponding problem of determining optimal full production periods. We introduce a function $A(\tau)$ $(\tau \in \{t_1, \ldots, t_2\})$ which denotes the minimum number of full production periods in $\{t_1, \ldots, \tau\}$ in any feasible solution of $P(t)$.

$$A(\tau) = \begin{cases} \lceil \frac{d_{t_1,\tau}}{C} \rceil & \text{for } \tau < t \\ \\ \lceil \frac{d_{t_1,\tau} - f}{C} \rceil & \text{for } \tau \geq t \end{cases}$$

A solution of $P(t)$ is feasible if and only if for any $\tau \in \{t_1, \ldots, t_2\}$ the number of full production periods in $\{t_1, \ldots, \tau\}$ is at least $A(\tau)$. The function A is integral and monotonically nondecreasing for the periods $\{t_1, \ldots, t-1\}$ and $\{t, \ldots, t_2\}$. Moreover, $A(t-1) \leq A(t) + 1$. Note that A can take on the values $\{0, \ldots, K\}$. Define for all $k \in \{1, \ldots, K\}$ the period w_k as the earliest period τ for which $A(\tau) = k$ holds. The following is obvious.

Feasibility condition
A production schedule is feasible, i.e., $I_\tau \geq 0$ for all $\tau \in \{t_1, \ldots, t_2\}$, if and only if for every period w_k $(1 \leq k \leq K)$, there are at least k production periods in $\{t_1, \ldots, w_k\}$.

This period w_k is called a *choice period* because it forces us to choose a k-th full production period in the set $\{1, \ldots, w_k\}$. We choose this production period as specified below.

Greedy algorithm
Start with the production plan in which only the fractional production takes place in period t. This period is not available for full production. The K full production periods are chosen as follows. Consider the choice periods w_k, $k \in \{1, \ldots, K\}$, in increasing order. The cheapest available period τ in the set $\{t_1, \ldots, w_k\}$ is chosen as production period, i.e., $cf(\tau)$ is minimal among the available periods $\tau \in \{t_1, \ldots, w_k\}$. If necessary, break ties by choosing the earliest period.

Example

We consider subplan $(1,7)$. The capacity C is 5 units. The cumulative demand is 18, and therefore $K=3$ and $f=3$. The other data are given in Table 1, where p_τ denotes the unit production costs in period τ, i.e., production costs are linear.

Table 1: Data of example

τ	1	2	3	4	5	6	7
d_τ	0	4	2	1	4	5	2
f_τ	4	7	5	8	7	7	5
p_τ	3	1	0	1	2	1	1
$cf(\tau)$	19	12	5	13	17	12	10

Let period 4 be the fractional period, i.e., $x_4 = 3$. Then the calculations of the greedy algorithm are shown in Table 2.

Table 2: Results of greedy algorithm

τ	1	2	3	4	5	6	7
$d_{1,\tau}$	0	4	6				
$d_{1,\tau} - f$				4	8	13	15
$A(\tau)$	0	1	2	1	2	3	3
choice	n	y	y	n	n	y	n
full prod.	n	y	y	n	n	y	n

The choice periods are 2, 3 and 6. In this example, the full production periods coincide with the choice periods. However, this is not the case in general, as can be seen by swapping the cost structure of periods 1 and 2. This would leave the choice periods unchanged, but period 1 would be chosen as a full production period instead of period 2. Finally, we note that the total cost of this plan is $12 + 5 + (8 + 3 * 1) + 12 = 40$.

The definition of choice period w_k ensures that the k-th full production period is chosen from a set of available periods which is as large as possible. The greedy aspect of the algorithm is that among all these available periods the cheapest one is chosen. Clearly, the greedy solution is feasible. Its optimality is proved next.

Let us first define an ordering on the feasible solutions of $P(t)$. Consider two feasible production plans S and S' and the first full production period in which they differ. If that period is earlier in S than it is in S', then solution S is called *lexicographically earlier* than solution S'. Note that the number of full production periods is equal to K in both solutions.

