Improved Algorithms for Economic Lot Size Problems

Alok Aggarwal; James K. Park

Operations Research, Vol. 41, No. 3. (May - Jun., 1993), pp. 549-571.

Stable URL:
http://links.jstor.org/sici?sici=0030-364X%28199305%2F06%2941%3A3%3C549%3AIAFELS%3E2.0.CO%3B2-Q

Operations Research is currently published by INFORMS.

IMPROVED ALGORITHMS FOR ECONOMIC LOT SIZE PROBLEMS

ALOK AGGARWAL

IBM T. J. Watson Research Center, Yorktown Heights, New York

JAMES K. PARK

Sandia National Laboratories, Sandia, New Mexico

(Received April 1990; revisions received February, June 1992; accepted June 1992)

Many problems in inventory control, production planning, and capacity planning can be formulated in terms of a simple economic lot size model proposed independently by A. S. Manne (1958) and by H. M. Wagner and T. M. Whitin (1958). The Manne-Wagner-Whitin model and its variants have been studied widely in the operations research and management science communities, and a large number of algorithms have been proposed for solving various problems expressed in terms of this model, most of which assume concave costs and rely on dynamic programming. In this paper, we show that for many of these concave cost economic lot size problems, the dynamic programming formulation of the problem gives rise to a special kind of array, called a Monge array. We then show how the structure of Monge arrays can be exploited to obtain significantly faster algorithms for these economic lot size problems. We focus on uncapacitated problems, i.e., problems without bounds on production, inventory, or backlogging; capacitated problems are considered in a separate paper.

This paper presents efficient algorithms for problems related to economic lot size models. Economic lot size models typically deal with production and/or inventory systems. A product (which could be a raw material, a purchased part, or a semifinished or finished product in manufacturing or retailing) is produced or purchased in batch quantities and placed in stock. As the stock is depleted by demands for the product, more of the product must be produced or purchased. The object of production planning is to minimize the cost of this cycle of filling and depleting the stock. Since there are usually a very large number of variables that affect production planning (which may include workforce levels, physical resources of the firm, and external variables such as federal regulations), economic lot size models typically make certain simplifying assumptions. Some researchers have studied models with the assumption that the demands on the inventory follow a given probabilistic distribution, while others have assumed that these demands are deterministic and known in advance. In this paper, we study models based on the latter assumption.

Harris (1915) is usually cited as the first to study economic lot size models that assume deterministic demands. He considered a model that assumes demands occur continuously over time. About three decades ago, a different approach was independently provided by Manne (1958) and by Wagner and Whitin (1958); they divided time into discrete *periods* and assumed that the demand in each period is known in advance. Since 1958, the Manne-Wagner-Whitin model has received considerable attention, and several hundred papers have directly or indirectly discussed this model; most of these papers have either extended this model or provided efficient algorithms for production problems that arise in it. (Indeed, Lee and Denardo (1986) have provided convincing reasons why the Manne-Wagner-Whitin model is a reasonable one.) The references given here and those given by Bahl, Ritzman and Gupta (1987) provide only some of the papers related to the Manne-Wagner-Whitin model. Today, even an introductory operations research textbook is likely to include a chapter on the Manne-Wagner-Whitin model and on some of its extensions. (See, for example, Johnson and Montgomery 1974, Wagner 1975, Denardo 1982, and Hax and Candea 1984.) Because of the immense interest in economic lot size models, a considerable amount of research effort has been focused on establishing the computational complexity of various economic lot size problems. (In particular, see Florian, Lenstra and Rinnooy Kan 1980, Bitran and Yanasse

Subject classifications: Analysis of algorithms: optimal and suboptimal algorithms. Computers/computer science: faster lot sizing using Monge arrays. Inventory/production: faster algorithms for various Wagner-Whitin models.
Area of review: OPTIMIZATION.

0030-364X/93/4103-0549 $01.25

1982, Luss 1982, Erickson, Monma and Veinott 1987, and Chung and Lin 1988.)

This paper reviews the Manne-Wagner-Whitin model and several of its extensions. It also provides efficient algorithms for a wide variety of concave cost production planning problems expressed in terms of this model. We focus on uncapacitated economic lot size problems, i.e., problems without bounds on production, inventory, or backlogging; capacitated problems, as well as related problems involving negative demands and shelf-life bounds, are considered in another paper (Aggarwal and Park 1992; see also Section 6). Our algorithms use dynamic programming (Bellman 1957) and array searching (Aggarwal et al. 1987, Wilber 1988, Aggarwal and Park 1989, Klawe 1989, Galil and Park 1990, Eppstein 1990, Larmore and Schieber 1991), and they typically improve the running times of previous algorithms by factors of n or $n/\lg n$, where n is the number of time periods under consideration; these improvements are listed in Tables I, II, and III. In many cases, the running times of our algorithms are optimal to within a constant factor or to within a factor of $\lg n$.

One of the critical contributions of this paper (and of Aggarwal and Park 1992) is our identification of the Monge arrays that arise in connection with the economic lot size model; it is these arrays that allow us to apply known array-searching techniques and improve the time bounds of previous algorithms for economic lot size problems so dramatically. We also raise several unresolved questions regarding the time complexities of various problems formulated in terms of the economic lot size model. It is our hope that these open questions will stimulate interest in the economic lot size model among researchers in theoretical computer science and related areas.

Recently, two groups of researchers from the operations research community—Federgruen and Tzur (1990, 1991) and Wagelmans, van Hoesel and Kolen (1992)—have independently obtained some of the results presented in this paper using very different techniques. We will briefly describe their work and contrast it with our own in the final section of this paper.

The remainder of this paper is organized as follows. In Section 1, we review the Manne-Wagner-Whitin model and several of its extensions, and we also list the main results of this paper. In Section 2, we discuss some of the techniques used in obtaining our results, and then in the Sections 3–5, we present our improved algorithms for three different types of economic lot size problems. Finally, in Section 6, we discuss several extensions to our work, relate our results to the aforementioned work of Federgruen and Tzur (1990, 1991) and of Wagelmans, van Hoesel and Kolen (1992), and present some open problems.

1. DEFINITIONS AND RESULTS

This section defines several variants of the economic lot size model and lists our algorithmic results for these models. Subsection 1.1 focuses on the basic economic lot size model, in which no backlogging of demand is allowed, while subsection 1.2 examines an extension to the basic model that does allow backlogging, and subsection 1.3 considers two infinite-planning horizon variants of the backlogging model.

1.1. The Basic Model

To describe the basic economic lot sizing model proposed by Manne (1958) and Wagner and Whitin (1958), we use the notation employed by Denardo. Demand for the product in question occurs during each of n consecutive time *periods* (i.e., intervals of time) numbered 1 through n. The demand that occurs during a given period can be satisfied by production during that period or during any earlier period, as inventory is carried forward in time. (This basic model differs from the backlogging model described in subsection 1.2 in that demand is not allowed to accumulate and be satisfied by future production.) Without loss of generality, we assume both the initial inventory (at the beginning of the first period) and the final inventory (at the end of period n) are zero. The model includes production costs and inventory costs, and the objective is to schedule production to satisfy demand at minimum total cost.

