Discrete Applied Mathematics 45 (1993) 291–312
North-Holland

Sensitivity analysis of the economic lot-sizing problem

Stan Van Hoesel* and Albert Wagelmans**

Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, Netherlands

Received 17 December 1990
Revised 9 August 1991

Abstract

Van Hoesel, S. and A. Wagelmans, Sensitivity analysis of the economic lot-sizing problem, Discrete Applied Mathematics 45 (1993) 291–312.

In this paper we study sensitivity analysis of the uncapacitated single level economic lot-sizing problem, which was introduced by Wagner and Whitin about thirty years ago. In particular we are concerned with the computation of the maximal ranges in which the numerical problem parameters may vary individually, such that a solution already obtained remains optimal. Only recently it was discovered that faster algorithms than the Wagner-Whitin algorithm exist to solve the economic lot-sizing problem. Moreover, these algorithms reveal that the problem has more structure than was recognized so far. When performing the sensitivity analysis we exploit these newly obtained insights.

Keywords. Economic lot-sizing, sensitivity analysis.

1. Introduction

In 1958 Wagner and Whitin published their seminal paper on the "Dynamic version of the economic lot size model", in which they proposed a dynamic programming algorithm that solves the problem considered in $O(n^2)$ time, n being the length of the planning horizon. It is well known that the same approach also solves

Correspondence to: Dr. A.P.M. Wagelmans, Econometric Institute, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, Netherlands.

* Supported by the Netherlands Organization for Scientific Research (NWO) under grant no. 611-304-017.

** On leave at the Operations Research Center, Room E 40-164, Massachusetts Institute of Technology, Cambridge, MA 02139; financial support of the Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged.

a slightly more general problem to which we will refer as the economic lot-sizing problem (ELS). In the last 30 years the research on this problem has concentrated on efficient implementations of the Wagner–Whitin algorithm, mainly through the use of so-called planning horizon theorems (see for instance Zabel [23], Eppen, Gould and Pashigian [6], Lundin and Morton [10], Evans [7] and Saydam and McKnew [15]). This did not result in an algorithm with a better complexity than $O(n^2)$ and therefore another line of research focused on the design and analysis of faster heuristics (see for instance Axsäter [3], Bitran, Magnanti and Yanasse [5] and Baker [4]).

Recently however, it was discovered independently by Aggarwal and Park [1], Federgruen and Tzur [8] and Wagelmans, Van Hoesel and Kolen [20] that the economic lot-sizing problem can be solved in $O(n \log n)$ time and in some nontrivial special cases even in linear time. This is surprising, because ELS is usually modeled as a shortest path problem on a network with $\Omega(n^2)$ arcs. Some of the new algorithms provide additional insight in the structure of ELS. In particular, this holds for the algorithm presented by Wagelmans, Van Hoesel and Kolen which has a rather transparent geometrical interpretation.

Typically, instances of ELS arise in environments that are highly dynamic. Most problem parameters may be subject to change. For instance, at the time that one has to come up with a production plan, only estimates of some parameters values may be available, while their true value becomes known later on. If the latter values had been known in advance, one would possibly have decided upon a different production plan. The maximal deviation of the true value from the estimated value of a parameter such that these production plans do not differ can be viewed as a measure of the stability of the proposed solution with respect to the parameter under consideration. The popularity of the use of heuristics is partly explained by the fact that they tend to produce solutions that are more stable than the optimal production plan. However, before applying a heuristic it may be worthwhile to find out how stable an optimal solution is with respect to changes in the problem parameters. Hence, the issue of sensitivity analysis arises quite naturally.

Sensitivity analysis of simple lot-sizing problems is studied in Richter [12], Richter and Vörös [13] and Van Hoesel and Wagelmans [18]. These papers were mainly concerned with *simultaneous* changes of parameters, i.e., one tries to characterize and determine the maximal region in the space of changing parameters such that a given optimal solution is optimal for all parameter combinations in that region. (Related results are presented in Richter [11] and Richter and Vörös [14].) In this paper we will exploit the new insights in the structure of ELS to compute the maximal ranges in which the numerical problem parameters may vary *individually*, such that an optimal production plan, obtained by the Wagelmans–Van Hoesel–Kolen algorithm, remains optimal. Lee [9] presents a theoretical framework to perform similar analyses on general dynamic programming problems and he gives an application to the lot-sizing problem considered here. However, he does not focus on the computational aspects of his approach. The basic concept of his framework is the construction of a so-called penalty network. For the lot-sizing problem this construction requires already $\Omega(n^2)$ time, while most of our algorithms have a lower running time.

This paper is organized as follows. In Section 2 we discuss ELS and present two $O(n \log n)$ algorithms, corresponding to a backward and a forward dynamic programming formulation of the problem. In Section 3 we prove some preliminary results that are useful in Section 4 where the actual sensitivity analysis is performed. Section 5 contains concluding remarks.

2. The economic lot-sizing problem

2.1. Definition and formulations

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.

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

x_i: number of units produced in period i,

s_i: 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 $d_{ij} = \sum_{t=i}^{j} d_t$, $1 \le i \le j \le n$, then a correct formulation of ELS is

$$\begin{array}{lll} \min & \sum_{i=1}^{n} (p_i x_i + f_i y_i + h_i s_i), & \\ \text{s.t.} & x_i + s_{i-1} - s_i = d_i & \text{for } i=1,\dots,n, \\ & d_{in} y_i - x_i \ge 0 & \text{for } i=1,\dots,n, \\ & s_0 = s_n = 0, & \\ & x_i \ge 0,\ s_i \ge 0,\ y_i \in \{0,1\} & \text{for } i=1,\dots,n. \end{array} \tag{I}$$

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

$$
\begin{array}{lll}
\min & \sum_{i=1}^{n} (c_i x_i + f_i y_i) - \sum_{i=1}^{n} h_i d_{1i}, & \\
\text{s.t.} & \sum_{t=1}^{n} x_t = d_{1n}, & \\
& \sum_{t=1}^{i} x_t \geq d_{1i} & \text{for } i = 1, \dots, n-1, \\
& d_{in} y_i - x_i \geq 0 & \text{for } i = 1, \dots, n, \\
& x_i \geq 0,\ y_i \in \{0,1\} & \text{for } i = 1, \dots, n.
\end{array}
\tag{II}
$$

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

From now on we will work with the marginal production costs c_i, $i = 1, \dots, n$, and objective function $\sum_{i=1}^{n} (c_i x_i + f_i y_i)$. Note that we have not made any assumption about the sign of the marginal production 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 algorithms that we will present assume 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 yields the optimal value of the original instance. Therefore, we assume from now on that all setup costs are nonnegative.

2.2. $O(n \log n)$ *algorithms*

Before presenting our algorithms we should point out that the goal of this subsection is to explain the essential ideas of the algorithms and to introduce basic techniques that will also be used when performing the sensitivity analysis. Therefore our exposition will be mainly geometrical, and for convenience we assume for the moment that d_i is strictly positive for all $i = 1, \dots, n$. For a more detailed presentation we refer to Wagelmans, Van Hoesel and Kolen [20].

