ECONOMIC LOT SIZING: AN $O(n \log n)$ ALGORITHM THAT RUNS IN LINEAR TIME IN THE WAGNER-WHITIN CASE

ALBERT WAGELMANS and STAN VAN HOESEL
Erasmus University, Rotterdam, The Netherlands

ANTOON KOLEN
Limburg University, Maastricht, The Netherlands

(Received October 1989; revision received October 1990; accepted November 1990)

We consider the n-period economic lot sizing problem, where the cost coefficients are not restricted in sign. In their seminal paper, H. M. Wagner and T. M. Whitin proposed an $O(n^2)$ algorithm for the special case of this problem, where the marginal production costs are equal in all periods and the unit holding costs are nonnegative. It is well known that their approach can also be used to solve the general problem, without affecting the complexity of the algorithm. In this paper, we present an algorithm to solve the economic lot sizing problem in $O(n \log n)$ time, and we show how the Wagner-Whitin case can even be solved in linear time. Our algorithm can easily be explained by a geometrical interpretation and the time bounds are obtained without the use of any complicated data structure. Furthermore, we show how Wagner and Whitin's and our algorithm are related to algorithms that solve the dual of the simple plant location formulation of the economic lot sizing problem.

In 1958, Wagner and Whitin published their seminal paper, "Dynamic Version of the Economic Lot Size Model." Their approach to solve the economic lot sizing problem still stands as a classical application of dynamic programming and it is frequently used in practice; for instance, in MRP packages the algorithm is often used to solve subproblems that occur in complex production structures.

In this paper, we consider the economic lot sizing problem for which the marginal production costs may differ between periods and all cost coefficients are unrestricted in sign. The problem originally treated by Wagner and Whitin is a special case of this problem because they assumed identical marginal production costs and nonnegative unit holding costs. However, it is well known that the Wagner-Whitin algorithm can easily be modified to deal with the general case. For both cases the algorithm runs in $O(n^2)$ time, where n is the number of periods of the problem instance; for an efficient implementation, see Evans (1985). In their paper, Wagner and Whitin suggested a way to lower the computational burden; related results can be found in Zabel (1964), Eppen, Gould and Pashigian (1969), and Lundin and Morton (1975). However, although possibly useful in practice, these results do not affect the complexity of the algorithm.

We will present an algorithm to solve the economic lot sizing problem that runs in $O(n \log n)$ time and show how a special case—including the Wagner-Whitin case— can be solved in $O(n)$ time. The algorithm is based on a well known dynamic programming formulation which uses a backward recursion. Assuming production in a given period, the recursion formula prescribes an optimal next production period. Our algorithm identifies periods that will never be chosen as production periods in an optimal production plan. The crucial idea is that from the remaining periods it is relatively easy to select an optimal next production period.

Aggarwal and Park (1990), and Federgruen and Tzur (1991) independently obtained results similar to those presented in this paper. However, our approach is significantly different from theirs and the main advantage of our method is that it can be explained easily by giving an insightful geometric interpretation.

In Section 1, we discuss the economic lot sizing problem and make some preliminary remarks. Our algorithm is explained in Section 2 and the necessary modifications to solve the Wagner-Whitin case in linear time are discussed in Section 3. The algorithm was discovered by solving the dual of the simple plant location formulation of the economic lot sizing

Subject classifications: Analysis of algorithms, computational complexity: economic lot sizing problem. Dynamic programming, applications: economic lot sizing by dynamic programming. Inventory/production: complexity of economic lot sizing.
Area of review: OPTIMIZATION

0030-364X/92/4001-S145 $01.25

problem, and in Section 4 we show how the structure of this dual enables us to solve it by inspection. Two algorithms are given: a greedy forward algorithm that corresponds to the Wagner-Whitin approach and a nongreedy backward algorithm that is essentially the algorithm of Section 2. Concluding remarks can be found in Section 5.

1. THE ECONOMIC LOT SIZING PROBLEM

In the economic lot sizing problem (ELS) one is asked to satisfy at minimum cost the known nonnegative demands for a specific commodity in a number of consecutive periods (the "planning horizon"). It is possible to store units of the commodity to satisfy future demands, but backlogging is not allowed. For every period, the production costs consist of two components: a cost per unit produced and a fixed setup cost that is incurred whenever production occurs in the period. In addition to the production costs, there are holding costs which are linear in the inventory level at the end of the period. Both the inventory at the beginning and at the end of the planning horizon are assumed to be zero. We can always assume that the demand in the last period is positive, because otherwise we could delete this period without really changing the problem.

Note that we do not make any assumption about the sign of the cost coefficients. This is motivated by the fact that instances of ELS often occur as subproblems while solving complex production problems. When, for instance, Langrangian relaxation is used, these subproblems may have negative cost coefficients.

It is useful to consider some mathematical formulations of ELS. Let n be the length of the planning horizon and d_i, p_i, f_i, h_i denote, respectively, the demand, marginal production cost, setup cost and unit holding cost in period i, $i = 1, \ldots, n$. Given the problem description above, the most natural way to formulate ELS as a mixed-integer program is by using the following variables:

x_i: the number of units produced in period i;
s_i: the number of units in stock at the end of period i;

$$y_i = \begin{cases} 1 & \text{if a setup occurs in period } i \\ 0 & \text{otherwise.} \end{cases}$$

Define $d_{ij} = \sum_{t=i}^{j} d_t$, $1 \leq i \leq j \leq n$, then a correct formulation of ELS is as follows.

Formulation I

$$\text{Minimize} \quad \sum_{i=1}^{n} (p_i x_i + f_i y_i + h_i s_i)$$

subject to

$$x_i + s_{i-1} - s_i = d_i \qquad \text{for } i = 1, \ldots, n$$

$$d_{in} y_i - x_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$

$$s_0 = s_n = 0$$

$$x_i \geq 0, \quad s_i \geq 0, \; y_i \in \{0, 1\} \quad \text{for } i = 1, \ldots, n.$$

Because $s_i = \sum_{t=1}^{i} x_t - \sum_{t=1}^{i} d_t$, $i = 1, \ldots, n$, we can eliminate these variables from the formulation. This results in the next formulation.

Formulation II

$$\text{Minimize} \quad \sum_{i=1}^{n} (c_i x_i + f_i y_i) - \sum_{i=1}^{n} h_i d_{1i}$$

subject to

$$\sum_{t=1}^{n} x_t = d_{1n}$$

$$\sum_{t=1}^{i} x_t \geq d_{1i} \qquad \text{for } i = 1, \ldots, n-1$$

$$d_{in} y_i - x_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$

$$x_i \geq 0, \quad y_i \in \{0, 1\} \quad \text{for } i = 1, \ldots, n.$$

Here $c_i \equiv p_i + \sum_{t=i}^{n} h_t$, $i = 1, \ldots, n$.. Note that the last summation in the objective function is a constant and can therefore be omitted. This reformulation is useful because it shows us that we can restrict our analysis to instance of ELS where the holding costs are zero.