The data in this model are the demands, the production cost functions, and the inventory cost functions. In particular, for $1 \leqslant i \leqslant n$,

d_i = the demand during period i;
$c_i(x)$ = the cost of producing x units of inventory during period i; and
$h_i(y)$ = the cost of storing y units of inventory from period $i-1$ to period i;

where we assume $d_i \geqslant 0$ for all i. Furthermore, the model has $2n+1$ variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_{n+1}$, where for $1 \leqslant i \leqslant n$,

x_i = the production during period i,

and for $1 \leqslant i \leqslant n+1$,

y_i = the inventory stored from period $i-1$ to period i.

Demand, production, and inventory occur in real quantities, and the problem of meeting demand at minimal total cost has the following mathematical representation:

$$\text{minimize} \sum_{i=1}^{n} \{c_i(x_i) + h_i(y_i)\}$$

subject to the constraints

$$y_1 = y_{n+1} = 0$$

$$x_i \geqslant 0 \qquad \text{for } 1 \leqslant i \leqslant n,$$

$$y_i \geqslant 0 \qquad \text{for } 1 \leqslant i \leqslant n, \text{ and}$$

$$y_i + x_i = d_i + y_{i+1} \quad \text{for } 1 \leqslant i \leqslant n.$$

In the above, the first constraint assures that the initial and final inventories are zero, while the second and third constraints limit production and inventory to nonnegative values. (Requiring inventory to be nonnegative ensures that the demand in period i is satisfied by production during that period or during earlier periods.) Finally, matter must be conserved, so the fourth constraint requires that the sum of the inventory at the start of a period and the production during that period equals the sum of the demand during that period and the inventory at the start of the next period.

The production levels $x_1, \ldots, x_n$ completely describe a *production plan* or *production schedule*, as for $1 \leqslant i \leqslant n + 1$, we must have

$$y_i = (d_1 + \ldots + d_{i-1}) - (x_1 + \ldots + x_{i-1}). \qquad (2)$$

We will say that a particular schedule is *feasible* if its production levels and the inventory levels determined by (2) satisfy the constraints of (1). Moreover, we will say that a particular schedule is *optimal* if it is a feasible production schedule that minimizes $\sum_{i=1}^{n} \{c_i(x_i) + h_i(y_i)\}$ over all feasible production schedules.

The basic economic lot size problem can also be formulated as a network flow problem. (This formulation was first proposed by Zangwill 1968.) Consider the directed graph depicted in Figure 1. This graph consists of a single source, capable of generating a net outflow of $\sum_{i=1}^{n} d_i$, and n sinks, such that the ith sink requires d_i units of net inflow. Furthermore, for $1 \leqslant i \leqslant n$, there is an arc from the source to ith sink with associated cost function $c_i(\cdot)$, and for $2 \leqslant i \leqslant n$, there is an arc from the $(i - 1)$st sink to the ith sink with associated cost function $h_i(\cdot)$. A minimum cost flow for this graph corresponds to an optimal production schedule for the associated economic lot size problem.

If the $c_i(\cdot)$ and $h_i(\cdot)$ are allowed to be arbitrary functions, then the basic economic lot size problem is

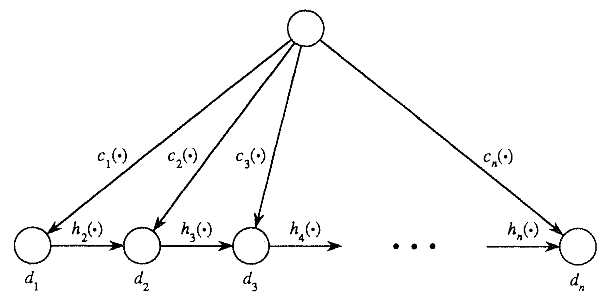

Figure 1. The basic economic lot size problem can be formulated as a network flow problem.

presumably quite difficult to solve, as Florian, Lenstra and Rinnooy Kan have shown it is (weakly) NP-hard. In view of this difficulty, certain assumptions are often made about the lot size problem's cost functions; we review some of these assumptions below.

1. In their pioneering papers, Manne (1958) and Wagner and Whitin (1958) assumed that for $1 \leqslant i \leqslant n$,

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

and $h_i(y) = h_i^1 y$, where c^1 and the c_i^0 and h_i^1 are all nonnegative constants. (The assumption that $c^1 \geqslant 0$ can be dropped, as changing c^1 affects only the cost of the optimal production schedule and not its structure.) Wagner and Whitin also provided an $O(n^2)$-time algorithm for computing an optimal production plan. Note that the *setup costs* c_i^0 are what make this problem interesting; if $c_i^0 = 0$ for all i, then the problem can be solved trivially.

2. Wagner (1960) showed that the $O(n^2)$-time algorithm of Wagner and Whitin can still be used if for $1 \leqslant i \leqslant n$, the function $c_i(x)$ is an arbitrary concave function (or more precisely, it is concave on $[0, +\infty)$, the relevant portion of its domain) and $h_i(y) = h_i^1 y$, where the h_i^1 are again nonnegative constants.

3. Zabel (1964) and Eppen, Gould and Pashigian (1969) considered a somewhat simpler cost structure; for $1 \leqslant i \leqslant n$, they assumed that

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

and $h_i(y) = h_i^1 y$, where the c_i^0, c_i^1, and h_i^1 are all nonnegative constants. (The assumption that $c_i^1 \geqslant 0$ for $1 \leqslant i \leqslant n$ can be dropped, as changing all the c_i^1 by the same amount affects only the cost of the optimal production schedule and not its structure.)

For this cost structure, both Zabel and Eppen, Gould and Pashigian provided some additional properties of an optimal production schedule. Both papers also exploited these properties to obtain algorithms for computing an optimal schedule that run faster in practice but which still require quadratic time in the worst case.

4. Zangwill (1969) again assumed that

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

for $1 \leqslant i \leqslant n$, but he allowed the $h_i(\cdot)$ to be arbitrary concave functions. For this cost structure, he showed that Wagner and Whitin's approach still yields an $O(n^2)$-time algorithm for computing an optimal production schedule (see also subsection 1.2).

5. Finally, Veinott (1963) showed that even if both the $c_i(\cdot)$ and the $h_i(\cdot)$ are arbitrary concave functions, Wagner and Whitin's approach gives an $O(n^2)$-time algorithm.

Observe that if we interpret $f(x)$ as the cost of producing (or storing) x items, then a concave $f(\cdot)$ implies *decreasing marginal costs*, or equivalently, *economies of scale*. Microeconomic theories often assume economies of scale, which is one of the reasons why the economic lot size model with linear or concave costs has received so much attention.