Traditionally, ELS is not solved by explicitly using any of the mathematical formulations given in Subsection 2.1, but by means of dynamic programming. The key observation to obtain a dynamic programming formulation of the problem is that it suffices to consider only feasible solutions that have the *zero-inventory property*, i.e., solutions in which the inventory at the beginning of production periods is zero. The latter implies that production in a period i equals 0 or d_{ik} for some $k \geq i$. The zero-inventory property was stated first by Wagner and Whitin [22] for the special case they considered. Later Wagner [21] showed that the property even holds under the assumption of concave production costs (see also Zangwill [24]).

First we present an algorithm that is essentially a backward dynamic programming algorithm. Define $G(i)$ to be the cost of an optimal solution to the instance of ELS with planning horizon consisting of periods i to n, $i = 1, \ldots, n$. Furthermore, $G(n+1)$ is defined to be zero. If the planning horizon starts in period i, then we will always produce in this period and the setup cost f_i will be incurred. Assume that the next production period is $t > i$, then exactly $d_{i,t-1}$ units will be produced (because of the zero-inventory property). Therefore, the following recursion holds:

$$G(i) := f_i + \min_{i < t \leq n+1} \{c_i d_{i,t-1} + G(t)\} \quad \text{for all } i = 1, \ldots, n. \tag{1}$$

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. However, we will show that given $G(t)$ for $t = i+1, \ldots, n+1$, it is possible to determine $\min_{t>i} \{c_i d_{i,t-1} + G(t)\}$ in $O(\log n)$ time. Because of (1) this implies that $G(i)$ can be determined in $O(\log n)$ time.

To start the exposition we plot the points $(d_{tn}, G(t))$ for $t = i+1, \ldots, n+1$, like in Fig. 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 $(d_{n+1,n}, G(n+1)) \equiv (0, 0)$. The curve LE is the lower convex envelope of the plotted points.

Now consider Fig. 2 where we have drawn the line with slope c_i that passes through an arbitrary point $(d_{jn}, G(j))$. The coordinate on the vertical axis of the intersection point of this line with the vertical line through $(d_{in}, 0)$ is exactly $c_i d_{i,j-1} + G(j)$.

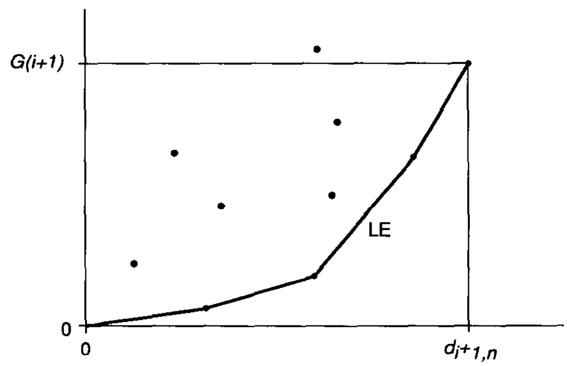

Fig. 1.

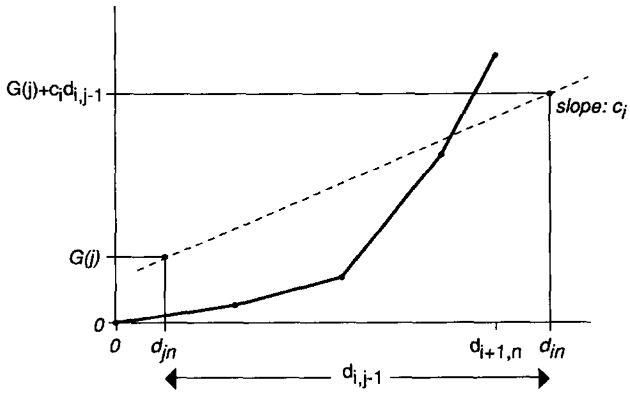

Fig. 2.

Hence, to determine $\min_{t>i}\{c_i d_{i,t-1}+G(t)\}$ we can proceed as follows (see Fig. 3): for every period $t>i$ we determine the intersection point of the vertical line through $(d_{in}, 0)$ and the line with slope c_i that passes through $(d_{tn}, G(t))$. The coordinate on the vertical axis of the lowest of these intersection points is equal to $\min_{t>i}\{c_i d_{i,t-1}+G(t)\}$.

From Fig. 3 it is clear that we are in fact looking for the line with slope c_i that is tangent to LE. This means that the period for which the minimum is attained corresponds to an extreme point of LE. Moreover, this point has the property that the slope of LE to the left of it is at most c_i, while the slope to the right is at least c_i. Because LE is convex, the slopes of its line segments are ordered and therefore an extreme point that corresponds to the minimum can be identified in $O(\log n)$ time by binary search. Hence, given LE, the value $\min_{t>i}\{c_i d_{i,t-1}+G(t)\}$, and thus $G(i)$, can be determined in $O(\log n)$ time.

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 convex lower envelope. Geometrically we can apply the following procedure (see Fig. 4): add the

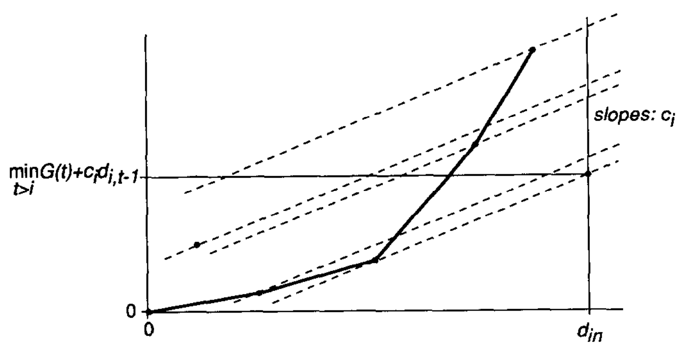

Fig. 3.

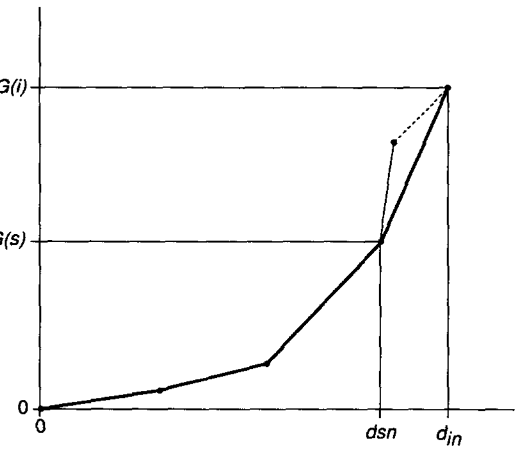

Fig. 4.

point $(d_{in}, G(i))$ and find the rightmost extreme point $(d_{sn}, G(s))$ of LE such that the slope of the line segment connecting $(d_{in}, G(i))$ to $(d_{sn}, G(s))$ is greater than the slope of the line segment to the left of $(d_{sn}, G(s))$. To find s we simply start with the rightmost extreme point of LE and work towards the left until we conclude that the desired extreme point has been found.

The complete algorithm should be clear by now: there are n iterations and $G(i)$ is calculated in iteration $n-i+1$. $G(1)$ is equal to the value of an optimal solution and the solution itself can be easily constructed if we have stored for every i the period $t>i$ for which $c_i d_{i,t-1} + G(t)$ is minimal. One can prove that all production periods of the optimal solution appear as extreme points in the final convex lower envelope. Note that recursion (1) is not valid if we allow demands to be zero, because then we do not automatically have a setup in the first period of the planning horizon. However, one can show that only a slight modification is needed to ensure that the approach described also works in the presence of zero demand.