From now on, we shall work with the marginal production costs c_i, $i = 1, \ldots, n$. As mentioned before, we do not make any assumption about the sign of these costs. The fact that such an assumption is unnecessary follows from the first constraint of **II**, which implies that adding the same amount to all marginal production costs shifts the objective function of all feasible solutions by the same amount. Hence, not the values, but rather the differences between marginal production costs play a role in determining the optimal solution. The algorithm that we present in the next section assumes nonnegative setup costs. However, this does not mean that instances with negative setup costs cannot be solved. If $f_i < 0$, then it will always be profitable to set up in period i (even if there is no production in that period). By redefining the setup costs for those periods to be zero, we obtain a problem instance with nonnegative setup costs. Solving this instance and adding all negative setup costs to the obtained solution value gives the optimal value of the original instance.

A third formulation played an important role in discovering the algorithm that we are going to present. In this formulation the x_t-variables are disaggregated into variables x_{it}:

x_{it}: the number of units produced in period i to satisfy demand of period $t \geq i$.

The formulation is as follows.

Formulation III

$$\text{Minimize} \quad \sum_{i=1}^{n} (f_i y_i + c_i \sum_{t=i}^{n} x_{it})$$

subject to

$$\sum_{i=1}^{t} x_{it} = d_t \qquad \text{for } t = 1, \ldots, n$$

$$d_t y_i - x_{it} \geq 0 \qquad \text{for } i = 1, \ldots, n, \text{ for } t = i, \ldots, n$$

$$x_{it} \geq 0, \quad y_i \in \{0, 1\} \qquad \text{for } i = 1, \ldots, n, \text{ for } t = i, \ldots, n.$$

This formulation is a scaled version of the so-called simple plant location formulation of ELS, in which one uses the variables:

z_{it}: the fraction of demand of period t satisfied by production in period $i \leq t$;

(*i.e.*, $z_{it} = x_{it}/d_t$ if $d_t > 0$). It has been shown in Krarup and Bilde (1977) that the LP-relaxation of that formulation has optimal solution in which the y-variables are integer; of course, this must also hold for the LP-relaxation of **III**. Although the dual programs of both relaxations are essentially the same, it is more convenient to solve the dual of the relaxation of **III**, especially when we want to allow zero-demand in some periods. In Section 4, we shall present algorithms to solve this dual.

ELS is traditionally not solved by explicitly using any of the formulations above, but by dynamic programming. The key observation to obtain a dynamic programming formulation of the problem is that it suffices to consider only feasible solutions in which the inventory at the beginning of production periods is zero (the "zero-inventory property"); in other words, production in period i equals 0 or d_{ik} for some $k \geq i$. This property was stated first by Wagner and Whitin (1958) for their special case. Later, Wagner (1960) showed that the property even holds under the assumption of concave production costs (see also Zangwill 1968).

In the next section, we shall present our algorithm which is essentially a backward dynamic programming algorithm. For the sake of completeness, we note here that a similar algorithm based on a forward recursion can be derived. However, the latter algorithm uses a somewhat more complicated data structure than the one given below; for details we refer to Van Hoesel (1991).

2. AN $O(n \log n)$ ALGORITHM TO SOLVE ELS

In this section, we assume that all setup costs are nonnegative. We define $G(t)$ to be the cost of an optimal solution to the instance of ELS with a planning horizon consisting of periods t to n, $t = 1, \ldots, n$. Furthermore, $G(n+1)$ is defined to be zero. Because of the zero-inventory property the following recursion holds:

$$G(i) = \begin{cases} \min\limits_{i<t\leq n+1} f_t + c_t d_{i,t-1} + G(t) & \text{if } d_i > 0 \\ \min\left[G(i+1), \min\limits_{i+1<t\leq n+1} f_t + c_t d_{i,t-1} + G(t) \right] & \text{if } d_i = 0. \end{cases} \quad (1)$$

The recursion formula states that if $d_i > 0$ it is necessary to set up in period i; we produce for a number of consecutive periods and follow an optimal strategy for the remaining periods. If $d_i = 0$ it is also possible to omit the setup and just follow the optimal strategy from period $i+1$ onward.

Using (1) for calculating $G(i)$ involves the comparison of $n - i + 1$ expressions. A straightforward application of this recursion leads to an $O(n^2)$ algorithm. We shall show that given $G(t)$ for $t = i+1, \ldots, n+1$, it is possible to determine $\min_{i<t\leq n+1}\{c_i d_{i,t-1} + G(t)\}$ in $O(\log n)$ time. Because of (1), this implies that $G(i)$ can be determined in $O(\log n)$ time.

To facilitate the exposition we plot the points $(d_{tn}, G(t))$ for $t = i+1, \ldots, n+1$, as in Figure 1, where cumulative demand is put on the horizontal axis and the vertical axis corresponds to the minimal costs. Note that one of the plotted points must be the origin because $(d_{n+1,n}, G(n+1) \equiv (0, 0)$. Let LE denote the lower convex envelope of the plotted points, then we define on $[0, d_{i+1,n}]$ the function g by $g(z) = w$ if and only if $(z, w) \in$ LE. It follows that $g(d_{i+1,n}) = G(i+1)$, because if $d_{i+1} = 0$ then, according to (1), all plotted points with a coordinate $d_{i+1,n}$ on the horizontal axis cannot lie below $(d_{i+1,n}, G(i+1))$. Furthermore, it is clear that g is a piecewise linear convex function on $[0, d_{i+1,n}]$ (see Figure 1).

A value $z \in (0, d_{i+1,n})$ in which g changes slope is called a *breakpoint* of g. We also define $z = 0$ and

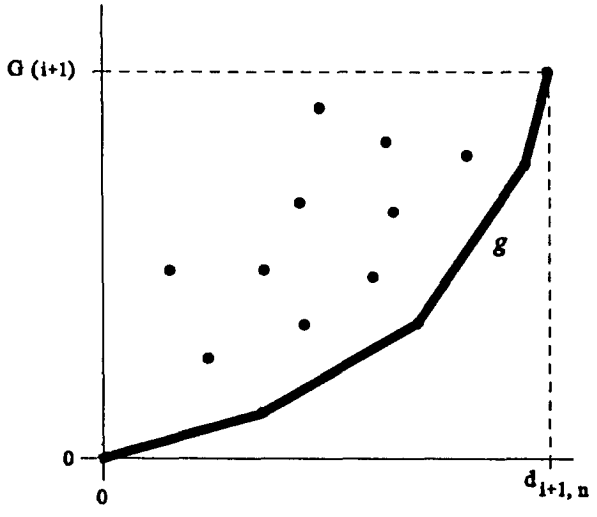

Figure 1. Diagram of minimal cost versus cumulative demand.