In Section 3, we provide efficient algorithms for several of the cost structures discussed above. The time complexities of these algorithms are listed in Table I. The new algorithms use dynamic programming, as well as some recently developed techniques for searching in what are known as Monge arrays. (For definitions of these arrays and for searching techniques related to them, see subsection 2.2.)

1.2. The Backlogging Model

Until now, we have assumed that the demand for a particular period is satisfied by production during that period or during earlier periods. Zangwill (1966) extended the basic model by relaxing this assumption and allowing demand to go unsatisfied during some period, provided it is satisfied eventually by production in some subsequent period. (Satisfying demand with future production is known as *backlogging demand*.) Zangwill's extension changes the formulation of the economic lot size problem given in subsection 1.1 in that it allows the variables y_2 through y_n in (1) to be negative. Equation 2 still identifies y_i as the total production during periods 1 through $i - 1$ less the total demand during those periods; however, when y_i is negative, it now represents a *shortage* of $-y_i$ units of unfulfilled (backlogged) demand that must be satisfied during periods i through n. Furthermore, when y_i is nonnegative, $h_i(y_i)$ remains equal to the cost of y_i units of inventory at the start of period i, but when y_i is negative, $h_i(y_i)$ becomes the cost of having a shortage of y_i units at the start of period i. To simplify our notation, we define backlogging cost functions $g_i(\cdot)$ such that $g_{i-1}(-y_i) = h_i(y_i)$ for $2 \leqslant i \leqslant n$.

The backlogging economic lot size problem, like the basic problem, can also be formulated as a network flow problem. We use the same single-source, n-sink directed graph as for the basic economic lot size problem, except that for $2 \leqslant i \leqslant n$, we add an arc from the ith sink to the $(i - 1)$st sink with associated cost function $g_{i-1}(\cdot)$. This new graph is depicted

Table 1
A Summary of Our Results for the Basic Economic Lot Size Problem

Cost Structure	Previous Results	Results of This Paper[a]
$c_i(0) = 0$ $c_i(x) = c_i^0 + c_i^1 x$ for $x > 0$ $c_i^0 \geqslant 0$ $h_i(y) = h_i^1 y$ $c_i^1 \leqslant c_{i-1}^1 + h_i^1$	$O(n^2)$ (Wagner and Whitin assumed $c_i^1 = c^1$)	$O(n)$ (Theorem 2)
$c_i(0) = 0$ $c_i(x) = c_i^0 + c_i^1 x$ for $x > 0$ $c_i^0 \geqslant 0$ $h_i(y) = h_i^1 y$	$O(n^2)$ (Zabel) (Eppen, Gould and Pashigian)	$O(n \lg n)$ (Theorem 3)
$c_i(\cdot)$ and $h_i(\cdot)$ concave	$O(n^2)$ (Veinott)	No improvement

[a] The results are bounds on the time to find an optimal production schedule, where n is the number of periods.

in Figure 2. Again, a minimum cost flow for this graph corresponds to an optimal production schedule for the associated economic lot size problem with backlogging.

As is the case for the basic economic lot size problem (given in subsection 1.1), the backlogging economic lot size problem is (weakly) NP-hard if arbitrary cost functions are allowed. For this reason, researchers have studied several different restricted cost structures, some of which are listed below.

1. Zangwill (1966) assumed that the $c_i(\cdot)$, $h_i(\cdot)$, and $g_i(\cdot)$ are all arbitrary concave functions and provided an $O(n^3)$-time dynamic programming algorithm for computing an optimal production plan.

2. Zangwill (1969) assumed that

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

for $1 \leqslant i \leqslant n$ (where c^1 and the c_i^0 are nonnegative constants) and that the $h_i(\cdot)$ and $g_i(\cdot)$ are arbitrary concave functions. For this cost structure, he provided an $O(n^2)$-time algorithm for computing an optimal production plan.

3. Blackburn and Kunreuther (1974) and Lundin and Morton (1975) assumed that

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

$g_i(z) = g_i^1 z$, and $h_i(y) = h_i^1 y$, where the c_i^0, c_i^1, g_i^1, and h_i^1 are all nonnegative. For this case, they obtained some characteristics of optimal production schedules; these characteristics are generalizations of those given by Eppen, Gould and Pashigian for the basic model (i.e., the one without backlogging). Both papers also gave algorithms for determining an optimal production plan, but these algorithms again take quadratic time in the worst case (though Lundin and Morton argued that their algorithm does run faster in practice than Zangwill's algorithm).

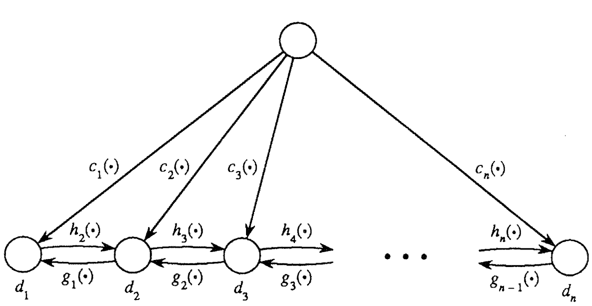

Figure 2. The backlogging economic lot size problem can be formulated as a network flow problem.

4. Finally, Morton (1978) considered a very simple cost structure in which

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

$g_i(z) = g^1 z$, and $h_i(y) = h^1 y$, where c^1, g^1, h^1, and the c_i^0 are again nonnegative. For this case, Morton provided a very simple $O(n^2)$-time algorithm, which seems to run quite efficiently in practice.

In Section 4, we provide asymptotically faster algorithms for most of the cost structures discussed above. The time complexities of these algorithms are listed in Table II. We again use both paradigms, viz., dynamic programming and searching in Monge arrays.

1.3. Two Periodic Models

Since market demands often display periodic behavior (which may arise, for example, because of the inherent cyclicity in seasonal demands), Graves and Orlin (1985) and Erickson, Monma and Veinott (1987) have studied two different variants of the backlogging economic lot size problem that assume the planning horizon is infinite, i.e., we are planning for an infinite number of periods, but the costs and demands are periodic with periodicity n.

Erickson, Monma and Veinott considered the problem of finding an infinite production schedule with minimum average cost per period, subject to the constraint that the production schedule also have periodicity n. Equivalently, they want a minimum cost, n-period production schedule for periods i through $i + n - 1$, where i is allowed to vary between 1 and n. Their model can be interpreted in a graph-theoretic sense as the backlogging flow network (given in Figure 2) with two additional arcs—one corresponding to inventory and the other corresponding to backlogging—between the first sink and the nth sink. For this problem, Erickson, Monma and Veinott obtained an $O(n^3)$-time algorithm.