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. Furthermore, it is not necessary to calculate d_{ij} for all pairs i, j with $1 \le i \le j \le n$. We only need to calculate the coefficients d_{in}, $i=1,\ldots,n$, which again takes linear time. This preprocessing enables us to calculate a coefficient d_{ij} in constant time, whenever necessary, since $d_{ij} = d_{in} - d_{j+1,n}$.

To keep track of the convex lower envelope we can simply use a stack which contains the periods corresponding to the extreme points. Note that every period is added and deleted to the stack at most once and that both operations take constant time. As noted before, it takes $O(\log n)$ time to perform a binary search among the periods in the stack. Because there are n iterations, the total time spent on searching is $O(n \log n)$. In every iteration we have to make a few comparisons to update the

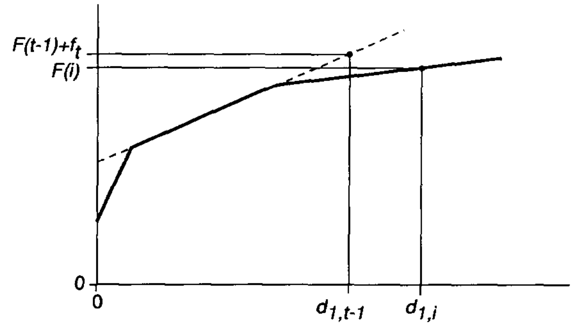

Fig. 5.

convex lower envelope. After every comparison we either conclude that we have found the new convex lower envelope or that we have to make at least one more comparison. The first case occurs exactly once in every iteration, i.e., in total n times. In the second case we delete a period from the stack. As every period is deleted from the stack at most once, this case can occur no more than n times. Thus, the overall complexity of calculating $G(1)$ is $O(n \log n)$.

We will now briefly discuss an $O(n \log n)$ algorithm that uses a forward recursion.[1] Let the variables $F(i)$, $i=1,\dots,n$, denote the value of the optimal production plan for the instance of ELS with planning horizon consisting of the periods $1,\dots,i$. Defining $F(0)\equiv 0$ we have the following recursion

$$F(i) := \min_{0<t\le i} \{F(t-1)+f_t+c_t d_{ti}\}.$$

To determine $F(i)$ when $F(t-1)$ is given for all $t\le i$, we can proceed as follows (see Fig. 5): for each $t\le i$ we plot the point $(d_{1,t-1}, F(t-1)+f_t)$ and draw the line with slope c_t that passes through this point. Now it is easy to verify that $F(i)$ is equal to the value of the concave lower envelope of these lines in coordinate d_{1i} on the horizontal axis. After constructing the line with slope c_i that passes through $(d_{1i}, F(i)+f_{i+1})$, we update the lower envelope and continue with the determination of $F(i+1)$.

Hence, the running time of this algorithm depends on the complexity of evaluating the lower envelope for a given point on the horizontal axis and the complexity of updating the concave lower envelope. If a *balanced tree* (see Aho, Hopcroft and Ullman [2]) is used to store the breakpoints and the corresponding slopes of the linear parts of the lower envelope, then one can show that the complexity can be bounded by $O(n \log n)$.

To conclude this section we mention that the described algorithms can be modified to run in linear time for some nontrivial special cases of ELS. In particular

[1] Actually one can view the backward and the forward algorithm as applications of exactly the same technique, that can be presented in different ways. For details see Van Hoesel [17].

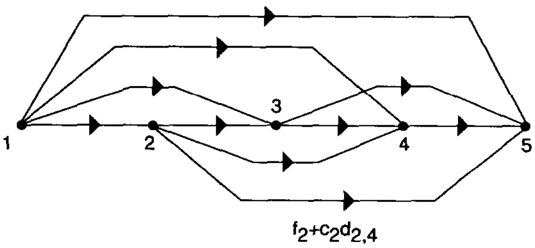

Fig. 6. Dynamic programming network for $n=4$.

this holds if $c_i \geq c_{i+1}$ for all $i=1,\dots,n-1$, which is for instance the case if $p_i=p$ and $h_i \geq 0$ for all $i=1,\dots,n$.

3. Preliminary results for the sensitivity analysis

This section contains some lemmas which are useful in Section 4, where the sensitivity analysis is performed. The following dynamic programming network facilitates the exposition (although it is never actually constructed in the algorithms to be presented):

The vertex set is $\{1,2,\dots,n+1\}$; the set of arcs is $\{(i,j) \mid 1 \leq i<j \leq n+1\}$ and the length of arc (i,j) is equal to $l_{ij} \equiv f_i + c_i d_{i,j-1}$ (see Fig. 6 for the case $n=4$).

From our dynamic programming formulations in the preceding section, it follows immediately that for all $i=1,\dots,n$ the length of a shortest path from i to $n+1$ in this network is equal to $G(i)$ and the length of a shortest path from 1 to i equals $F(i-1)$.[2] Moreover, the following holds:

Lemma 3.1. *$F(j-1) \leq F(i-1)+l_{ij}$ and $G(i) \leq l_{ij}+G(j)$ for all i, j, $1 \leq i<j \leq n+1$.*

Suppose that for every i, $1 \leq i \leq n+1$, we have determined a shortest path from 1 to i and a shortest path from i to $n+1$ in the DP-network. Let i and j be two periods $(1 \leq i<j \leq n+1)$. j is called the *successor* of i if j immediately successes i on the shortest path from i to $n+1$. It follows that $G(i)=l_{ij}+G(j)$. We denote the successor of i by $sc(i)$. Analogously, i is called the *predecessor* of j if i immediately precedes j on the shortest path from 1 to j. Hence, $F(j-1)=F(i-1)+l_{ij}$. The predecessor of j is denoted by $pr(j)$.

Lemma 3.2. (a) *If $k=sc(j)$ and $j=sc(i)$ then $l_{ij}+l_{jk} \leq l_{ik}$.*
(b) *If $i=pr(j)$ and $j=pr(k)$ then $l_{ij}+l_{jk} \leq l_{ik}$.*

Proof. (a) From $j=sc(i)$ it follows that $G(i)=l_{ij}+G(j)$, and $k=sc(j)$ implies

[2] In the sequel we will use "path" and "production plan" as well as "length" and "cost" as synonyms.

$G(j)=l_{jk}+G(k)$. Combining these equalities results in $G(i)=l_{ij}+l_{jk}+G(k)$. Now the desired inequality follows from $G(i)\leq l_{ik}+G(k)$ (Lemma 3.1).

(b) Analogously to the proof of part (a). $\square$

Lemma 3.3. (a) *If $j=sc(i)$ then $F(i-1)+G(i)\geq F(j-1)+G(j)$.*

(b) *If $i=pr(j)$ then $F(i-1)+G(i)\leq F(j-1)+G(j)$.*

Proof. (a) $F(i-1)+G(i)=F(i-1)+l_{ij}+G(j)\geq F(j-1)+G(j)$, where the inequality follows from Lemma 3.1.

(b) Analogously to the proof of part (a). $\square$

Convention. We assume that if $k=sc(j)$ then $f_j>0$ or $d_{j,k-1}>0$.