Lemma 1 *The greedy algorithm constructs the lexicographically earliest optimal production plan for $P(t)$.*

Proof. Let S be the lexicographically earliest optimal solution. Suppose it is not equal to the solution S_G created by the greedy algorithm.

Let $w_1, \ldots, w_K$ be the choice periods for the greedy algorithm, and let $\tau_1, \ldots, \tau_K$ be the respective full production periods chosen by the greedy algorithm. Let k be the smallest index such that τ_k is not in S. There is a period τ' in $\{t_1, \ldots, w_k\}$ that is a production period in S but not in S_G, because otherwise S would have less than k production periods in $\{t_1, \ldots, w_k\}$, violating the feasibility condition.

Consider the following cases.

(1) If $cf(\tau') < cf(\tau_k)$, then this contradicts the fact that the greedy algorithm chooses the cheapest available production period for w_k.

(2) If $cf(\tau') = cf(\tau_k)$ and $\tau' < \tau_k$, then this contradicts the fact that the greedy algorithm chooses the earliest period among the cheapest available ones.

The feasibility condition is also satisfied by the solution created from S by replacing τ' by τ_k as a production period. Therefore, we can conclude the following.

(3) If $cf(\tau') > cf(\tau_k)$, then the solution S can be improved.

(4) If $cf(\tau') = cf(\tau_k)$ and $\tau' > \tau_k$, then the solution S is not the lexicographically earliest optimal solution.

Hence, the assumption that $\tau' \neq \tau_k$ always leads to a contradiction. We conclude that S_G is equal to S, the lexicographically earliest optimal solution.

□

When referring to *the* optimal solution in the remainder of this paper, we will mean the lexicographically earliest optimal solution.

By solving $P(t)$ for all $t \in \{t_{min}, \ldots, t_{max}\}$, we can determine the optimal solution for subplan (t_1, t_2). Instead of solving each of these problems separately, we will propose an iterative algorithm in Section 4. This algorithm not only computes the optimal solutions of the problems $P(t)$ ($t \in \{t_{min}, \ldots, t_{max}\}$), but also the optimal solutions of the problems defined as follows. Let $t \in \{t_{min}, \ldots, t_{max}\}$, then $P(t)'$ is the problem of finding an optimal schedule when f units become available in period t completely for free, i.e., *without costing any money or capacity*. Clearly, a feasible solution for this problem corresponds to a choice of K full production periods. The only difference with problem $P(t)$ is that period t is now also available for full production (at cost $cf(t)$). It is easily seen that an optimal solution of $P(t)'$ can be found by applying the greedy algorithm. Note that the choice periods for $P(t)$ and $P(t)'$ are identical. Again, when referring to *the* optimal solution of $P(t)'$, we will mean the solution constructed by the greedy algorithm. The following properties play a key role in the algorithm.

Lemma 2 *Let $t \in \{t_{min}, \ldots, t_{max}\}$. The optimal solutions of $P(t+1)'$ and $P(t)'$ differ with respect to the full production periods in at most one period. Moreover, if there is a difference, then the optimal solution of $P(t)'$ is obtained from the optimal solution of $P(t+1)'$ by moving production from a period in $\{t_1, \ldots, t\}$ to a period in $\{t+1, \ldots, t_2\}$.*

Proof. We will prove that the solutions produced by the greedy algorithm in both problems differ in at most one production period, as described in the lemma.

The problems $P(t)'$ and $P(t+1)'$ differ with respect to function A only in period t: $A(t) = \lceil \frac{d_{t_1,t}}{C} \rceil$ in $P(t+1)'$ and $A(t) = \lceil \frac{d_{t_1,t}-f}{C} \rceil$ in $P(t)'$. Thus, $A(t)$ may be one unit less in $P(t)'$ than in $P(t+1)'$. This gives a possible difference in the set of choice periods, which can only occur if t is a choice period in $P(t+1)'$, say the k-th. In that case, the k-th choice period in $P(t)'$ may be a period u with $t < u < w_{k+1}$. All other choice periods are identical in both problems.