$z = d_{i+1,n}$ to be breakpoints. It is obvious that if z is a breakpoint, then $(z, g(z)) = (d_{tn}, G(t)$ for some period $t \in \{i + 1, \ldots, n + 1\}$. Suppose that there are r breakpoints of g and let $i + 1 = t(1) < \ldots < t(r) = n + 1$ denote the corresponding periods. These periods will be called *efficient*, because we shall show that only these periods need to be considered for the determination of $\min_{i<t\leq n+1}\{c_i d_{i,t-1} + G(t)\}$.

Proposition 1. *Let* $t(p)$, $p = 1, \ldots, r$, *denote the efficient periods, then*

$$\min_{i<t\leq n+1} c_i d_{i,t-1} + G(t) = \min_{1\leq p\leq r} c_i d_{i,t(p)-1} + G(t(p)).$$

Proof. Suppose that j, $i + 1 < j < n + 1$ is not an efficient period and k and l are the two consecutive efficient periods, such that $k < j < l$. The slope of g on $[d_{ln}, d_{kn}]$ is equal to $[G(k) - G(l)]/d_{k,l-1}$, hence

$$g(d_{jn}) = G(l) + \frac{G(k) - G(l)}{d_{k,l-1}} d_{j,l-1}.$$

Furthermore, we know that $G(j) \geq g(d_{jn})$.

First assume that $c_i \geq [G(k) - G(l)]/d_{k,l-1}$, then

$$\begin{aligned} & c_i d_{i,j-1} + G(j) \\ & \geq c_i d_{i,k-1} + c_i d_{k,j-1} + g(d_{jn}) \\ & \geq c_i d_{i,k-1} + \frac{G(k) - G(l)}{d_{k,l-1}} d_{k,j-1} + G(l) \\ & \quad + \frac{G(k) - G(l)}{d_{k,l-1}} d_{j,l-1} \\ & = c_i d_{i,k-1} + G(k). \end{aligned}$$

This means that $c_i d_{i,j-1} + G(j)$ is dominated by $c_i d_{i,k-1} + G(k)$. For the case, $c_i < [G(k) - G(l)]/d_{k,l-1}$ we can prove that $c_i d_{i,j-1} + G(j) \geq c_i d_{i,l-1} + G(l)$:

$$\begin{aligned} & c_i d_{i,j-1} + G(j) \\ & \geq c_i d_{i,l-1} - c_i d_{j,l-1} + g(d_{jn}) \\ & \geq c_i d_{i,l-1} - \frac{G(k) - G(l)}{d_{k,l-1}} d_{j,l-1} + G(l) \\ & \quad + \frac{G(k) - G(l)}{d_{k,l-1}} d_{j,l-1} \\ & = c_i d_{i,l-1} + G(l). \end{aligned}$$

Hence, we have shown that for the determination of $\min_{i<t\leq n+1}\{c_i d_{i,t-1} + G(t)\}$ it suffices to consider only the efficient periods.

For a geometric interpretation of the proof of Proposition 1 consider Figure 2, where c_i is assumed greater than the slope of g on $[d_{ln}, d_{kn}]$. Note that $c_i d_{i,j-1} + G(j)$ is exactly the coordinate on the vertical axis of the intersection point of the vertical line through $(d_{in}, 0)$ and the line with slope c_i that passes through $(d_{jn}, G(j))$. Of course, the corresponding values for the other periods can be determined analogously. We have proven that $(d_{in}, c_i d_{i,j-1} + G(j))$ never lies below $(d_{in}, c_i d_{i,k-1} + G(k))$.

From the discussion above it is also clear how we can determine $\min_{1\leq p\leq r}\{c_i d_{i,t(p)-1} + G(t(p))\}$ geometrically (see Figure 3): For every efficient period t we determine the intersection point of the vertical line through $(d_{in}, 0)$ and the line with slope c_i that passes through $(d_{in}, G(t))$. The coordinate on the vertical

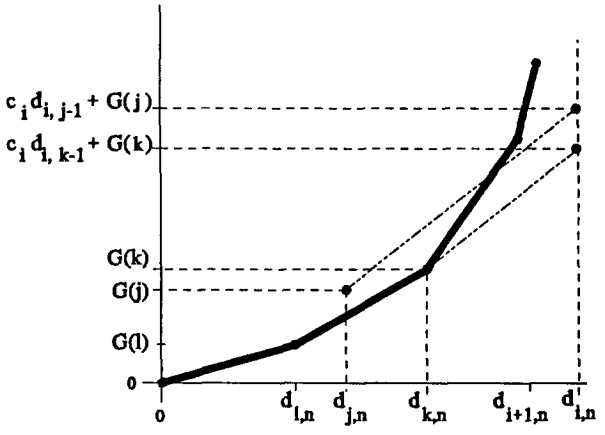

Figure 2. Geometric interpretation of proof of Proposition 1.

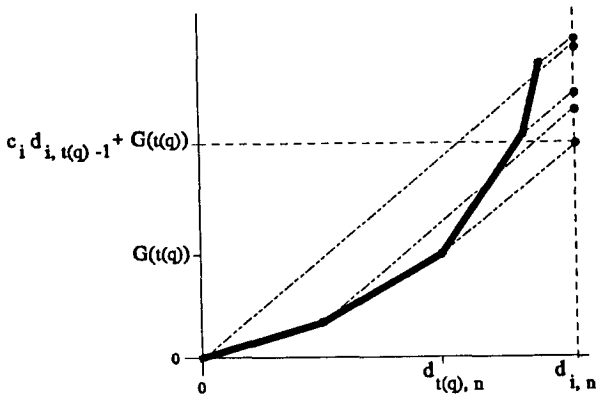

Figure 3. Geometric determination of minimum value.

axis of the lowest of these intersection points is equal to $\min_{1 \leq p \leq r}\{c_i d_{i,t(p)-1} + G(t(p))\}$. This means that we are in fact looking for the line with slope c_i that supports the epigraph of g. In Figure 3 it is clear that the slope of g is smaller than c_i on $[0, d_{t(q),n}]$ and greater than c_i on $[d_{t(q),n}, d_{i+1,n}]$. For a formal proof of this property it is sufficient to prove the following fact.