This convention excludes degenerate optimal solutions in which period j is declared to be an intermediate production period, while actually nothing is produced in that period. It is easy to adapt the algorithms given in the preceding section such that degenerate solutions are never generated. Otherwise, it takes linear time to transform a set of optimal solutions in a set of nondegenerate optimal solutions by redefining some successors. Moreover, we only need the convention to facilitate the proofs of some results that also hold in general.

Lemma 3.4. *If $j\leq n$ and $j=sc(i)$ then $c_i\geq c_j$.*

Proof. By Lemma 3.2, $l_{ij}+l_{j,sc(j)}\leq l_{i,sc(j)}$ and therefore (after rewriting) $f_j\leq (c_i-c_j)d_{j,sc(j)-1}$. Since by our convention $f_j>0$ or $d_{j,sc(j)-1}>0$ it follows that $c_i\geq c_j$. $\square$

Lemma 3.5. *Let $j\leq n$ and $j=sc(i)$. Then $F(i-1)+l_{it}\geq F(j-1)+l_{jt}$ for all $t>sc(j)$.*

Proof.

$$
\begin{aligned}
&F(i-1)+l_{it} \\
&\quad =F(i-1)+l_{i,sc(j)}+c_i d_{sc(j),t-1} && \text{(definitions of } l_{it} \text{ and } l_{i,sc(j)}) \\
&\quad \geq F(i-1)+l_{i,sc(j)}+c_j d_{sc(j),t-1} && \text{(Lemma 3.4)} \\
&\quad \geq F(i-1)+l_{ij}+l_{j,sc(j)}+c_j d_{sc(j),t-1} && \text{(Lemma 3.2)} \\
&\quad \geq F(j-1)+l_{j,sc(j)}+c_j d_{sc(j),t-1} && \text{(Lemma 3.1)} \\
&\quad =F(j-1)+l_{jt} && \text{(definitions of } l_{jt} \text{ and } l_{j,sc(j)}). \quad \square
\end{aligned}
$$

4. Sensitivity analysis

In this section we give algorithms to calculate for all the numerical problem parameters the maximal ranges in which they can vary individually such that an optimal solution already obtained remains optimal. In most of the algorithms these (individual) ranges of the problem parameters are calculated simultaneously for all periods. For instance, we will present an algorithm that computes simultaneously the maximal allowable increases of all the coefficients f_i, $i=1,\dots,n$, in $O(n \log n)$ time. We assume that all the relevant information from the forward and backward dynamic programming algorithms is available, i.e., the values $F(i-1)$, $G(i)$ for $i=1,\dots,n+1$, the periods $sc(i)$ and $pr(i)$ for $i=1,\dots,n$, the final convex lower envelope associated with the backward recursion and finally the optimal production schedule.

The parameters are divided into three sets depending on the set of arcs that change cost if the parameter is altered.

Set I: f_i, c_i, p_i, $i=1,\dots,n$.

If for a given i one of these parameters changes then exactly the arcs that have i as a tail will change in cost. If f_i changes by δ, then all these arcs will also change by δ in cost. If c_i or p_i changes by δ, then the arcs will have a cost change depending on the cumulative demand: the cost of arc (i,j) will change by $\delta d_{i,j-1}$. Note that a change of p_i by δ is a special case of changing c_i by δ. That the latter is indeed more general follows from the fact that, for instance, changing h_i by δ and h_{i-1} by $-\delta$ results in a change of c_i by δ while leaving the coefficients c_t, $t \neq i$, unaltered. In the sequel changes in p_i are implicitly dealt with by analyzing changes in c_i.

Set II: h_i, $i=1,\dots,n$.

If h_i changes by δ then all c_j, $j \leq i$, are perturbed by δ because $c_j = p_j + \sum_{t \geq j} h_t$. Therefore, the costs of arcs with tail in $\{1,\dots,i\}$ are changed by an amount depending on the cumulative demand: $\delta d_{j,k-1}$ for arc (j,k), where $j \leq i$.

Set III: d_i, $i=1,\dots,n$.

If d_i is perturbed then all arc costs in which the demand of period i is involved will change. These are the arcs (j,k) where $j \leq i$ and $k > i$. The cost change of such an arc is δc_j, where δ is the change in d_i.

In the following subsections we treat each of these sets separately. Furthermore, we distinguish between increases and decreases of parameters, since it turns out that these two cases have to be treated differently.

4.1. Sensitivity analysis of the setup and marginal production costs

Suppose f_i or c_i is changed by δ. The shortest path from 1 to i in the DP-network remains unchanged, and thus its cost is $F(i-1)$. Moreover, the paths not through i do not have a change in cost either. On the other hand, the costs of all paths from i to $n+1$ change.

We first consider cost decreases.

Case 1: *f_i decreases to $f_i - \delta$.*

The optimal path from 1 to i remains the same with cost $F(i-1)$ and the optimal path from i to $n+1$ remains the same with cost $G(i)-\delta$.

If i is a production period in the optimal schedule then this will certainly also hold after the cost change. The cost of the optimal schedule is $F(n)-\delta=F(i-1)+G(i)-\delta$. The only upper bound on δ is imposed by the nonnegativity of f_i. Thus, δ is bounded by f_i.

If i is not a production period in the optimal schedule then the shortest path from 1 to $n+1$ does not pass through i. This path has value $G(1)$ and the shortest path through i has value $F(i-1)+G(i)-\delta$. The latter path is shorter if $F(i-1)+G(i)-\delta<G(1)$, so the optimal path does not change for $\delta \leq F(i-1)+G(i)-G(1)$. Because of the nonnegativity of f_i, δ is bounded by $\min\{f_i, F(i-1)+G(i)-G(1)\}$. We have shown our first complexity result.

Theorem 4.1. *The maximal allowable decrease of f_i can be calculated in constant time for each i, $i=1,\ldots,n$.*

Case 2: *c_i decreases to $c_i-\delta$.*

If i is a production period in the optimal schedule this will remain so, since only paths that contain i have a decrease in cost. However, the shortest path from i to $n+1$ may change. Let $j=sc(i)$, then we have to determine the maximal value of δ such that j is still the successor of i. To this end we consider the convex lower envelope in Fig. 7.

As we have already noted in Section 2, all the production periods of the optimal schedule appear in the final lower convex envelope. In particular this holds for the periods i and j. Clearly all the periods $t \geq j$ that appear in the final convex lower envelope were already present when $G(i)$ was determined. Moreover, no period $t \geq j$ that appeared in the convex lower envelope at that time has been removed from the lower envelope in the meantime, because that would imply that j has also been removed. Hence, the convex lower envelope corresponding to the periods $t \geq j$ is im-

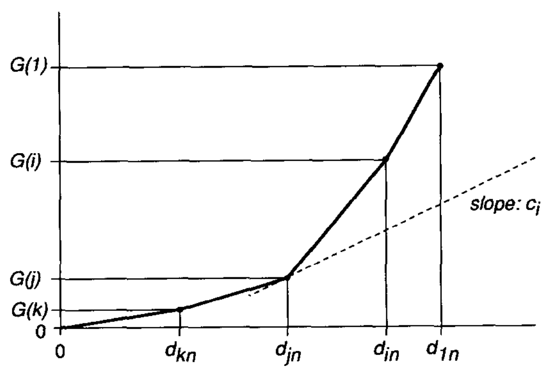

Fig. 7. Final convex lower envelope.

mediately available from the final lower envelope in Fig. 7. Therefore, j remains the successor of i as long as $c_i - \delta \geq \{G(j) - G(k)\}/d_{j,k-1}$, where k is the smallest period which appears in the lower envelope and is greater than j. It follows that the maximal allowable decrease of c_i and the new successor of i can be determined in constant time.