Clearly, because the first $k-1$ choice periods are identical, the first $k-1$ production periods chosen by the greedy algorithm will be the same for both problems. If all choice periods are identical in both problems, or if the greedy algorithm chooses the same production periods at t and u, then the optimal solutions do not differ. Hence, we only have to examine the case where the choices in t and u differ, say τ' is chosen at t in problem $P(t+1)'$, and τ'' is chosen at u in problem $P(t)'$.

By definition, τ' is the available cheapest period in $\{t_1, \ldots, t\}$, and τ'' is the cheapest available period in $\{t_1, \ldots, u\}$. Thus, $\tau'' \neq \tau'$ implies $\tau'' > t$ and $cf(\tau'') < cf(\tau')$.

We will show that in the remainder of the greedy algorithm the number of different production periods for both problems remains at most one, and that the difference is always as specified in the lemma.

As argued before, the choice periods after u are equal for both problems. Let those periods be $w_{k+1}, \ldots, w_K$, and consider the production period chosen at w_{k+1}.

(a) Suppose that τ' is the period chosen at w_{k+1} in problem $P(t)'$.
Because τ' is the cheapest available period up to w_{k+1} in $P(t)'$, it follows that τ'' is the cheapest available period up to w_{k+1} in $P(t+1)'$. Clearly, from w_{k+1} on the partial solutions are equal again.

(b) Suppose $\tau \neq \tau'$ is the period chosen at w_{k+1} in problem $P(t)'$.
τ is the cheapest available period up to w_{k+1}, and therefore $\tau > t$ (since τ' is the cheapest available period up to t). Moreover, in $P(t+1)'$ it is also the cheapest available period, unless τ'' is cheaper. However, which of these periods is chosen does not matter. In both cases the difference with respect to the partial solution of $P(t)'$ remains one period, either τ or τ'', and both are later than t.

If (a) occurs, then it follows immediately that the full production periods of the optimal solutions of $P(t)'$ and $P(t+1)'$ are equal. If (b) occurs, the above argument can be repeated for the later choice periods $w_{k+2}, \ldots, w_K$, and the lemma is proved.

□

If t is not chosen as a full production period in the optimal solution of $P(t)'$, then it is clearly optimal to take the same full production periods as the solution of $P(t)$. In case the optimal solutions are not equal, we have the following result which can be proved using similar arguments as in the proof of Lemma 2.

Lemma 3 *Let $t \in \{t_{min}, \ldots, t_{max}\}$ and suppose that t is a full production period in the optimal solution of $P(t)'$. Then the optimal solution of $P(t)$ differs from the optimal solution of $P(t)'$ only in the fact that the full production in t is reallocated.*

4 Global algorithm

The global algorithm for solving the lot-sizing problem consists of two phases. In the first phase we find the optimal solutions for the subplans. In the second phase these solutions are used to determine an optimal solution of the overall problem.

Phase 1: Find the minimum cost for all subplans (t_1, t_2), $1 \leq t_1 \leq t_2 \leq T$.

Phase 2: Find, in the directed graph with vertices $\{0, \ldots, T\}$ and arcs $(t_1 - 1, t_2)$, $1 \leq t_1 \leq t_2 \leq T$, the shortest path from vertex 0 to vertex T, where the length of arc $(t_1 - 1, t_2)$ is equal to the minimum cost of subplan (t_1, t_2).

Except for vertex 0, the vertices on the shortest path found in Phase 2 correspond to the last periods of the subplans which constitute an optimal production plan. Given the cost of each subplan, the second phase can be implemented in $O(T^2)$ time, since the graph is acyclic and the number of arcs is $O(T^2)$. Thus, Phase 2 is not the bottleneck of the algorithm. We will therefore focus on Phase 1. By considering all possible fractional production periods and using the greedy algorithm, a minimum cost solution for a given subplan can be found in $O(T^2)$ time. Because there are $O(T^2)$ possible subplans, this implies an $O(T^4)$ algorithm for Phase 1. We will give improvements that lead to an $O(T^3)$ implementation.