The second periodic variant of the backlogging problem, considered by Graves and Orlin, is also concerned with finding an infinite production schedule with minimum average cost per period. However, the schedule is not restricted to have periodicity n; instead, an assumption is made about the limiting behavior of the $g_i(\cdot)$ and $h_i(\cdot)$ (see Section 5 for more details). For this problem, Graves and Orlin gave an $O(p^3 n^3)$-time algorithm, where p is a parameter that

Table II
A Summary of Our Results for the Economic Lot Size Problem With Backlogging

Cost Structure	Previous Results	Results of This Paper[a]
$c_i(0) = 0$ $c_i(x) = c_i^0 + c_i^1 x$ for $x > 0$ $c_i^0 \geq 0$ $g_i(z) = g_i^1 z$ $h_i(y) = h_i^1 y$ $c_i^1 \leq c_{i+1}^1 + g_i^1$ $c_i^1 \leq c_{i-1}^1 + h_i^1$	$O(n^2)$ (Morton assumed $c_i^1 = c^1$; $g_i^1 = g^1$; $h_i^1 = h^1$)	$O(n)$ (Theorem 4)
$c_i(0) = 0$ $c_i(x) = c_i^0 + c_i^1 x$ for $x > 0$ $c_i^0 \geq 0$ $g_i(z) = g_i^1 z$ $h_i(y) = h_i^1 y$	$O(n^2)$ (Blackburn and Kunreuther and Lundin and Morton)	$O(n \lg n)$ (Theorem 5)
$c_i(0) = 0$ $c_i(x) = c_i^0 + c^1 x$ for $x > 0$ $c_i^0 \geq 0$ $h_i(\cdot)$ and $g_i(\cdot)$ concave	$O(n^2)$ (Zangwill 1969)	No improvement
$c_i(\cdot)$, $h_i(\cdot)$, and $g_i(\cdot)$ concave	$O(n^3)$ (Zangwill 1966)	$O(n^2)$ Theorem 6

[a] The results are bounds on the time to find an optimal production schedule, where n is the number of periods.

depends upon production, inventory, and backlogging costs.

In Section 5, we give efficient algorithms for both Erickson, Monma and Veinott's problem and Graves and Orlin's problem. The time complexities of these algorithms are given in Table III.

2. MAIN TECHNIQUES USED

In this section, we sketch some of the techniques used in obtaining our results. Subsection 2.1 gives the basic dynamic programming framework developed by previous researchers for solving economic lot size problems. Subsection 2.2 presents the array-searching techniques that we combine with the techniques of subsection 2.1 to obtain our improved results.

2.1. Arborescent Flows and Dynamic Programming

As we mentioned in subsections 1.1 and 1.2, both the basic and backlogging variants of the economic lot size problem can be formulated as network flow problems. Moreover, if the cost functions $c_i(\cdot)$, $g_i(\cdot)$, and $h_i(\cdot)$ assigned to these networks' edges are all concave,

Table III
A Summary of Our Results for the Two Periodic Economic Lot Size Problems

Problem and Cost Structure	Previous Results	Results of This Paper
Erickson, Monma and Veinott's problem $c_i(0) = 0$ $c_i(x) = c_i^0 + c^1 x$ for $x > 0$ $c_i^0 \geq 0$ $h_i(\cdot)$ and $g_i(\cdot)$ concave and nondecreasing	None	$O(n^2)$ (Theorem 7)
Erickson, Monma and Veinott's problem $c_i(\cdot)$, $h_i(\cdot)$, and $g_i(\cdot)$ concave	$O(n^3)$ (Erickson, Monma and Veinott)	No improvement
Graves and Orlin's problem $c_i(\cdot)$, $h_i(\cdot)$, and $g_i(\cdot)$ concave	$O(p^3 n^3)$ (Graves and Orlin)	$O(p^2 n^3)$ (Theorem 8)

[a] The results are bounds on the time to find an optimal production schedule, where n is the periodicity and p is a function of the $c_i(\cdot)$, $g_i(\cdot)$, and $h_i(\cdot)$.

then we need only consider flows of a certain type in finding a minimum cost flow. Specifically a flow in an uncapacitated directed graph G is called *arborescent* if the directed edges of G carrying nonzero flow, when viewed as undirected edges, form an undirected acyclic graph on the vertices of G. As the following theorem shows, we can restrict our attention to arborescent flows in network flow problems with concave edge cost functions.

Theorem 1. (folklore; see Zangwill 1968 and Erickson, Monma and Veinott 1987) *Consider the flow problem associated with a directed graph G, where each arc e of G is assigned a cost function $c_e(\cdot)$ and the only constraint on the flow f_e on arc e is $f_e \geqslant 0$. If $c_e(\cdot)$ is concave for all arcs e, then some minimum cost flow in G is arborescent.*

This theorem appears (in one form or another) in all the papers dealing with the economic lot size problem that we consider. It is important because it implies that we need only consider production schedules that supply the demand for period i from at most one of the following sources: production during period i, inventory from period $i-1$, or, in the case of the backlogging model, demand backlogged to period $i+1$. Consequently, the basic and backlogging economic lot size problems have dynamic programming formulations. Specifically, let $E(1) = 0$, and for $1 < j \leqslant n+1$, let $E(j)$ denote the minimum cost of supplying the demands of periods 1 through $j-1$ such that the inventory y_j carried forward to (or backlogged from) period j is zero. This definition implies that $E(n+1)$ is the cost of the desired optimal production schedule for periods 1 through n. Moreover, as suggested in Figures 3 and 4, if P_j is an optimal production schedule achieving $E(j)$, then there exists an i in the range $1 \leqslant i < j$ such that P_j can be decomposed into a single period of production satisfying the demands of periods i through $j-1$, and an optimal production schedule achieving $E(i)$. Thus, if we let $d_{ij} = \sum_{m=i}^{j-1} d_m$ for $1 \leqslant i < j \leqslant n+1$, then for the basic problem,

$$E(j) = \min_{1 \leqslant i < j} \left\{ E(i) + c_i(d_{i,j}) + \sum_{m=i+1}^{j-1} h_m(d_{m,j}) \right\},$$

and for the backlogging problem,

$$E(j) = \min_{1 \leqslant i \leqslant k < j} \left\{ E(i) + c_k(d_{i,j}) + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}) \right\},$$

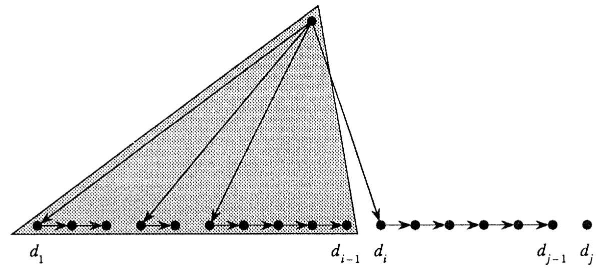

Figure 3. Consider any instance of the basic economic lot size problem, and suppose that P_j is a minimum cost arborescent production schedule satisfying the demands of periods 1 through $j-1$ such that no inventory is carried forward to period j. Furthermore, suppose that P_j's last production occurs during period i. Since P_j is arborescent, the demands of periods i through $j-1$ must all be satisfied by the production during period i. Moreover, the subschedule of P_j corresponding to periods 1 through $i-1$ (indicated by the shaded region) must be a minimum cost arborescent production schedule satisfying the demands of periods 1 through $i-1$ such that no inventory is carried forward to period i.

provided we view summations of the form $\sum_{m=i}^{j}(\ldots)$ as evaluating to 0 if $i > j$.