We now turn to the case that i is not a production period in the optimal solution. Because the cost of an arc (i,t) is altered by $\delta d_{i,t-1}$, the optimal path from i to $n+1$ has value $f_i + \min_{t>i}\{(c_i - \delta)d_{i,t-1} + G(t)\}$. Period i will not be a production period in an optimal schedule as long as $F(i-1) + f_i + \min_{t>i}\{(c_i - \delta)d_{i,t-1} + G(t)\} > F(n)$. Hence, the maximal allowable decrease of c_i is the value of δ for which $\min_{t>i}\{(c_i - \delta)d_{i,t-1} + G(t)\} = F(n) - F(i-1) - f_i$.

Note that the period for which the minimum is attained when δ equals the maximal allowable decrease, is the *possibly new* successor of i in the optimal path from 1 to $n+1$ through i. It follows that if δ is equal to the maximal allowable decrease, both i and this period appear in the *new* final lower convex envelope. The crucial observation to be made here is that we only have to consider the periods $t>i$ that appear in the *already known* final convex lower envelope. This follows from the fact that if c_i is decreased, then the values $G(t)$, $t \leq i$, do not increase and the values $G(t)$, $t>i$, remain the same. Therefore, the points $(d_{tn}, G(t))$, $t>i$, that do not belong to the known lower envelope can certainly not be present in the new lower envelope, since the latter does not lie above the former.

We now arrive at the actual computation of the maximal allowable decrease of c_i. Consider the final convex lower envelope restricted to the periods $t>i$. In the backward algorithm described in Subsection 2.2 we determined the line with given slope that is tangent to this lower envelope; the value of this line in coordinate d_{in} was the minimum value we were looking for. Now this last value is given, namely $F(n) - F(i-1) - f_i$, and we have to determine the slope of the line that is tangent to the lower envelope and passes through the point $(d_{in}, F(n) - F(i-1) - f_i)$ (see Fig. 8). This is easily seen to take $O(\log n)$ time by binary search. Moreover, the slope of this tangent gives us the minimum value of $c_i - \delta$ for which the optimal schedule does not change and thus the maximum value of δ.

To summarize, we have the following result.

Theorem 4.2. *If i is a production period in the optimal schedule, then the maximal allowable decrease of c_i can be calculated in constant time; otherwise,* $O(\log n)$ *time suffices.*

We will now consider increasing cost coefficients. For a period i that is not a production period in the optimal schedule, the coefficients f_i and c_i can be increased arbitrarily without causing the optimal solution to change. This follows trivially from the fact that $G(i)$ increases while $G(1)$ remains constant and therefore $F(i-1) + G(i) \geq G(1)$ continues to hold. Hence we only have to consider the production periods of the optimal schedule. Let $i \neq 1$ be such a period, then the following

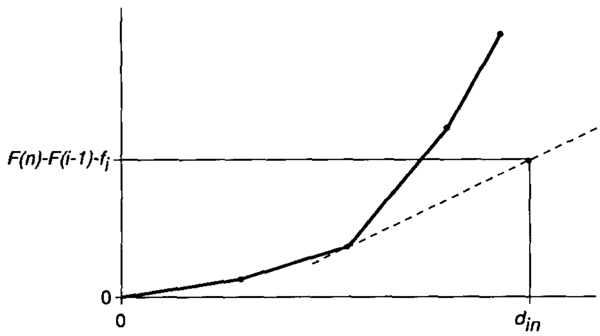

Fig. 8. Determination of maximal allowable decrease when i is not a production period.

value determines the optimal path from 1 to $n+1$ in the DP-network that does not pass through i:

$$M_i \equiv \min_{j<i<t} \{F(j-1)+l_{jt}+G(t)\}.$$

Now $M_i \equiv \min_{j<i}\{M_{ji}\}$, where for $j<i$

$$M_{ji} \equiv \min_{t>i} \{F(j-1)+l_{jt}+G(t)\}.$$

Before giving algorithms to calculate the maximal allowable increases of f_i and c_i, we will first show how to calculate M_i for all production periods $i \neq 1$ of the optimal schedule simultaneously in $O(n \log n)$ time. To this end, we partition the periods before i into two sets:

$$I_i^0 \equiv \{j<i \mid i \text{ is not on the shortest path from } j \text{ to } n+1\},$$

$$I_i^1 \equiv \{j<i \mid i \text{ is on the shortest path from } j \text{ to } n+1\}.$$

We define $M_i^0 \equiv \min_{j \in I_i^0}\{M_{ji}\}$ and $M_i^1 \equiv \min_{j \in I_i^1}\{M_{ji}\}$; clearly, $M_i = \min\{M_i^0, M_i^1\}$. First we focus on the computation of M_i^0.

Lemma 4.3. *Suppose $j \in I_i^0$ and let $k<i$ such that $sc(k)>i$ and k on the shortest path from j to $n+1$, then $M_{ji} \geq F(j-1)+G(j) \geq F(k-1)+G(k)=M_{ki}$.*

Proof. The first inequality follows from

$$M_{ji} = \min_{t>i} \{F(j-1)+l_{jt}+G(t)\} \geq \min_{t>j} \{F(j-1)+l_{jt}+G(t)\}$$

$$= F(j-1) + \min_{t>j} \{l_{jt}+G(t)\} = F(j-1)+G(j).$$

The second inequality follows by induction from Lemma 3.3(a) since k is on the optimal path from j to $n+1$. Finally, the equality follows from the fact that $sc(k)>i$:

$$\begin{aligned}M_{ki} &= \min_{t>i} \{F(k-1)+l_{kt}+G(t)\}\\ &\le F(k-1)+l_{k,sc(k)}+G(sc(k))\\ &= F(k-1)+G(k)\\ &= \min_{t>k} \{F(k-1)+l_{kt}+G(t)\}\\ &\le \min_{t>i} \{F(k-1)+l_{kt}+G(t)\} = M_{ki}. \qquad \square\end{aligned}$$

From Lemma 4.3 it follows that calculating M_i^0 is equivalent to determining the minimum of $\{F(j-1)+G(j) \mid j<i \text{ and } sc(j)>i\}$; note that we do not have to require explicitly $j\in I_i^0$. Using this fact, we are able to compute values M_i^0 for all production periods $i\neq 1$ of the optimal schedule simultaneously in $O(n \log n)$ time by the algorithm given next.