Proposition 2. *Let k and l, $k < l$, be two consecutive efficient periods. If*

$$\frac{G(k) - G(l)}{d_{k,l-1}} < c_i$$

then

$$c_i d_{i,k-1} + G(k) < c_i d_{i,l-1} + G(l);$$

otherwise, $c_i d_{i,k-1} + G(k) \geq c_i d_{i,l-1} + G(l)$.

Proof. Suppose that $[G(k) - G(l)]/d_{k,l-1} < c_i$, then $G(k) < c_i d_{k,l-1} + G(l)$. Adding $c_i d_{i,k-1}$ on both sides of the inequality sign results in $c_i d_{i,k-1} + G(k) < c_i d_{i,l-1} + G(l)$. In the other case, the proof is completely analogous.

From Proposition 2 and the convexity of g on $[0, d_{i+1,n}]$, it follows that $c_i d_{i,t(q)-1} + G(t(q)) = \min_{1 \leq p \leq r}\{c_i d_{i,t(p)-1} + G(t(p))\}$ if we take

$$q := \min\left[r, \min\left\{p \mid 1 \leq p < r \text{ and } \frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}} < c_i\right\}\right],$$

because then

$$c_i d_{i,t(p)-1} + G(t(p)) \geq c_i d_{i,t(p+1)-1} + G(t(p+1)) \quad \text{for } p = 1, \ldots, q-1$$

and

$$c_i d_{i,t(p)-1} + G(t(p)) < c_i d_{i,t(p+1)-1} + G(t(p+1)) \quad \text{for } p = q, \ldots, r-1.$$

A crucial observation at this point is that once the efficient periods are known, that is, given the *ordered* ratios

$$\frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}}, \quad p = 1, \ldots, r-1,$$

we can determine $t(q)$ in $O(\log n)$ time by binary search.

Note that the choice of $t(q)$ as the next production period after i clearly has a sensible economic interpretation: the ratios

$$\frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}}, \quad p = 1, \ldots, r-1,$$

are marginal costs per unit of demand, and we produce up to the period where this quantity drops below the marginal production costs of period i.

After $G(i)$ has been determined for a certain $i > 1$, we want to proceed with the analogous calculation of $G(i-1)$. However, first we must update the set of efficient periods. Geometrically we can apply the following procedure: add the point $(d_{in}, G(i))$ and find the smallest efficient period $t(s)$, such that the slope of the line segment connecting $(d_{in}, G(i))$ to $(d_{t(s),n}, G(t(s)))$ is greater than the slope of the line segment connecting $(d_{t(s+1),n}, G(t(s+1)))$ to $(d_{t(s),n}, G(t(s)))$. The new set of efficient periods consists of period i and the periods $t(s)$ to $t(r)$ (see Figure 4).

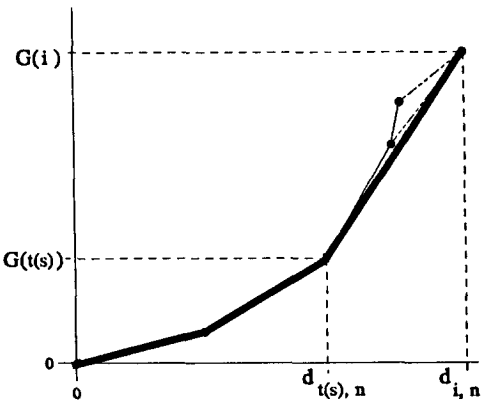

Figure 4. Updating the lower convex envelope.

To define $t(s)$ formally a few remarks are needed. We adopt the convention that if $d_i = 0$, then the efficient period $i+1$ is replaced by the efficient period i. It is trivial that in the case where $d_i = 0$ and $G(i) = G(i-1)$, all other efficient periods remain efficient, i.e., we let $s := 2$. Otherwise, it always holds that $s \leq q$. To see this, suppose that $q < r$. By the definition of $t(q)$ and because $G(i) = f_i + c_i d_{i,t(q)-1} + G(t(q))$ the following holds

$$\frac{G(i) - G(t(q))}{d_{i,t(q)-1}} \geq c_i > \frac{G(t(q)) - G(t(q+1))}{d_{t(q),t(q+1)-1}}$$

and this implies that $s \leq q$.

We then define s as:

$$s := \min\left[q, \min\left\{p \mid 1 \leq p < q,\ d_{i,t(p)-1} > 0\right.\right.$$

$$\left.\left. \text{and } \frac{G(i) - G(t(p))}{d_{i,t(p)-1}} > \frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}}\right\}\right].$$

Note that $d_{i,t(p)-1} = 0$ can only occur if $t(p) = i+1$ and $d_i = 0$. Hence, the condition $d_{i,t(p)-1} > 0$ guarantees that if $d_i = 0$ the period $i+1$ is no longer efficient.

To find s we simply compare

$$\frac{G(i) - G(t(p))}{d_{i,t(p)-1}}$$

to

$$\frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}}$$

for increasing p, $1 \leq p < q$, and stop as soon as the first expression is greater than the second. Note that if a period is not efficient during the calculation of $G(i)$, it can never be efficient during the calculation of $G(j)$ for all $j < i$; i.e., a period becomes inefficient at most once.

Before giving a complete description of the algorithm a few remarks may clarify that it can indeed be implemented to run in $O(n \log n)$ time. First note that the marginal production costs c_i, $i = 1, \ldots, n$, can be calculated from p_i and h_i, $i = 1, \ldots, n$, in $O(n)$ time. Redefining the setup costs is of the same complexity. In the implementation below we assume nonnegative setup costs; the modifications needed otherwise are straightforward. Furthermore, it is not necessary to calculate d_{ij} for all pairs i, j with $1 \leq i \leq j \leq n$. At the start of the algorithm we only calculate the coefficients d_{in}, $i = 1, \ldots, n$ (again in linear time). Because $d_{ij} = d_{in} - d_{j+1,n}$, $1 \leq i \leq j < n$, these coefficients can then be obtained in constant time when needed.

The main part of the algorithm consists of n iterations: $G(i)$ is calculated in iteration $n - i + 1$. The implementation below uses a list ("stack") L that contains the efficient periods at the beginning of iteration $n - i + 1$ in increasing order. As noted before, we can find the period $t(q)$ by binary search in L. The total time spent on searching is therefore $O(n \log n)$. In every iteration we have to make a few comparisons to determine the period $t(s)$. After every comparison, we either conclude that we have found $t(s)$ or we have to continue by considering the next period in L. The first case occurs exactly once in every iteration, i.e., in total n times. In the second case, we delete a period from L. As every period is deleted from L at most once, this case can occur no more than n times. Thus, the overall complexity of calculating $G(1)$ is $O(n \log n)$. At the end of the algorithm an optimal solution is constructed in linear time using information obtained during the iterations. The output is given in terms of the x- and y-variables used in models **I** and **II** of Section 1. Initially L is empty and all variables are equal to zero.