4.1 Iterative algorithm for Phase 1

We will show that the minimum cost of each subplan (t_1, t_2) $(1 \leq t_1 \leq t_2 \leq T)$ can be calculated in $O(T)$ amortized time. The algorithm consists of the following steps for each subplan.

Let t_{min}, t_{max} and the optimization problems $P(t)'$ and $P(t)$ $(t \in \{t_{min}, \ldots, t_{max}\})$ be as defined in the previous section.

Initialization
Compute the optimal solution of $P(t_{max})'$. This solution is also optimal for $P(t_{max})$.

Iterations
For t from $t_{max} - 1$ down to t_{min} do

Step 1: Determine the optimal solution of $P(t)'$ from the optimal solution of $P(t+1)'$.

Step 2: Determine the optimal solution of $P(t)$ from the optimal solution of $P(t)'$.

Example (continued)
Consider again the subplan $(1, 7)$ with $C = 5$, $f = 3$, $K = 3$ and the data in Table 1. Note that $t_{min} = 1$ and $t_{max} = 6$. Table 3 shows in the second column the choice periods

for varying fractional periods t. The optimal full production periods for $P(t)'$ and $P(t)$ are shown in the third and fourth column, respectively. The last column gives the optimal value of $P(t)$. Hence, in the example, it is optimal to have the fractional production in period 4 and full production in the periods 2, 3 and 6.

Table 3: Optimal solutions for varying fractional periods

t	choice	$P(t)'$	$P(t)$	cost
6	2 3 5	2 3 5	2 3 5	44
5	2 3 6	2 3 6	2 3 6	42
4	2 3 6	2 3 6	2 3 6	40
3	2 4 6	2 3 6	2 4 6	42
2	2 4 6	2 3 6	1 3 6	46
1	2 4 6	2 3 6	2 3 6	42

The row for $t = 6$ corresponds to the initialization of the iterative algorithm. The other rows correspond to the iterations. For each row of these rows, first the solution in the third column is computed from the solution immediately above it. Then this solution is used to compute the solution in the fourth column. Note that Lemma 2 is reflected by the fact that the difference between two consecutive rows in the third column is at most one period. Furthermore, the third and fourth column differ on the same row in at most one period. This reflects Lemma 3. On the other hand, as can be seen in this example, in the fourth column the difference between two consecutive rows may be two periods. This is exactly why we introduced the problems $P(t)'$. Instead of trying to derive an optimal solution of $P(t)$ directly from an optimal solution of $P(t+1)$, which may be complicated, we perform two relatively simple steps involving $P(t+1)'$ and $P(t)'$.

We will now show how the initialization and the iterations can be implemented in linear amortized time.

4.2 Implementation of initialization

The initialization can be carried out simultaneously for all subplans (t_1, t_2) with t_1 fixed and $t_2 \in \{t_1, \ldots, T\}$ by using the following lemma.

Lemma 4 *Let $1 \leq t_1 \leq t_2 \leq T - 1$. Consider the optimal solutions for subplans (t_1, t_2) and $(t_1, t_2 + 1)$, where the fractional periods are the last production periods. Then the set of full production periods in the solution for subplan (t_1, t_2) is a subset of the set of full production periods in the solution for subplan $(t_1, t_2 + 1)$.*

Proof. This follows from the fact that the choice periods for the smaller subplan are a subset of the set of choice periods of the larger subplan. If $d_{t_1,t_2} < kC \leq d_{t_1,t_2+1}$ for some k, then one extra production period is chosen in the larger subplan.

□

From Lemma 4, it follows that performing the initialization for all subplans with first period t_1 has a total running time that is of the same order as the running time of the initialization for subplan (t_1, T) only. The latter can easily be implemented in $O(T^2)$ time. Hence, the overall algorithm takes $O(T^3)$ in the initialization step.