The data structure that we will use is a *binary heap* which has the following properties (see Aho, Hopcroft and Ullman [2]): the minimum of the values stored in the heap can be found in constant time; if at most n values are stored then it takes $O(\log n)$ time to update the heap after the addition or deletion of an element. Let i_2 be the smallest production period greater than 1. In the initial heap we store the values $F(j-1)+G(j)$ for all $j<i_2$. To find $M_{i_2}^0$ we consider the period j^* corresponding to the minimum of these values. If $sc(j^*)>i_2$, then $M_{i_2}^0=F(j^*-1)+G(j^*)$; otherwise we delete $F(j^*-1)+G(j^*)$ from the heap. We repeat this step until the minimum is attained for a period which has a successor greater than i_2. After $M_{i_2}^0$ has been determined in this way we add to the heap the values $F(j-1)+G(j)$ for all j with $i_2<j<i_3$, where i_3 is the next production period and determine $M_{i_3}^0$ analogously. We proceed in this way until all desired values have been calculated. The time bound follows from the fact that in total only n additions and at most n deletions take place.

We now come to the determination of the values M_i^1 for all production periods $i>1$. The following lemma states a similar result as Lemma 4.3.

Lemma 4.4. *Let $j\in I_i^1$ and $k=sc(j)<i$, then $M_{ji}\ge M_{ki}$.*

Proof. From Lemma 3.5 it follows that

$$F(j-1)+l_{jt}+G(t)\ge F(k-1)+l_{kt}+G(t) \quad \text{for all } t>sc(k).$$

Now $j\in I_i^1$ implies that $sc(k)\le i$. Hence, the inequality certainly holds for all $t>i$. Moreover, by definition we have

$$F(k-1)+l_{kt}+G(t)\ge M_{ki} \quad \text{for all } t>i.$$

Combining the inequalities gives

$$F(j-1)+l_{jt}+G(t)\ge M_{ki} \quad \text{for all } t>i.$$

Therefore, $M_{ji} = \min_{t>i}\{F(j-1) + l_{jt} + G(t)\} \geq M_{ki}$. $\square$

From Lemma 4.4 we deduce that to calculate M_i^1 it is sufficient to determine the minimum of $\{M_{ji} \mid sc(j) = i\}$. Now consider a j with $sc(j) = i$. By definition $M_{ji} = F(j-1) + \min_{t>i}\{l_{jt} + G(t)\}$, and to evaluate the minimum in this expression it suffices to determine the line with slope c_j that is tangent to the convex lower envelope of the points $(d_{tn}, G(t))$ for $t > i$. Hence, M_{ji} can be calculated in $O(\log n)$ time and M_i^1 can be determined in $O(m_i \log n)$ time, where $m_i = |\{j \mid sc(j) = i\}|$. Since $\sum_{i=1}^{n} m_i \leq n$, computing M_i^1 for all relevant periods takes $O(n \log n)$ time.

In general we do not have the convex lower envelope of the points $(d_{tn}, G(t))$, $t > i$, immediately available for all production periods i. However, we can just construct these lower envelopes for decreasing i by the method given in Section 2. As we have seen, this takes overall $O(n \log n)$ time. Moreover, determining the set of periods j with $sc(j) = i$ can be done in $O(n)$ time simultaneously, since $sc(j)$ is given for all $j = 1, \ldots, n$.

Note that calculating M_i^0 is done in a forward fashion while calculating M_i^1 is done backwards. Finally we set $M_i := \min\{M_i^0, M_i^1\}$, which finishes the whole process of calculating simultaneously for all production periods $i > 1$ of the optimal schedule the value of the shortest path from 1 to $n+1$ not containing i.

The calculations above do not concern $i = 1$. If $d_i > 0$ then period 1 is always a production period and we define $M_1 \equiv \infty$. If $d_1 = 0$ and period 1 is a production period in the optimal schedule, then we define $M_1 \equiv G(2)$.

We now proceed with the calculations of the maximal allowable increases of f_i and c_i for a production period i.

Case 3: *f_i increases to $f_i + \delta$.*

If i is a production period in the optimal schedule, then the shortest path through i from 1 to $n+1$ has value $F(n) + \delta$. Now the corresponding schedule will remain optimal as long as there is no better schedule without i as production period, i.e., as long as $F(n) + \delta \leq M_i$. So the maximal allowable increase equals $M_i - F(n)$.

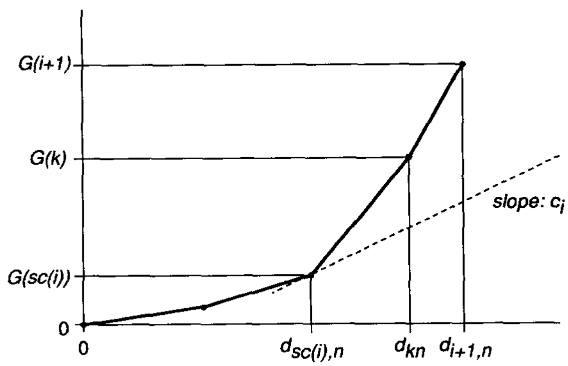

Fig. 9. Convex lower envelope for periods after i.

Case 4: c_i *increases to* $c_i + \delta$.

Arc (i, j), $j > i$, increases in cost by $\delta d_{i,j-1}$. If i is a production period in the optimal schedule, then the value of this schedule is now $F(n) + \delta d_{i,sc(i)-1}$. An upper bound on the maximal allowable increase is $\{M_i - F(n)\}/d_{i,sc(i)-1}$, which follows from $F(n) + \delta d_{i,sc(i)-1} \leq M_i$.

However, the optimal schedule can also change if it gets more attractive to take an earlier successor of i. To determine the smallest value of δ for which this happens we can use again the convex lower envelope of the points $(d_{tn}, G(t))$, $t > i$ (see Fig. 9); the relevant information is given by the slope $\{G(k) - G(sc(i))\}/d_{k,sc(i)-1}$ of the linear part to the right of $(d_{sc(i),n}, G(sc(i)))$.

Note that $\{G(k) - G(sc(i))\}/d_{k,sc(i)-1}$ can be obtained at the same time that M_i^1 is determined. It follows that $-c_i + \{G(k) - G(sc(i))\}/d_{k,sc(i)-1}$ is also an upper bound on the maximal allowable increase. The latter value is equal to the minimum of both bounds.

To summarize, we have shown the following.

Theorem 4.5. *The maximal allowable increases of* f_i *and* c_i *can be calculated for all* $i = 1, \ldots, n$ *simultaneously in* $O(n \log n)$ *time.*

4.2. Sensitivity analysis of the holding costs

Since $c_j = p_j + \sum_{t \geq j} h_t$, changing h_i to $h_i + \delta$ results in changing c_j by δ for all $j \leq i$. Now let $1 = i_1 < i_2 < \cdots < i_{r-1} \leq i < i_r < \cdots < i_k < i_{k+1} = n + 1$ denote the production periods of an arbitrary production schedule, then for this schedule the cost change equals $\delta d_{1,i_r - 1}$. So the value of an optimal schedule is given by

$$V_i(\delta) \equiv \min_{t \leq i < j} \{F(t-1) + l_{tj} + G(j) + \delta d_{1,j-1}\}.$$

Case 5: h_i *increases to* $h_i + \delta$.

For $1 \leq i < j \leq n$ we define $V_{ij}(\delta) \equiv \min_{t \leq i}\{F(t-1) + l_{tj}\} + G(j) + \delta d_{1,j-1}$. By definition $V_i(\delta) = \min_{j > i}\{V_{ij}\{\delta\}\}$. Moreover, we can show the following relation.