The next description should be self-explanatory, except for $l(p)$ which is used to indicate the successor of p in L.

Algorithm

(Input: $p, h \in \mathbb{R}^n, f, d \in \mathbb{R}^n_{\geq 0}$
Output: $x \in \mathbb{R}^n$, $y \in B^n$).

Initialization

calculate c_i and d_{in}, $i = 1, \ldots, n$
add $n+1$ to L.

Iterations

For $i := n$ down to 1 **do**
begin

$$\text{search for } q(i) := \min\left[n+1, \min\left\{p \in L \mid p < n+1 \text{ and } \frac{G(p) - G(l(p))}{d_{pn} - d_{l(p),n}} < c_i\right\}\right],$$

$G(i) := f_i + c_i[d_{1n} - d_{q(i),n}] + G(q(i))$,
if ($d_i = 0$ and $G(i+1) < G(i)$) **then**,
begin
$G(i) := G(i+1)$
$s := l(i+1)$
end

else
begin
if $d_i > 0$, **then** $s := i + 1$ **else** $s := l(i + 1)$
while

$$\left(\frac{G(i) - G(s)}{d_{in} - d_{sn}} \leqslant \frac{G(s) - G(l(s))}{d_{sn} - d_{l(s),n}} \text{ and } s < q(i)\right)$$

do $s := l(s)$
end
delete all p with $i + 1 \leqslant p < s$ from L
add i to L
end.

Construction of an optimal solution

$i := 1$
while $i \leqslant n$ **do**
if $(d_i = 0$ and $G(i) = G(i + 1))$, **then** $i := i + 1$ else
begin
$x_i := d_{in} - d_{q(i),n}$
$y_i := 1$
$i := q(i)$
end.

3. THE WAGNER-WHITIN CASE

Wagner and Whitin originally proposed their algorithm for the special case in which $p_i = p$ and $h_i \geqslant 0$, $i = 1, \ldots, n$. For the marginal production costs $c_i \equiv p_i + \sum_{t=i}^{n} h_t$, it follows that $c_i \geqslant c_{i+1}$, $i = 1, \ldots, n - 1$. The discussion of the complexity of our algorithm in the previous section makes clear that apart from the binary searches, the number of elementary operations is $O(n)$. We shall now show that when the marginal production costs are nonincreasing over time we can replace the binary searches by a much simpler search strategy that requires in total $O(n)$ comparisons. Hence, the Wagner-Whitin model is an interesting special case of a class of lot sizing problems that can be solved in linear time.

In our exposition, we shall use the same notation as in the descripton of the algorithm at the end of the previous section. Consider iteration $n - i + 1$, $1 \leqslant i < n$, in which we determine

$$q(i) := \min\left[n + 1, \min\left\{p \in L \mid p < n + 1 \text{ and } \frac{G(p) - G(l(p))}{d_{pn} - d_{l(p),n}} < c_i\right\}\right],$$

where L is the current list of efficient periods.

First suppose that $q(i + 1) \notin L$, then it must have occurred in iteration $n - i$ that $s > q(i + 1)$. But this is only possible if $d_{i+1} = 0$, $G(i + 1) = G(i + 2)$ and $q(i + 1) = i + 2$. In that case, L has been updated by simply replacing period $i + 2$ by period $i + 1$. Hence, $l(i + 1)$ is the same period as the successor of $i + 2$ in iteration $n - i$. It follows that

$$\frac{G(i + 1) - G(l(i + 1))}{d_{i+1,l(i+1)-1}} = \frac{G(i + 2) - G(l(i + 1))}{d_{i+2,l(i+1)-1}} < c_{i+1} \leqslant c_i,$$

where the strict inequality follows from the definition of $q(i + 1)$. We conclude that $q(i) = i + 1$.

Now suppose that $q(i + 1) \in L$. Note that as long as $q(i + 1)$ is efficient it has the same successor $l(i + 1)$ in L. Using again the definition of $q(i + 1)$ we obtain

$$\frac{G(q(i + 1)) - G(l(q(i + 1)))}{d_{q(i+1),l(q(i+1))-1}} < c_{i+1} \leqslant c_i.$$

Hence, it follows that in this case $q(i) \leqslant q(i + 1)$.

We can adapt the implementation of the algorithm as follows: Let $m(p)$ denote the predecessor of p in L. Initialize s and $q(n + 1)$ to $n + 1$ and replace the "search for $q(i)$" by the following statements:

if $s > q(i + 1)$, **then** $q(i) := i + 1$ **else**
begin
$q(i) := m(q(i + 1))$
while $\dfrac{G(q(i)) - G(l(q(i)))}{d_{q(i),n} - d_{l(q(i)),n}} < c_i$
do $q(i) := m(q(i))$
$q(i) := l(q(i))$
end.

After each comparison in the while-loop we either conclude that $q(i)$ has been found or that we must make at least one more comparison. The first case occurs exactly once per iteration. The second case implies that we found a period p such that $q(j) < p$ for all $j \leqslant i$. For every period p this conclusion is drawn at most once. The total complexity of the algorithm is therefore $O(n)$.

Wagner and Whitin's "Planning Horizon Theorem" states that if we consider the forward recursion, then the last production period in the optimal solution for periods 1 to $i + 1$ is not smaller than the last production period in the optimal solution for periods 1 to i. The fact that $q(i) \leqslant q(i + 1)$, $1 \leqslant i \leqslant n$, means that a similar result holds with respect to recursion (1). As far as we know, this result has never before been stated explicitly in the literature.

Table I
Input Data ($\sum_{i=1}^{n} h_i d_{1i} = 3860$)

i	d_{in}	f_i	c_i
1	630	85	12
2	561	102	11
3	532	102	10
4	496	101	9
5	435	98	8
6	374	114	7
7	348	105	6
8	314	86	5
9	247	119	4
10	202	110	3
11	135	98	2
12	56	114	1

To end this section, we give the results of our algorithm for the example from Wagner and Whitin (1958). The (transformed) input data are displayed in Table I.

The output is given in Table II and should be interpreted as follows: Iterations correspond to rows. The first column of every row contains the period i for which we are calculating $G(i)$. In the second column we show the efficient periods at the beginning of the corresponding iteration. For the efficient periods between brackets we have already concluded that $q(j)$ is less than these periods for $j \leq i$. The values of $q(i)$ and $G(i)$ are given in the next two columns. The last column contains the ratio

$$\frac{G(i) - G(l(i))}{d_{i,n} - d_{l(i),n}}$$

that is calculated at the end of the iteration. To find $q(i)$ we compare c_i with the ratios of the efficient periods that are not between brackets, starting with the largest one. We stop as soon we find a ratio that is greater than c_i.