Note that these dynamic programming formulations for the basic and backlogging economic lot size problems give $O(n^2)$-time and $O(n^3)$-time algorithms, respectively, for computing the cost of an optimal production schedule; we merely evaluate $E(1)$, $E(2)$, $\ldots$, $E(n+1)$ in the naive fashion. Furthermore, we can extract an optimal production schedule (not just its cost) in $O(n)$ additional time, provided for each $E(j)$ we remember the i such that

$$E(j) = E(i) + c_i(d_{i,j}) + \sum_{m=i+1}^{j-1} h_m(d_{m,j})$$

or the i and k such that

$$E(j) = E(i) + c_k(d_{i,j}) + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}).$$

2.2. Searching in Monge Arrays

In this subsection, we first define the notion of a Monge array and then relate Monge arrays to dynamic

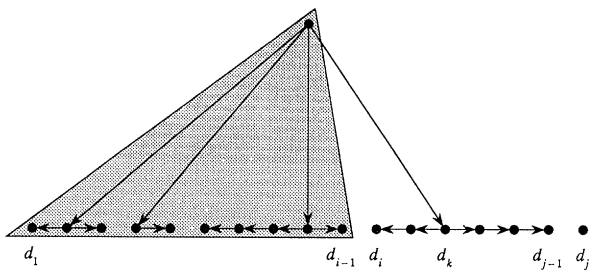

Figure 4. Consider any instance of the backlogging economic lot size problem, and suppose that P_j is a minimum-cost arborescent production schedule satisfying the demands of periods 1 through $j-1$ such that no inventory is carried forward to or backlogged from period j. Furthermore, suppose that P_j's last production occurs during period k and that i is first period whose demand is satisfied by this production. Since P_j is arborescent, the demands of periods i through $j-1$ must all be satisfied by the production during period k. Moreover, the subschedule of P_j corresponding to periods 1 through $i-1$ (indicated by the shaded region) must be a minimum cost arborescent production schedule satisfying the demands of periods 1 through $i-1$ such that no inventory is carried forward to or backlogged from period i.

programming. An $m \times n$ two-dimensional array $A = \{a[i, j]\}$ is said to satisfy the *Monge condition* if for $1 \leq i < m$ and $1 \leq j < n$,

$$a[i, j] + a[i+1, j+1] \leq a[i, j+1] + a[i+1, j].$$

Similarly, A is said to satisfy the *inverse Monge condition* if for $1 \leq i < m$ and $1 \leq j < n$,

$$a[i, j] + a[i+1, j+1] \geq a[i, j+1] + a[i+1, j].$$

For $d \geq 3$, a d-dimensional array A is said to satisfy the Monge condition if every two-dimensional plane of A (corresponding to fixed values of all but two of A's d coordinates) satisfies the Monge condition. Similarly, A is said to satisfy the inverse Monge condition if every two-dimensional plane of A satisfies the inverse Monge condition. Finally, an array satisfying the Monge condition is called *Monge*, and an array satisfying the inverse Monge condition is called *inverse Monge*.

Note that an $m \times n$ array $A = \{a[i, j]\}$ is Monge if and only if for $1 \leq i < k \leq m$ and $1 \leq j < l \leq n$,

$$a[i, j] + a[k, l] \leq a[i, l] + a[k, j],$$

as it is not hard to show that this latter property follows from the Monge condition. In fact, this property is often the one used in the literature to define two-dimensional Monge arrays.

Two-dimensional Monge arrays were first considered by Hoffman (1963); he showed (among other things) that a fast greedy algorithm solves the classical transportation problem if and only if the problem's cost array is Monge. Higher-dimensional Monge arrays were introduced by Aggarwal and Park (1989).

Monge arrays are closely related to submodular functions: We can view a $2 \times 2 \times \ldots \times 2$ d-dimensional Monge array $A = \{a[i_1, i_2, \ldots, i_d]\}$ as a submodular function $f(\cdot)$ on subsets of $\{1, 2, \ldots, d\}$ if we let $f(S) = a[i_1, i_2, \ldots, i_d]$, where for $1 \leq k \leq d$, $i_k = 1$ if $k \notin S$ and $i_k = 2$ if $k \in S$. (See Lovász 1983 for an overview of the theory of sub- and supermodular functions.)

One very important property of two-dimensional Monge and inverse-Monge arrays is that we can find maximal or minimal entries in such arrays quite efficiently. In particular, we can find a maximum (or minimum) entry in each row (or column) of an $n \times n$ Monge (or inverse-Monge) array A in $O(n)$ time, provided any particular entry of A can be looked up (or computed) in constant time. This result is due to Aggarwal et al., who considered a slight variant of this array-searching problem in the context of several problems from computational geometry and VLSI river routing. We call this array-searching problem an *off-line* problem, because all of the entries of A are available all the time.

In developing a linear time algorithm for the concave least-weight subsequence problem, Wilber (1988) extended the algorithm of Aggarwal et al. to a dynamic programming setting. Specifically, he gave an algorithm for the following *on-line* variant of the Monge-array, column-minima problem. Let $W = \{w[i, j]\}$ denote an $n \times (n+1)$ Monge array, where any entry of W can be computed in constant time. Furthermore, let $A = \{a[i, j]\}$ denote the $n \times (n+1)$ array defined by

$$a[i, j] = \begin{cases} E(i) + w[i, j] & \text{if } i < j, \\ +\infty & \text{if } i \geq j, \end{cases}$$

where $E(1) = 0$ and for $1 < j \leq n+1$,

$$E(j) = \min_{i<j} a[i, j].$$

Using the Mongité of A, which follows from its definition, Wilber showed that the column minima of A (and, hence, $E(2), \ldots, E(n+1)$) can be computed in

$O(n)$ time. This is called an *on-line* problem because certain inputs to the problem (i.e., entries of A) are available only after certain outputs of the problem (i.e., column minima of A) have been calculated.