Lemma 4.6. $\min_{j>i}\{V_{ij}(\delta)\} = \min_{j>t}\{F(j-1) + G(j) + \delta d_{1,j-1}\}$ *for* $i = 1, \ldots, n-1$.

Proof. Let $j > i$ then clearly

$$\min_{t \leq i} \{F(t-1) + l_{tj}\} \geq \min_{t < j} \{F(t-1) + l_{tj}\} = F(j-1) \tag{2}$$

and therefore

$$V_{ij}(\delta) \geq F(j-1) + G(j) + \delta d_{1,j-1}. \tag{3}$$

First suppose $pr(j) \leq i$ then, using (2), we obtain

$$F(j-1) = F(pr(j) - 1) + l_{pr(j),j} \geq \min_{t \leq i} \{F(t-1) + l_{tj}\} \geq F(j-1).$$

It follows that $\min_{t \le i}\{F(t-1)+l_{tj}\}=F(j-1)$ and therefore

$$V_{ij}(\delta)=F(j-1)+G(j)+\delta d_{1,j-1}, \quad \text{if } pr(j)\le i. \tag{4}$$

Now suppose $pr(j)>i$. Let $k>i$ be the largest period on the shortest path from 1 to j such that $pr(k)\le i$. It follows immediately from (4) that

$$V_{ik}(\delta)=F(k-1)+G(k)+\delta_{1,k-1}.$$

Furthermore, by repeatedly applying Lemma 3.3(b) we deduce that $F(k-1)+G(k)\le F(j-1)+G(j)$, and from $k<j$ it follows that $d_{1,k-1}\le d_{1,j-1}$. Since $\delta\ge 0$ we obtain

$$V_{ik}(\delta)=F(k-1)+G(k)+\delta d_{1,k-1}\le F(j-1)+G(j)+\delta d_{1,j-1}.$$

Combining this result with inequality (3) yields $V_{ik}(\delta)\le V_{ij}(\delta)$. It follows that while determining $V_i(\delta)=\min_{j>i}\{V_{ij}(\delta)\}$ we can restrict ourselves to the periods j with $pr(j)\le i$. The desired result now follows from (4). □

From Lemma 4.6 it follows that we can obtain the optimal solution value as a function of $\delta\ge 0$, by constructing the lower envelope of the lines $F(j-1)+G(j)+\delta d_{1,j-1}$ for all $j>i$. Note that this lower envelope is piece-wise linear and concave. So in fact we are performing a complete parametric analysis; i.e., the optimal solution is found for al $\delta\ge 0$. However, we are only interested in largest value of δ for which the current optimal solution is still optimal. First suppose that this maximal allowable increase of h_i is strictly positive. Because the current optimal solution is clearly optimal for $\delta=0$, it follows that the line corresponding to that solution must be present in the lower envelope. To be more precise, it defines the line segment of the lower envelope for δ ranging from 0 to the first breakpoint of the lower envelope. Hence, this first breakpoint is exactly the maximal allowable increase of h_i. If the line corresponding to the current optimal solution is not present in the lower envelope, then it follows that this solution is not optimal for any positive value of δ. Hence, in that case the maximal allowable increase of h_i is 0. To obtain the maximal allowable increases for all h_i, $i=1,\ldots,n$, we construct the lower envelopes for decreasing i. Given the convex lower envelope for a fixed i, the lower envelope for $i-1$ is obtained after also taking the line $F(i-1)+G(i)+\delta d_{1,i-1}$ into consideration. This means that the lines are added in order of nonincreasing slope and it is not difficult to see that in this case $O(n)$ time is required to construct all lower envelopes (see for instance Wagelmans [19, Chapter 2]). Moreover, it follows that the maximal allowable increases of the parameters h_i are nondecreasing for increasing i.

We summarize our main result here.

Theorem 4.7. *The maximal allowable increases of all h_i, $i=1,\ldots,n$, can be calculated simultaneously in* $O(n)$ *time.*

Case 6: *h_i decreases to $h_i-\delta$.*

For this case we do not have a dominance relation similar to Lemma 4.6. Therefore we will describe a simple $O(n^2)$ algorithm that can also be used to perform a complete parametric analyis for $\delta \geq 0$. The functions of interest are

$$v_i(\delta) \equiv \min_{t \leq i < j} \{F(t-1) + l_{tj} + G(j) - \delta d_{1,j-1}\}, \quad i = 1, \ldots, n.$$

For $j > i$ we define $u_{ij} \equiv \min_{t \leq i}\{F(t-1) + l_{tj}\}$ and $v_{ij}(\delta) \equiv u_{ij} + G(j) - \delta d_{1,j-1}$. We first determine the values u_{ij} for all i, j with $1 \leq i < j \leq n+1$. For fixed j ($j = 2, \ldots, n$), we compute the values u_{ij}, $1 \leq i < j$, using $u_{1j} = l_{1j}$ and the simple recursion

$$u_{ij} := \min\{u_{i-1,j}, F(i-1) + l_{ij}\} \quad \text{for } 2 \leq i < j.$$

Hence, the determination of all values u_{ij}, $1 \leq i < j \leq n+1$, takes $O(n^2)$ time. Now consider a fixed i ($i = 1, \ldots, n$). Since now the lines $v_{ij}(\delta)$, $j > i$, are known and their slopes are already ordered, $v_i(\delta)$ can be determined as the lower envelope of these lines in $O(n)$ time. It follows that the construction of $v_i(\delta)$ for all $i = 1, \ldots, n$ can be done in $O(n^2)$ time and therefore the following holds.

Theorem 4.8. *The maximal allowable decreases of all h_i, $i = 1, \ldots, n$ can be calculated simultaneously in $O(n^2)$ time.*

4.3. Sensitivity analysis of the demands

It turns out that the analysis of changes in the demands resembles the earlier sensitivity analysis of the holding costs. Therefore, we will sometimes skip parts of proofs in this section.

If the demand d_i, $i = 1, \ldots, n$, changes by δ, then the cost of an arbitrary schedule changes as follows: let $1 = i_1 < i_2 < \cdots < i_r \leq i < i_{r+1} < \cdots < i_k < i_{k+1} = n+1$ denote the production periods of the schedule, then for this schedule the cost change equals δc_{i_r}. So the value of an optimal schedule is given by

$$W_i(\delta) \equiv \min_{j \leq i < t} \{F(j-1) + l_{jt} + G(t) + \delta c_j\}.$$

Case 7: d_i increases to $d_i + \delta$.

For $j \leq i$ we define $W_{ij}(\delta) \equiv \min_{t > i}\{F(j-1) + l_{jt} + G(t) + \delta c_j\}$. By definition $W_i(\delta) = \min_{j \leq i}\{W_{ij}(\delta)\}$. Moreover, we can show the following relation.

Lemma 4.9. $\min_{j \leq i}\{W_{ij}(\delta)\} = \min_{j \leq i}\{F(j-1) + G(j) + \delta c_j\}$ *for* $i = 1, \ldots, n$.