The optimal policy is to produce in periods 1, 3, 5, 8, 10 and 11. Because of the transformation of the cost coefficients we should subtract 3,860 from the value 4,724 to obtain the optimal value 864.

4. SOLVING THE DUAL OF THE SIMPLE PLANT LOCATION FORMULATION

In this section, we show how our algorithm relates to one that solves the dual of the LP-relaxation of the simple plant location formulation of ELS. In fact the algorithm presented in the previous section was developed after an $O(n \log n)$ algorithm to solve this dual had been found. As mentioned in Section 1, we know that the LP-relaxation of the simple plant location formulation has an integer optimal solution. This fact was first proven in Krarup and Bilde who suggested an $O(n^2)$ algorithm to solve this formulation. For convenience, we shall consider a slightly different formulation, viz. the LP-relaxation of model **III** of Section 1. For the dual the only consequence is that the variables are scaled differently. Apart from the fact that this facilitates the exposition, this dual is easier to treat when zero-demand is allowed.

The dual of the LP-relaxation of **III** is the following program.

Program D

$$\text{Maximize} \quad \sum_{t=1}^{n} d_t v_t - \sum_{t=1}^{n} \lambda_t$$

subject to

$$\sum_{t=i}^{n} d_t w_{it} - \lambda_i \leq f_i \qquad \text{for } i = 1, \ldots, n$$

$$v_t - w_{it} \leq c_t \qquad \text{for } i = 1, \ldots, n, \text{ for } t = i, \ldots, n$$

$$w_{it}, \lambda_i \geq 0 \qquad \text{for } i = 1, \ldots, n, \text{ for } t = i, \ldots, n$$

$$v_t \quad \text{free} \qquad \text{for } t = 1, \ldots, n.$$

In an optimal solution we can always take $w_{it} := \max\{0, v_t - c_t\}$. If $f_i \geq 0$, then the restriction $y_i \leq 1$ in the LP-relaxation of **III** is superfluous and therefore the corresponding dual variable λ_i can be taken equal to zero. If $f_i < 0$ we can take $\lambda_i = -f_i$ and solve the remaining problem. This corresponds to the way negative setup costs have been treated before. We therefore show how to solve the following program, where $f_i \geq 0$, $i = 1, \ldots, n$.

Table II
Results of Algorithm

i	Efficient Periods	$q(i)$	$G(i)$	Ratio
12	13	13	170	3.04
11	13, 12	13	368	2.73
10	13, 11	11	679	4.64
9	(13) 11, 10	11	935	5.69
8	(13) 11, 10, 9	10	1325	5.82
7	(13, 11) 10, 9, 8	8	1634	9.09
6	(13, 11, 10, 9) 8, 7	8	1859	8.90
5	(13, 11, 10, 9) 8, 6	8	2391	8.81
4	(13, 11, 10, 9) 8, 5	5	3041	10.66
3	(13, 11, 10, 9, 8) 5, 4	5	3463	11.72
2	(13, 11, 10, 9, 8) 5, 4, 3	4	3858	13.62
1	(13, 11, 10, 9, 8, 5) 4, 3, 2	3	4724	12.87

Program D′

$$\text{Maximize} \quad \sum_{t=1}^{n} d_t v_t$$

subject to

$$\sum_{t=i}^{n} d_t \max\{0, v_t - c_i\} \leq f_i \quad \text{for } i = 1, \ldots, n.$$

This program is highly structured and can be solved by inspection. One way of doing that is by considering the variables in the order of increasing index: If $d_j = 0$ we can give v_j, an arbitrary value, otherwise we let

$$v_j := \min_{i \leq j} \left\{ c_i + f_i - \frac{\sum_{t=i}^{j-1} d_t \max\{0, v_t - c_i\}}{d_j} \right\},$$

i.e., given v_t, $t < j$, v_j is assigned a value as large as possible. The solution constructed in this greedy forward way is clearly feasible and to prove optimality we first show that if $d_j \neq 0$ and $d_{j+1} \neq 0$, then $v_j \geq v_{j+1}$:

Let $k \leq j$ be such that

$$v_j = c_k + \frac{f_k - \sum_{t=k}^{j-1} d_t \max\{0, v_t - c_k\}}{d_j}.$$

Thus $v_j \geq c_k$ and $f_k - \sum_{t=k}^{j} d_t \max\{0, v_t - c_k\} = 0$. Because

$$v_{j+1} \leq c_k + \frac{f_k - \sum_{t=k}^{j} d_t \max\{0, v_t - c_k\}}{d_{j+1}}$$

it follows that $v_{j+1} \leq c_k \leq v_j$. Since the variables corresponding to periods with zero-demand can be given an arbitrary value, we may assume that the constructed solution satisfies $v_j \geq v_{j+1}$, $j = 1, \ldots, n-1$.

Now define $F(j)$ to be the cost of an optimal solution to the ELS with the planning horizon consisting of periods 1 to j, $j = 1, \ldots, n$. To prove optimality of the solution it is sufficient to show that $\sum_{t=1}^{j} d_t v_t = F(j)$, because by duality and the structure of (D'), $\sum_{t=1}^{j} d_t v_t \leq F(j)$ must hold. The proof is by induction on j, starting with $j = 1$ for which the statement is trivial.

For $j > 1$ we only have to consider the case of $d_j \neq 0$, because, otherwise, $F(j) = F(j-1)$. Let $k \leq j$ be such that

$$v_j = c_k + \frac{f_k - \sum_{t=k}^{j-1} d_t \max\{0, v_t - c_k\}}{d_j},$$

then

$$\sum_{t=1}^{j} d_t v_t = c_k d_j + f_k + \sum_{t=1}^{k-1} d_t v_t + \sum_{t=k}^{j-1} d_t (v_t - \max\{0, v_t - c_k\}).$$

Using the induction hypothesis $\sum_{t=1}^{k-1} d_t v_t = F(k-1)$, and the fact that $v_t \geq v_j \geq c_k$, $t < j$, we obtain

$$\sum_{t=1}^{j} d_t v_t = f_k + c_k d_{kj} + F(k-1). \tag{2}$$

Because

$$F(j) = \min_{i \leq j} f_i + c_i d_{ij} + F(i-1),$$

(2) implies that $\sum_{t=1}^{j} d_t v_t \geq F(j)$, which together with $\sum_{t=1}^{j} d_t v_t \leq F(j)$ yields the desired result.