4.3 Implementation of iterations

The iterations are implemented for each subplan (t_1, t_2) separately. Suppose that the optimal solutions of problems $P(t+1)', \ldots, P(t_{max})'$ and the related optimal solutions of $P(t+1), \ldots, P(t_{max})$ have been computed.

Step 1
To compute the optimal solution of $P(t)'$ from the optimal solution of $P(t+1)'$, we first move the f units from $t+1$ to t, while keeping all full production periods the same. The effect is that I_t increases by f units. Recall that the capacity in period t remains C in $P(t)'$. From Lemma 2, it follows that there is at most one period in $\{t_1, \ldots, t\}$ from which we have to move production to a period in $\{t+1, \ldots, t_2\}$.

Let the following data be available:
Period u, the earliest period in $\{t, \ldots, t_2\}$ such that $I_u < C$; note that $I_{t_2} = 0$.
For all $v \in \{t_1, \ldots, u\}$: δ_v, the earliest cheapest available period in $\{t_1, \ldots, v\}$.
Period s, the latest period in $\{t_1, \ldots, t\}$ such that $I_{s-1} < C$; by definition $I_{t_1-1} = 0$
For all $r \in \{s, \ldots, t\}$: γ_r, the most expensive production period in $\{s, \ldots, r\}$.

Note that moving production from a certain period to a later period reduces the inventory of the original production period and that of each intermediate period by C. Hence, feasibility conditions restrict us to moving production from a period in $\{s, \ldots, t\}$ to a period in $\{t+1, \ldots, u\}$. We will only perform this move if the resulting plan is really cheaper, i.e., if $cf(\gamma_t) > cf(\delta_u)$. Note that if this holds then $\delta_u > t$, otherwise this profitable move would already have been possible in problem $P(t+1)'$.

Suppose we actually move production from γ_t to δ_u. Then we update u by setting it equal to t. To see that this is correct, note that, if production is moved, then $I_t < C + f$, because otherwise the move would have been feasible (and profitable) in $P(t+1)'$. Moreover, if $cf(\gamma_t) < cf(\delta_t)$, or if $cf(\gamma_t) = cf(\delta_t)$ and $\gamma_t < \delta_t$, then we set $\delta_\tau = \gamma_t$ for $\tau \in \{\gamma_t, \ldots, t\}$. We do not need the values of s and γ_r ($r \in \{s, \ldots, t\}$) in Step 2. Therefore, these values are not updated between Steps 1 and 2 of the same period t.

Example (continued)
Consider the iteration for $t = 5$. The full production periods in the optimal solution of $P(6)'$ are 2, 3 and 5. Table 4 shows the situation just after moving the f units to period 5. We see that $u = 6$, because, starting at period 5, it is the earliest period with an inventory level below $C = 5$. Similarly, the latest period before period 5 with an inventory level less than C is period 4. Therefore, $s = 5$.

Because $cf(5) > cf(6)$, we move production from period 5 to period 6. The updated situation is shown in Table 5. We now have $u = t = 5$. Because $cf(5) > cf(4)$, the value of δ_5 does not change.

Table 4: Situation for $t = 5$

τ	1	2	3	4	5	6	7
$cf(\tau)$	19	12	5	13	17	12	10
prod.	n	y	y	n	y	n	n
I_τ	0	1	4	3	7	2	0
δ_τ	1	1	1	4	4	6	
γ_τ					5		

Table 5: Optimal solution of $P(5)'$

τ	1	2	3	4	5	6	7
$cf(\tau)$	19	12	5	13	17	12	10
prod.	n	y	y	n	n	y	n
I_τ	0	1	4	3	2	2	0
δ_τ	1	1	1	4	4		

Step 2
If t is not a full production period in the optimal solution of $P(t)'$, then the optimal solution of $P(t)$ follows immediately. Otherwise, we use Lemma 3 to compute the optimal solution of $P(t)$ from the optimal solution of $P(t)'$, i.e., we only move the production of C units from period t to another period. Due to feasibility restrictions the latter period must be chosen in $\{t_1, \ldots, u\}$. Clearly, it is optimal to take the cheapest one available, i.e., δ_u.