In a subsequent paper dealing with the modified string-editing problem, Eppstein generalized Wilber's result. He showed that the column minima of A can be computed in $O(n)$ time even if we relax the assumption that $E(j)$ is the jth column minimum of A and assume only that it can be computed in constant time once both $E(j-1)$ *and* the jth column minimum of A are known. Note that this last assumption implies Eppstein's algorithm must first compute the second column minimum of A, then $E(2)$, then the third column minimum of A, then $E(3)$, and so on. This property is significant because it allows the computation of $E(2), \ldots, E(n+1)$ to be interleaved with the computation of some other sequence $F(2), \ldots, F(n+1)$ (corresponding to the column minima of another Monge array) such that $E(j)$ depends on $F(j-1)$ and $F(j)$ depends on $E(j)$ (see subsection 4.1.1 for more details).

Finally, though Eppstein's algorithm is sufficient for the purposes of this paper, we note that Klawe (1989), Galil and Park (1990), and Larmore and Schieber (1991) independently extended Eppstein's result a step further. They showed that the array A need not have the form $a[i, j] = E(i) + w[i, j]$ for $i < j$; so long as $a[i, j]$ can be computed in constant time once the first through ith column minima of A are known, the column minima of A can still be computed in $O(n)$ time.

To obtain our improved algorithms for the various economic lot size problems discussed in Sections 3–5, we use the off-line algorithm of Aggarwal et al. and the on-line algorithm of Eppstein. We also use a simple $O(n \lg n)$-time divide-and-conquer algorithm for the on-line array-searching problem when W does not satisfy the Monge condition, but some permutation of its rows and/or columns makes suitably chosen subarrays of W satisfy the Monge condition (see Lemma 2 below).

An important technical contribution of this paper is our identification of the Monge arrays that arise in the context of various economic lot size problems. The following two lemmas characterize the two most important (at least for the purposes of this paper) origins of these arrays. The proofs of these lemmas are easy and are therefore omitted.

Lemma 1. *Let $A = \{a[i,j]\}$ denote an $m \times n$ Monge array, let $B = \{b[i]\}$ denote an m-vector, and let $C = \{c[j]\}$ denote an n-vector. Furthermore, let $A' = \{a'[i, j]\}$ denote the $m \times n$ array where $a'[i, j] = a[i, j] + b[i] + c[j]$. A' is Monge.*

Lemma 2. *Let $f(\cdot)$ denote a concave function, let $B = \{b[i]\}$ denote an m-vector such that $b[1] \leq b[2] \leq \ldots \leq b[m]$, and let $C = \{c[j]\}$ denote an n-vector such that $c[1] \leq c[2] \leq \ldots \leq c[n]$. Furthermore, let $A_{f,B,C} = \{a_{f,B,C}[i, j]\}$ denote the $m \times n$ array, where $a_{f,B,C}[i, j] = f(b[i] + c[j])$, and for any vector $X = \{x[i]\}$, let $-X$ denote the vector obtained by negating all the entries of X. Then $A_{f,B,C}$, $A_{f,-B,-C}$, $A_{-f,-B,C}$ and $A_{-f,B,-C}$ are Monge, and $A_{f,-B,C}$, $A_{f,B,-C}$, $A_{-f,B,C}$ and $A_{-f,-B,-C}$ are inverse Monge.*

As an example of why these rather simple observations are useful, consider the following corollary to Lemmas 1 and 2.

Corollary 1. *Let $B = \{b[i]\}$ and $D = \{d[i]\}$ denote arbitrary m-vectors, and let $C = \{c[j]\}$ and $E = \{e[j]\}$ denote arbitrary n-vectors. Furthermore, let $A = \{a[i, j]\}$ denote the $m \times n$ array, where $a[i, j] = b[i]c[j] + d[i] + e[j]$. If $b[1] \leq b[2] \leq \ldots \leq b[m]$ and $c[1] \geq c[2] \geq \ldots \geq c[n]$, then A is Monge.*

Proof. Let $A' = \{a'[i, j]\}$ denote the $m \times n$ array where $a'[i, j] = b[i]c[j]$, and consider the concave function $f(x) = -2^x$, the m-vector $B' = \{b'[i]\}$ where $b'[i] = \lg b[i]$, and the n-vector $C' = \{c'[j]\}$ where $c'[j] = -\lg(c[j])$. Clearly, $a'[i, j] = -f(b'[i] - c'[j])$, and the entries of B' and C' are both in increasing order; thus, by Lemma 2, A' is Monge. Furthermore, since $a[i, j] = a'[i, j] + d[i] + e[j]$, A is also Monge by Lemma 1.

Note that even if the entries of B and C in the above corollary are not sorted, we can still make the array A Monge by permuting its rows and columns. Specifically, if we find permutations β and γ such that $b[\beta(1)] \leq b[\beta(2)] \leq \ldots \leq b[\beta(m)]$ and $c[\gamma(1)] \geq c[\gamma(2)] \geq \ldots \geq c[\gamma(n)]$, then the array $A'' = \{a''[i, j]\}$ where,

$$a''[i, j] = b[\beta(i)]c[\gamma(j)] + d[\beta(i)] + e[\gamma(j)]$$
$$= a[\beta(i), \gamma(j)]$$

is Monge. We make use of this observation repeatedly.

3. THE BASIC PROBLEM

This section investigates the time complexity of the basic economic lot size problem under several different assumptions about the production and inventory cost functions. In subsection 3.1, we consider nearly linear production costs and linear inventory costs, while in subsection 3.2, we discuss other concave production and inventory cost functions.

3.1. Nearly Linear Costs

In this subsection, we give results for instances of the basic economic lot size problem with what we will call *nearly linear* costs. Specifically, for $1 \leqslant i \leqslant n$, we assume

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

and $h_i(y) = h_i^1 y$, where c_i^0, c_i^1, and h_i^1 are constants and $c_i^0 \geqslant 0$. (The restriction on c_i^0 is necessary to ensure that $c_i(x)$ is concave, so that the techniques of subsection 2.1 can be applied.) In the operations research literature, this cost structure is often described as consisting of *fixed-plus-linear* production costs and linear inventory costs.

We begin with a special case in subsection 3.1.1: For $1 < i \leqslant n$, we assume that $c_i^1 \leqslant c_{i-1}^1 + h_i^1$. For this special case of the basic lot size problem, we give a linear time algorithm for computing the optimal production schedule. Then, in subsection 3.1.2, we remove this constraint on the coefficients of the cost functions, at the expense of an increase in our algorithm's running time by factor of lg n.

3.1.1. Restricted Coefficients

In this subsection, we consider a nearly linear cost structure where the cost efficients satisfy $c_i^1 \leqslant c_{i-1}^1 + h_i^1$ for $1 < i \leqslant n$. In other words, we assume that the marginal cost of producing during period i is at most the marginal cost of producing during period $i - 1$ plus the marginal cost of storing inventory from period $i - 1$ to period i. This particular cost structure subsumes those considered by Manne (1958) and Wagner and Whitin (1958). The latter paper gave an $O(n^2)$-time algorithm for computing an optimal production schedule; we improve this time bound to $O(n)$ for our slightly more general cost structure.