From our discussion above (in particular, equation (2)) it is clear that solving **D′** in the greedy forward way directly provides an optimal solution of ELS and is closely related to the forward dynamic programming approach of Wagner and Whitin. In fact this dual algorithm is the algorithm given in Krarup and Bilde, where it was presented in a more general context.

An alternative approach to solve **D′** is closely related to the algorithm presented in Section 2. It will not come as a surprise that this algorithm works backward, i.e., variables and constraints are considered in order of decreasing index. Contrary to the forward algorithm, we cannot follow a greedy strategy: While determining a proper value for v_j we may have to revise, at the same time, values of some variables v_t, $t > j$. It is somewhat surprising that in spite of this, the algorithm can be implemented to run in $O(n \log n)$ time, while the greedy forward algorithm has a complexity of $O(n^2)$. The gain in complexity comes from the fact that we can again determine certain important indices by binary search and revising values of variables can subsequently be done in constant time. In our exposition we shall try to clarify the correspondence to the algorithm of Section 2.

Throughout the algorithm the idea is to keep the variables such that $v_j \geq v_{j+1}$, $j = i, \ldots, n-1$. Here i is the index of the variable and the constraint that we are considering in the current iteration. At the beginning of the iteration the variables v_{i+1} to v_n satisfy the constraints indexed $i+1$ to n and will always have the following property. There are indices $i + 1 = t(1) < \ldots < t(r) = n + 1$, such that $v_{t(p)} > v_{t(p+1)}$ and $v_t = v_{t(p)}$ for all t with $t(p) < t < t(p+1)$ and $p = 1, \ldots, r-1$. (Here v_{n+1} is defined sufficiently small.)

Furthermore, it holds that $d_{t(p),t(p+1)-1} \neq 0$ and $\sum_{t=t(p)}^{n} d_t v_t = G(t(p))$ for all $p < r$.

We shall call this the *staircase property*. Of course, the indices $t(1)$ to $t(r)$ correspond to the efficient periods of Section 2. Note that

$$v(t(p)) = \frac{G(t(p)) - G(t(p+1))}{d_{t(p),t(p+1)-1}}.$$

We are now going to show how to obtain a solution v'_t, $t \geq i$, such that this solution satisfies the constraints indexed i to n and the staircase property holds, i.e., in particular $\sum_{t=i}^{n} d_t v_t = G(i)$. The current slack in constraint i is

$$\delta_1 := f_i - \sum_{t=i+1}^{n} d_t \max(0, v_t - c_t)$$

$$= f_i - \sum_{t=i+1}^{t(q)-1} d_t (v_t - c_t),$$

where $q := \min[r, \min\{p \mid 1 \leq p < r \text{ and } v_t < c_t\}]$.

We distinguish between three cases.

Case 1. $d_i = 0$. By taking $v'_t := v_t$, $t > i$, and $v'_i := v'_{i+1}$, we obtain a solution with all the desired properties, because

$$\sum_{t=i}^{n} d_t v'_t = \sum_{t=i+1}^{n} d_t v'_t = \sum_{t=i+1}^{n} d_t v_t$$

$$= G(i+1) \geq G(i),$$

and by duality, $\sum_{t=i}^{n} d_t v'_t \leq G(i)$.

Case 2. $d_i \neq 0$ and $d_i v_{i+1} \leq c_i d_i + \delta_1$. Let $v'_i := c_i + \delta_1/d_i$ and $v'_t = v_t$, $t > i$, then $v'_i / \geq v'_{i+1}$, constraints i to n are satisfied and

$$\sum_{t=i}^{n} d_t v'_t$$

$$= c_i d_i + \delta_1 + \sum_{t=i+1}^{n} d_t v'_t$$

$$= c_i d_i + f_i - \sum_{t=i+1}^{t(q)-1} d_t (v_t - c_t) + \sum_{t=i+1}^{n} d_t v_t$$

$$= f_i + c_i d_{i,t(q)-1} + \sum_{t=t(q)}^{n} d_t v_t$$

$$= f_i + c_i d_{i,t(q)-1} + G(t(q)) \geq G(i).$$

Again, we conclude that $\sum_{t=i}^{n} d_t v'_t = G(i)$.

Case 3. $d_i v_{i+1} > c_i d_i + \delta_1$. Satisfying constraint i with a solution v'_t, $t = i, \ldots, n$, such that $v'_t \geq v'_{t+1}$, $t = i, \ldots, n-1$, is now only possible if $v'_t < v_t$ for some t, $i+1 \leq t \leq t(q)-1$. Consider an arbitrary u, $2 \leq u \leq q$ and suppose that we would choose the solution $\hat{v}_t := v_{t(u)}$, $i \leq t < t(u)$, $\hat{v}_t = v_t$, $t(u) \leq t \leq n$, then the slack in constraint i would be

$$\delta_1 - d_i(\hat{v}_i - c_i) + \sum_{t=i+1}^{t(u)-1} d_t (v_t - \hat{v}_t)$$

$$= \delta_1 - d_i(v_{t(u)} - c_i) + \sum_{t=i+1}^{t(u)-1} d_t (v_t - v_{t(u)})$$

$$= f_i - \sum_{t=i+1}^{t(q)-1} d_t (v_t - c_t)$$

$$- d_i(v_{t(u)} + c_i) + \sum_{t=i+1}^{t(u)-1} d_t (v_t - v_{t(u)})$$

$$= f_i + c_i d_{i,t(q)-1} - v_{t(u)} d_{i,t(u)-1} - \sum_{t=t(u)}^{t(q)-1} d_t v_t$$

$$= f_i + c_i d_{i,t(q)-1} - v_{t(u)} d_{i,t(u)-1} - G(t(u)) + G(t(q)). \quad (3)$$