Example (continued)
Table 6 shows the situation at the beginning of Step 2 in the iteration for $t = 3$, i.e., the optimal solution of $P(3)'$. Note that $u = 5$. Because period 3 is a production period, we replace it by period $\delta_5 = 4$ to obtain the optimal solution of $P(3)$.

Table 6: Situation for $t = 3$

τ	1	2	3	4	5	6	7
$cf(\tau)$	19	12	5	13	17	12	10
prod.	n	y	y	n	n	y	n
I_τ	0	1	7	6	2	2	0
δ_τ	1	1	1	4	4		

Updating the data in succesive iterations
Consider the iteration for period $t-1$. Starting with the optimal solution of $P(t)'$, we first move the f units from t to $t-1$. This increases the inventory of period $t-1$ by f units. We update u correctly by setting it equal to $t-1$ if $I_{t-1} < C$.

It can easily be verified that, unless $t - 1 < s$, there is no need to update s if production has not been moved in Step 1 of the preceding iteration. Furthermore, we have the following result.

Lemma 5 *Suppose that in Step 1 of the iteration for period t, production is moved from a period in $\{s, \ldots, t\}$ to a period in $\{t+1, \ldots, u\}$. Then it is not necessary to check whether production should be moved in Step 1 of the iterations for periods $\{s, \ldots, t-1\}$.*

Proof. Suppose that a full production period is moved in Step 1 of the iteration for t from a period in $\{s, \ldots, t\}$ to a period after t. Note that this move reduces the inventory of t to a level below C. Suppose that the lemma is false and there are periods in $\{s, \ldots, t-1\}$ for which it is profitable to move a full production period in Step 1. Consider the first iteration for which this happens and let τ be the corresponding period. A profitable move with respect to τ consists of moving full production from a period in $\{s, \ldots, \tau\}$ to a period after τ, but not later than t. This move would also have been a feasible and profitable one with respect to the solution given at the start of the iteration for t. As this was the optimal solution of $P(t+1)'$, we have derived a contradiction. Hence, the lemma holds.

$\square$

This lemma justifies that, after a move has been performed in Step 1, we do not perform this step until we reach the iteration for $s-1$. Therefore, updating s is done correctly as follows. At the beginning of the iteration for $t-1$ we check whether $t-1 < s$. If this is the case, then we determine the new value of s and we compute the periods γ_r for all $r \in \{s, \ldots, t-1\}$.

Figure 1 summarizes how the data are initialized and updated.

Initialization
solve $P(t+1)'$
$u := t_{max}$; compute δ_v $(v \in \{t_1, \ldots, u\})$
determine s; compute γ_r $(r \in \{s, \ldots, t_{max}\})$
moved:='no'

Iterations
for $t := t_{max} - 1$ down to t_{min} do
    take solution of $P(t+1)'$; move f units from $t+1$ to t
    if $I_t < C$, then $u := t$
    if $t < s$ then
        determine s; compute γ_r $(r \in \{s, \ldots, t\})$
        moved:='no'
    if moved='no' then (Step 1:)
        move production if profitable $\rightarrow$ solution of $P(t)'$
        if production is moved then
            moved:='yes'
            $u := t$
            update δ_τ $(\tau \in \{\gamma_t, \ldots, t\})$
    perform Step 2

Figure 1: Overview of algorithm

Let us now turn to the complexity of the iterations. We will show that to compute and update the data during the iterations, each period is considered not more than a constant number of times. This implies the desired result that the iterations for a given subplan require in total $O(T)$ time.

Initially, for $t = t_{max}$, we have $u = t_{max}$ and the initial values of δ_v are computed for all $v \in \{t_1, \ldots, u\}$ simultaneously by considering v in increasing order. The initial value of s is determined by considering the periods in decreasing order, from t_{max} onwards, until the first period with inventory level less than C is found. The values of γ_r for $r \in \{s, \ldots, t_{max}\}$ are computed by considering the elements in $\{s, \ldots, t_{max}\}$ in increasing order.