Recall the dynamic programming formulation of the basic economic lot size problem given in subsection 2.1: If we let $E(j)$ denote the minimum cost of satisfying the demands of periods 1 through $j - 1$ such that the inventory y_j carried forward from period $j - 1$ to period j is 0, then $E(1) = 0$ and for $2 \leqslant j \leqslant n + 1$,

$$E(j) = \min_{1 \leqslant i < j} \left\{ E(i) + c_i(d_{i,j}) + \sum_{m=i+1}^{j-1} h_m(d_{m,j}) \right\},$$

where $d_{i,j} = d_i + d_{i+1} + \ldots + d_{j-1}$. Solving this recurrence in the naive fashion gives the $O(n^2)$-time algorithm of Wagner and Whitin.

To compute $E(2), \ldots, E(n + 1)$ in linear time, we consider the $n \times (n + 1)$ array $A = \{a[i, j]\}$ where

$$a[i,j] = \begin{cases} E(i) + c_i^0 + c_i^1 d_{i,j} + \sum_{m=i+1}^{j-1} h_m^1 d_{m,j} & \text{if } i < j, \\ +\infty & \text{if } i \geqslant j. \end{cases}$$

(One is tempted to use instead the $n \times (n + 1)$ array $B = \{b[i, j]\}$ where

$$b[i, j] = \begin{cases} E(i) + c_i(d_{i,j}) + \sum_{m=i+1}^{j-1} h_m^1 d_{m,j} & \text{if } i < j, \\ +\infty & \text{if } i \geqslant j, \end{cases}$$

but this array may not be Monge.) Now if $d_{j-1} = 0$, then $E(j) = E(j - 1)$. On the other hand, if $d_{j-1} > 0$, then $d_{m,j} > 0$ for all $m < j$, which implies

$$E(i) + c_i(d_{i,j}) + \sum_{m=i+1}^{j-1} h_m(d_{m,j}) = a[i, j]$$

for all $i < j$ and

$$E(j) = \min_{1 \leqslant i \leqslant n} a[i, j].$$

Combining these two observations gives the following recurrence for $E(j)$ when $2 \leqslant j \leqslant n + 1$:

$$E(j) = \begin{cases} E(j - 1) & \text{if } d_{j-1} = 0, \\ \min_{i \leqslant i \leqslant n} a[i, j] & \text{if } d_{j-1} > 0. \end{cases}$$

At this point, we would like to apply one of the online array-searching algorithms mentioned in subsection 2.2 to compute the column minima of A (and, hence, $E(2), \ldots, E(n + 1)$). Since $E(i)$ can be computed in constant time from $E(i - 1)$ and the ith column minimum of A, Eppstein's algorithm will suffice, provided we can prove the following two lemmas.

Lemma 3. *A is Monge.*

Proof. For $1 \leqslant i < j \leqslant n + 1$,

$$\begin{aligned} a[i,j] &= E(i) + c_i^0 + c_i^1 d_{i,j} + \sum_{m=i+1}^{j-1} h_m^1 d_{m,j} \\ &= E(i) + c_i^0 + c_i^1 (d_{1,j} - d_{1,i}) \\ &\quad + \sum_{m=1}^{j-1} h_m^1 d_{m,j} - \sum_{m=1}^{i} h_m^1 (d_{1,j} - d_{1,m}) \\ &= \left[E(i) + c_i^0 - c_i^1 d_{1,i} + \sum_{m=1}^{i} h_m^1 d_{1,m} \right] \\ &\quad + \left[\sum_{m=1}^{j-1} h_m^1 d_{m,j} \right] + \left[\left(c_i^1 - \sum_{m=1}^{i} h_m^1 \right) d_{1,j} \right]. \end{aligned}$$

Now consider the $n \times (n+1)$ array $A' = \{a'[i,j]\}$ where

$$a'[i,j] = \left[E(i) + c_i^0 - c_i^1 d_{1,i} + \sum_{m=1}^{i} h_m^1 d_{1,m}\right] + \left[\sum_{m=1}^{j-1} h_m^1 d_{m,j}\right] + \left[\left(c_i^1 - \sum_{m=1}^{i} h_m^1\right) d_{1,j}\right]$$

for $1 \leq i \leq n$ and $1 \leq j \leq n+1$. If we can show that A' is Monge, then A must also be Monge, because every 2×2 subarray of A is either a 2×2 subarray of A' or its left- and bottommost entry is a $+\infty$.

To show that A' satisfies the Monge condition, we note that the first bracketed term in its definition depends only on i, the second bracketed term depends only on j, and the third bracketed term is the product of

$$c_i^1 - \sum_{m=1}^{i} h_m^1,$$

which depends only on i, and $d_{1,j}$, which depends only on j. Furthermore,

$$c_1^1 - \sum_{m=1}^{1} h_m^1 \geq c_2^1 - \sum_{m=1}^{2} h_m^1 \geq \ldots \geq c_n^1 - \sum_{m=1}^{n} h_m^1$$

(since, by assumption, $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$) and $0 = d_{1,1} \leq d_{1,2} \leq \ldots \leq d_{1,n+1}$ (since, by assumption, $d_i \geq 0$ for $1 \leq i \leq n$). Thus, by Corollary 1, A' is Monge.

Lemma 4. *Given $O(n)$ preprocessing time, we can compute $a[i,j]$ from $E(i)$ in constant time, for all i and j.*

Proof. If $i \geq j$, then $a[i,j] = +\infty$, i.e., computing the entry is easy. If, on the other hand, $i < j$, then

$$a[i,j] = E(i) + c_i^0 + c_i^1 d_{i,j} + \sum_{m=i+1}^{j-1} h_m^1 d_{m,j}.$$

Now suppose that we precompute $d_{1,i}$ for $1 \leq i \leq n$; this takes $O(n)$ time. This preprocessing gives us any $d_{i,j}$ in constant time, since $d_{i,j} = d_{1,j} - d_{1,i}$. Suppose that we also precompute

$$\sum_{m=1}^{j-1} h_m^1$$

for $2 \leq j \leq n+1$. This preprocessing again takes $O(n)$ time, and it allows us to precompute

$$\sum_{m=1}^{j-1} h_m^1 d_{m,j}$$

for $2 \leq j \leq n+1$ in an additional $O(n)$ time, since

$$\sum_{m=1}^{j-1} h_m^1 d_{m,j} = \sum_{m=1}^{j-2} h_m^1 d_{m,j-1} + \left(\sum_{m=1}^{j-1} h_m^1\right) d_{j-1}.$$

Moreover, since

$$\sum_{m=i+1}^{j-1} h_m^1 d_{m,j} = \sum_{m=1}^{j-1} h_m^1 d_{m,j} - \sum_{m=1}^{i} h_m^1 d_{m,i+1} - \left(\sum_{m=1}^{i} h_m^1\right) d_{i+1,j},$$

these precomputations allow us to compute $a[i,j]$ from $E(i)$ in constant time.