The value of expression (3) must increase for increasing u and it is certainly positive for $u = q$. Let s be the smallest index, such that (3) is positive if $u = s$. In other words, s is the smallest index among 1 to q, such that by letting all variables with a smaller index be equal to $v_{t(s)}$, while not changing the values of the other variables, constraint i is satisfied. Let δ_2 be the corresponding slack, i.e.,

$$\delta_2 := f_i + c_i d_{i,t(q)-1} - v_{t(s)} d_{i,t(s)-1}$$

$$- G(t(s)) + G(t(q)) > 0.$$

The solution v'_t, $t \geq i$, is now defined as:

$$v'_t := v_{t(s)} + \delta_2 / d_{i,t(s)-1}, \quad t = 1, \ldots, t(s) - 1$$

$$v'_t := v_t, \quad t \geq t(s).$$

The new solution satisfies constraint i with equality and the constraints $i+1$ to n are also satisfied because $v'_t \leq v_t$, $t \geq i+1$. Moreover, the staircase property holds because

$$\sum_{t=i}^{n} d_t v'_t = \sum_{t=i}^{t(s)-1} d_t v'_t + \sum_{t=t(s)}^{n} d_t v_t$$

$$= v_{t(s)} d_{i,t(s)-1} + \delta_2 + G(t(s))$$

$$= f_i + c_i d_{i,t(q)-1} + G(t(q)).$$

It follows that $\sum_{t=i}^{n} d_t v'_t = G(i) = f_i + c_i d_{i,t(q)-1} + G(t(q))$. Now the equivalence of the choice of s here and in Section 2 becomes clear: (3) is positive

if and only if

$$G(i) - v_{t(u)}\, d_{i,t(u)-1} - G(t(u)) > 0$$

$$\Leftrightarrow \frac{G(i) - G(t(u))}{d_{i,t(u)-1}} > v_{t(u)}$$

$$= \frac{G(t(u)) - G(t(u+1))}{d_{t(u),t(u+1)-1}}.$$

This completes the description of an iteration of the backward algorithm to solve **D′** (and at the same time ELS).

The reader should have no difficulties in finding an implementation of the algorithm that runs in $O(n \log n)$ time. The crucial observation to be made is that the solution is always completely determined by the values $v_{t(p)}$, $p = 1, \ldots, r-1$.

5. CONCLUDING REMARKS

We have shown that ELS can be solved in $O(n \log n)$ time and how this fact is related to the special structure of the dual of the simple plant location formulation. Furthermore, we showed that the Wagner-Whitin case can be solved in linear time.

The algorithm presented here has some similarities with the algorithm to solve ELS that can be found in Van Hoesel, Kolen and Wagelmans (1991). This last algorithm is based on solving the dual of a complete linear description of the convex hull of feasible solutions of formulation **II**, which was found by Barany, Van Roy and Wolsey (1984). It also works backward, but instead of considering the periods one by one it can be regarded as only calculating $G(t)$ for the periods t that are efficient after the last iteration of the algorithm in Section 2. The algorithm is of the greedy type and runs in $O(n^2)$ time. A nongreedy $O(n \log n)$ algorithm to solve the same dual may well exist.

In Van Hoesel (1991), the approach described in Section 2 is generalized and applied to other lot sizing problems. For instance, it turns out that the model in which backlogging is allowed can be solved in $O(n \log n)$ time. (Using the different approaches this result has also been obtained by Aggarwal and Park 1990, and Federgruen and Tzur 1991.)

In this paper, we have focused on the theoretical properties of our algorithm. However, it seems worthwhile to study the performance of the algorithm on practical problems. For instance, a proper implementation of the algorithm can be expected to be computationally more efficient than the efficient implementation of the Wagner-Whitin algorithm described in Evans (1985), while requiring less storage. The algorithm could also be competitive with various widely used heuristics (see, e.g., Baker 1989), although these may produce solutions that are more attractive from other points of view. For limited computational results we refer to Van Hoesel.

ACKNOWLEDGMENT

We thank Karen Aardal, Alexander Rinnooy Kan, Luk Van Wassenhove and Laurence Wolsey for their comments on an earlier draft of this paper, and Hein Heuvelmans for his assistance in making the figures. Part of this work was done while the first author was visiting the Center for Operations Research and Econometrics (CORE), Louvain-la-Neuve, Belgium. The research of the second author was supported by the Netherlands Organization for Scientific Research (NWO) under grant 611-304-017.

REFERENCES

AGGARWAL, A., AND J. K. PARK. 1990. Improved Algorithms for Economic Lot-Size Problems. Working Paper, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge Mass.

BAKER, K. R. 1989. Lot-Sizing Procedures and a Standard Data Set: A Reconciliation of the Literature. Report, The Amos Tuck School of Business Administration, Dartmouth College, Hanover, N.H.

BARANY, I., T. J. VAN ROY AND L. A. WOLSEY. 1984. Uncapacitated Lot-Sizing: The Convex Hull of Solutions. *Math. Prog. Study* **22,** 32–43.

EPPEN, G. D., F. J. GOULD AND B. P. PASHIGIAN. 1969. Extensions of the Planning Horizon Theorem in the Dynamic Lot Size Problem. *Mgmt. Sci.* **15,** 268–277.

EVANS, J. R. 1985. An Efficient Implementation of the Wagner-Whitin Algorithm for Dynamic Lot-Sizing. *J. Opns. Mgmt.* **5,** 229–235.

FEDERGRUEN, A., AND M. TZUR. 1991. A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models With n Periods in $O(n \log n)$ or $O(n)$ Time. *Mgmt. Sci.* **37,** 909–925.

FEDERGRUEN, A., AND M. TZUR. 1990. The Dynamic Lot Sizing Model With Backlogging: A Simple $O(n \log n)$ Algorithm. Working Paper, Graduate School of Business, Columbia University, New York.

KRARUP, J., AND O. BILDE. 1977. Plant Location, Set Covering and Economic Lot Size: An $O(mn)$-Algorithm for Structured Problems. *Numerische Methoden bei Optimierungsaufgaben, Band 3:*

Optimierung bei Graph-Theoretischen und Ganzzahligen Problemen. Birkhauser Verlag, Basel, Switzerland, 155–180.

LUNDIN, R. A., AND T. E. MORTON. 1975. Planning Horizons for the Dynamic Lot Size Model: Zabel vs. Protective Procedures and Computational Results. *Opns. Res.* **23,** 711–734.

VAN HOESEL, C. P. M. 1991. Algorithms for Single Item Lot-Sizing Problems. Ph.D. Dissertation, Erasmus University Rotterdam, Rotterdam, The Netherlands.

VAN HOESEL, C. P. M., A. W. J. KOLEN AND A. P. M. WAGELMANS. 1991. A Dual Algorithm for the Economic Lot-Sizing Problem. *Eur. J. Opnl. Res.* **52,** 315–325.

WAGNER, H. M. 1960. A Postscript to Dynamic Problems in the Theory of the Firm. *Naval Res. Logist. Quart.* **7,** 7–12.

WAGNER, H. M., AND T. M. WHITIN. 1958. Dynamic Version of the Economic Lot Size Model. *Mgmt. Sci.* **5,** 89–96.

ZABEL, E. 1964. Some Generalizations of an Inventory Planning Horizon Theorem. *Mgmt. Sci.* **10,** 465–471.

ZANGWILL, W. I. 1968. Minimum Concave Cost Flows in Certain Networks. *Mgmt. Sci.* **14,** 429–450.