Updating u is done by checking $I_t < C$ for each t during the algorithm. Updating δ_v is done only if a move is performed in Step 1. In that case, we update the value for the periods $\{\gamma_t, \ldots, t\}$, where $\gamma_t \geq s$. Step 1 will be performed again only for $t < s$, and thus the mentioned values will not be updated for a second time.

Each time s is determined we move in decreasing order from t to the first period for which the starting inventory is less than C. This step will not be repeated for any $t \geq s$, so the check takes place for each period at most once. Finally, we compute γ_r for $r \in \{s, \ldots, t\}$ just after s has been determined, by considering the elements in $\{s, \ldots, t\}$ in increasing order. Again, each period will only be considered once.

5 Concluding remarks

We have presented an $O(T^3)$ dynamic programming algorithm for solving the economic lot-sizing problem with constant capacities, concave production costs and linear holding costs. Our algorithm is an improvement over the algorithm of Florian and Klein [6] by a factor T. However, the latter algorithm also solves the more general problem in which the holding costs are concave. For our approach the linearity of the holding costs seems essential. It allows us to formulate an equivalent problem without holding costs, for which it is easy to calculate the change in costs when a full production period is moved.

The improvement in running time of our algorithm is based on the idea that for a given subplan many similar subproblems have to be solved. The algorithm exploits the fact that the optimal solutions to these problems are partially equal. The only possible way in which a further improvement could be achieved, seems to be the exploitation of relations concerning the positioning of optimal fractional periods in closely related subplans. Until now, we have not been able to identify such relations.

Acknowledgements
We would like to thank two anonymous referees and an Associate Editor of *Management Science* for their comments on an earlier draft of this paper.

References

[1] A. Aggarwal and J.K. Park. Improved Algorithms for Economic Lot-Size Problems. *Operations Research*, 41(3):549–571, 1993.

[2] G.B. Bitran and H.H. Yanasse. Computational Complexity of the Capacitated Lot Size Problem. *Management Science*, 28(10):1174–1185, 1982.

[3] H.-D. Chen, D.W. Hearn, and C.-Y. Lee. A New Dynamic Programming Algorithm for the Single Item Capacitated Dynamic Lot Size Model. *Journal of Global Optimization*, 4:285–300, 1994.

[4] C.-S. Chung and C.-H.M. Lin. An $O(T^2)$ Algorithm for the *NI/G/NI/ND* Capacitated Lot Size Problem. *Management Science*, 34:420–426, 1988.

[5] A. Federgruen and M. Tzur. A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n Periods in $O(n \log n)$ or $O(n)$ Time. *Management Science*, 37:909–925, 1991.

[6] M. Florian and M. Klein. Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science*, 18:12–20, 1971.

[7] M. Florian, J.K. Lenstra, and A.H.G. Rinnooy Kan. Deterministic Production Planning: Algorithms and Complexity. *Management Science*, 26:669–679, 1980.

[8] S. Van Hoesel, A. Wagelmans, and B. Moerman. Using Geometric Techniques to Improve Dynamic Programming Algorithms for the Economic Lot-Sizing Problem and Extensions. Report 9212/A, Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands, 1992. *(to appear in European Journal of Operational Research).*

[9] Y. Pochet and L.A. Wolsey. Lot-Sizing with Constant Batches: Formulation and Valid Inequalities. *Mathematics of Operations Research*, 18:767–785, 1993.

[10] A.F. Veinott, Jr. Unpublished class notes. Program in Operations Research, Stanford University, Stanford, California, 1963.

[11] A. Wagelmans, S. Van Hoesel, and A. Kolen. Economic Lot-Sizing: An $O(n \log n)$ Algorithm that Runs in Linear Time in the Wagner–Whitin Case. *Operations Research*, 40:S145–156, 1992.