Proof. The idea of the proof is analogous to the proof of Lemma 4.6. Let $j \leq i$ and $k = sc(j)$. If $k > i$, then it is easy to show that $W_{ij}(\delta) = F(j-1) + G(j) + \delta c_j$. If $k \leq i$, then the following implies that we do not have to consider j.

$$\begin{aligned} W_{ij}(\delta) &= F(j-1) + \delta c_j + \min_{t > i}\{l_{jt} + G(t)\} \\ &\geq F(j-1) + \delta c_j + G(j) \quad \text{(definition of } G(j)) \end{aligned}$$

$$\geq F(k-1)+\delta c_j+G(k) \quad \text{(Lemma 3.3(a))}$$
$$\geq F(k-1)+\delta c_k+G(k) \quad \text{(Lemma 3.4).} \qquad \square$$

From Lemma 4.9 it follows that to determine the maximal allowable increase of the parameter d_i, it suffices to determine the first breakpoint of the lower envelope of the lines $F(j-1)+G(j)+\delta c_j$ for all $j \leq i$. Because there is in general no natural order of these lines with respect to their slopes or constant terms, our result here is the following.

Theorem 4.10. *The maximal allowable increases of all d_i, $i=1,\ldots,n$, can be calculated simultaneously in* O($n \log n$) *time.*

Case 8: *d_i decreases to $d_i-\delta$.*

Because no equivalent of Lemma 4.9 is known for this case, we will describe a simple O(n^2) algorithm to perform a complete parametric analysis for $0 \leq \delta \leq d_i$. The functions of interest are

$$w_i(\delta) \equiv \min_{j \leq i < t} \{F(j-1)+l_{jt}+G(t)-\delta c_j\}, \quad i=1,\ldots,n.$$

By definition $w_i(\delta)$ is the lower envelope of the lines $w_{ij}(\delta)$, $j \leq i$, defined by $w_{ij}(\delta) \equiv F(j-1)+f_j-\delta c_j+\min_{t>i}\{c_j d_{j,t-1}+G(t)\}$. Using a simple recursion we can again calculate the values $\min_{t>i}\{c_j d_{j,t-1}+G(t)\}$ for all $1 \leq j \leq i \leq n+1$ in O(n^2) time. Furthermore, it takes O($n \log n$) time to order the coefficients c_j, $j=1,\ldots,n$. Hence, we can compute all relevant lines $w_{ij}(\delta)$ and order them according to nonincreasing slope in O(n^2) time. Subsequently it takes O(n) time to construct $w_i(\delta)$ for a fixed period i.

The discussion above implies our last complexity result.

Theorem 4.11. *The maximal allowable decreases of all d_i, $i=1,\ldots,n$, can be calculated simultaneously in* O(n^2) *time.*

5. Concluding remarks

In Table 1 we have summarized the complexity of our algorithms. The running times refer to the computation of the allowable changes for all similar parameters. We have indicated when a single parameter can be treated separately, in which case the complexity should be divided by n.

From the table we see that our algorithms to compute maximal allowable increases and decreases have for most coefficients different complexities. The difference for the holding costs is especially striking. However, such asymmetrical phenomena are also encountered when performing sensitivity analysis of shortest path and minimum spanning trees (see for instance Spira and Pan [16]). On the

Table 1
Summary of complexity results

Parameter	Increase	Decrease
f_i	$n \log n$	n[a]
c_i, p_i	$n \log n$	$n \log n$[a]
h_i	n	n^2
d_i	$n \log n$	n^2

[a] The computations can be carried out for each period separately.

other hand, ELS has so much structure that more symmetrical results could be hoped for.

We have been able to show that the techniques described in Section 2 can be generalized to solve other lot-sizing problems. For instance the problem in which backlogging is allowed as well as the problem with start-up costs can be solved in $O(n \log n)$ time (see Van Hoesel [17]). It would be interesting to study sensitivity analysis of these more general problems.

Acknowledgement

We would like to thank Antoon Kolen and Alexander Rinnooy Kan for their comments on an earlier draft of this paper, and Auke Woerlee for his assistance in making the figures.

References

[1] A. Aggarwal and J.K. Park, Improved algorithms for economic lot-size problems, Working paper, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA (1990).
[2] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms (Addison-Wesley, London, 1974).
[3] S. Axsäter, Worst case performance of lot-sizing heuristics, European J. Oper. Res. 9 (1982) 339-343.
[4] K.R. Baker, Lot-sizing procedures and a standard data set: a reconciliation of the literature, J. Manufacturing Oper. Management 2 (1990) 199-221.
[5] G.R. Bitran, T.L. Magnanti and H.H. Yanasse, Approximation methods for the uncapacitated dynamic lot size problem, Management Sci. 30 (1984) 1121-1140.
[6] G.D. Eppen, F.J. Gould and B.P. Pashigian, Extensions of the planning horizon theorem in the dynamic lot size problem, Management Sci. 15 (1969) 268-277.
[7] J.R. Evans, An efficient implementation of the Wagner-Whitin algorithm for dynamic lot-sizing, J. Oper. Management 5 (1985) 229-235.

[8] 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 Sci. 37 (1991) 909–925.

[9] I.-S. Lee, On sensitivity analysis for shortest-path dynamic programming models, Report, University of California, Los Angeles, CA (1986).

[10] R.A. Lundin and T.E. Morton, Planning horizons for the dynamic lot size model: Zabel vs. protective procedures and computational results, Oper. Res. 23 (1975) 711–734.

[11] K. Richter, The two-criterial dynamic lot size problem, System Anal. Modelling Simulation (1986) 99–105.

[12] K. Richter, Stability of the constant cost dynamic lot size model, European J. Oper. Res. 31 (1987) 61–65.

[13] K. Richter and J. Vörös, A parametric analysis of the dynamic lot-sizing problem, J. Inform. Process. Cybernet. 25 (1989) 67–73.

[14] K. Richter and J. Vörös, On the stability region for the multi-level inventory problems, European J. Oper. Res. 41 (1989) 169–173.

[15] C. Saydam and M. McKnew, A fast microcomputer program for ordering using the Wagner–Whitin algorithm, Production Inventory Management J. 28 (1987) 15–19.

[16] P.M. Spira and A. Pan, On finding and updating spanning trees and shortest paths, SIAM J. Comput. 4 (1975) 375–380.

[17] C.P.M. Van Hoesel, Models and algorithms for single item lot-sizing problems, Ph.D. dissertation, Erasmus University, Rotterdam (1991).

[18] C.P.M. Van Hoesel and A.P.M. Wagelmans, A note on "Stability of the constant cost dynamic lot size model" by K. Richter, European J. Oper. Res. 55 (1991) 112–114.

[19] A.P.M. Wagelmans, Sensitivity analysis in combinatorial optimization, Ph.D. dissertation, Erasmus University, Rotterdam (1990).

[20] A.P.M. Wagelmans, C.P.M. Van Hoesel and A.W.J. Kolen, Economic lot-sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner–Whitin case, Oper. Res. 40 (1992) 145–156.

[21] H.M. Wagner, A postscript to dynamic problems in the theory of the firm, Naval Res. Logist. Quart. 7 (1960) 7–12.

[22] H.M. Wagner and T.M. Whitin, Dynamic version of the economic lot size model, Management Sci. 5 (1958) 89–96.

[23] E. Zabel, Some generalizations of an inventory planning horizon theorem, Management Sci. 10 (1964) 465–471.

[24] W.I. Zangwill, Minimum concave cost flows in certain networks, Management Sci. 14 (1968) 429–450.