Given Lemmas 3 and 4, we can now apply Eppstein's on-line array-searching algorithm and obtain the following theorem.

Theorem 2. *Given an n-period instance of the basic economic lot size problem such that*

- *for $1 \leq i \leq n$,*

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 and c_i^1 are constants and $c_i^0 \geq 0$,
- *for $1 \leq i \leq n$, $h_i(y) = h_i^1 y$, where h_i^1 is a constant, and*
- *for $1 < i \leq n$, $c_i^1 \leq c_{i-1}^1 + h_i^1$,*

we can find an optimal production schedule in $O(n)$ time.

3.1.2. Arbitrary Coefficients

In this subsection, we remove the constraint that $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$ and allow the c_i^1 and h_i^1 to be arbitrary constants. This cost structure is the one considered by Zabel and by Eppen, Gould, and Pashigian. Both papers gave $O(n^2)$-time algorithms for this variant of the basic economic lot size problem; we improve this time bound to $O(n \lg n)$.

With arbitrary coefficients c_i^1 and h_i^1, the array A defined in the last subsection no longer satisfies the Monge condition, because we no longer have

$$c_1^1 - \sum_{m=1}^{1} h_m^1 \geq c_2^1 - \sum_{m=1}^{2} h_m^1 \geq \ldots \geq c_n^1 - \sum_{m=1}^{n} h_m^1.$$

However, we can circumvent this difficulty by reordering the rows of A. Intuitively, we sort the n quantities $r_1, \ldots, r_n$, where

$$r_i = c_i^1 - \sum_{m=1}^{i} h_m^1,$$

i.e., we find a permutation γ such that $r_{\gamma(1)} \geq r_{\gamma(2)} \geq \ldots \geq r_{\gamma(n)}$. If we then use γ to permute the rows of A, we obtain a new array that is more or less Monge.

We will now give a precise description of our $O(n \lg n)$-time algorithm for the basic economic lot size problem with nearly linear costs. The algorithm uses a divide-and-conquer approach, and it involves solving several subproblems, each corresponding to a range of consecutive periods. These subproblems are slightly more general than the basic economic lot size problem, in that solving the subproblem corresponding to periods s through $t - 1$ involves computing $E(j)$ for $s < j \leq t$, where $E(j)$ corresponds to an optimal production schedule for periods 1 through $j - 1$ (rather than periods s through $j - 1$). In particular, the schedule corresponding to $E(j)$ may have its last nonzero production occur in some period $i < s$.

To describe our algorithm in detail, we must first introduce some new notation. For $1 \leq s \leq n$ and $s < j \leq n + 1$, let

$$F_s(j) = \min_{1 \leq i < s} a[i, j].$$

Roughly speaking, $F_s(j)$ is the cost of the minimum cost production schedule satisfying the demands of periods 1 through $j - 1$ such that the inventory y_j carried forward from period $j - 1$ to period j is 0 and the schedule's last nonzero production occurs in some period $i < s$. For a subproblem that corresponds to periods $s, \ldots, t - 1$ and for $s < j \leq t$, we then have

$$E(j) = \begin{cases} E(j-1) & \text{if } d_{j-1} = 0, \\ \min \{F_s(j), \min_{s \leq i < t} a[i, j]\} & \text{if } d_{j-1} > 0. \end{cases}$$

Note that so long as $E(s)$ and $F_s(s + 1), \ldots, F_s(t)$ are known, the only entries of A that we need to consider in computing $E(s + 1), \ldots, E(t)$ are those lying in the subarray of A consisting of rows s through $t - 1$ and columns $s + 1$ through t. This subarray is depicted in Figure 5.

For $1 \leq s < t \leq n + 1$, we can now define the subproblem corresponding to periods s through $t - 1$ as follows. Given $E(s)$ and $F_s(s + 1), \ldots, F_s(t)$, solving this subproblem entails

1. computing $E(j)$ for $s + 1 \leq j \leq t$, and
2. sorting $r_s, \ldots, r_{t-1}$, where

$$r_i = c_i^1 - \sum_{m=1}^{i} h_m^1,$$

i.e., finding a permutation $\gamma_{s,t}$ such that

$$r_{\gamma s,t(1)+s-1} \geq r_{\gamma s,t(2)+s-1} \geq \ldots \geq r_{\gamma s,t(t-s)+s-1}.$$

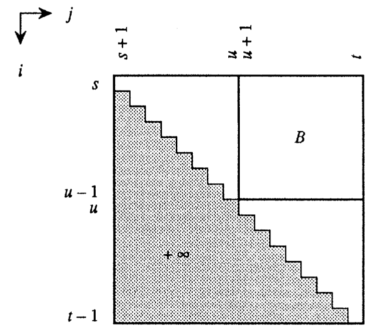

Figure 5. Given $E(s)$ and $F_s(s + 1), \ldots, F = (t)$, where $1 \leq s < t \leq n + 1$, we can compute $E(s + 1), \ldots, E(t)$ from the entries in rows s through t and columns $s + 1$ through $t + 1$ of A.

(Our reason for including the computation of $\gamma_{s,t}$ as part of the subproblem will become apparent in a moment.) Since $E(1) = 0$ and $F_1(j) = \infty$ for $2 \leq j \leq n + 1$, solving the subproblem corresponding to periods 1 through $n + 1$ gives us a solution for the original n-period economic lot size problem.

To solve the subproblem corresponding to periods s through $t - 1$ given $E(s)$ and $F_s(s + 1)$ through $F_s(t)$, we first recursively solve the subproblem corresponding to periods s through $u - 1$, where $u = \lfloor (s + t)/2 \rfloor$. This recursive computation is possible because $E(s)$ and $F_s(s + 1), \ldots, F_s(u)$ are known. Solving this subproblem gives us $E(s + 1)$ through $E(u)$ and the permutation $\gamma_{s,u}$.

Next, we compute the column minima of the subarray B consisting of rows s through $u - 1$ and columns $u + 1$ through t of A (see Figure 5). To find these column minima, we first permute the rows of B according to the permutation $\gamma_{s,u}$ obtained by solving the subproblem corresponding to periods s through $u - 1$. This permutation gives the $(u - s) \times (t - u)$ array $B' = b'[i, j)$, where

$$b'[i, j] = \Bigg[E(\gamma_{s,u}(i) + s - 1) + c^0_{\gamma s,u(i)+s-1} - c^1_{\gamma s,u(i)+s-1} d_{1,\gamma s,u(i)+s-1} + \sum_{m=1}^{\gamma_{s,u}(i)+s-1} h_m^1 d_{1,m}\Bigg] + \Bigg[\sum_{m=1}^{j+u-1} h_m^1 d_{m,j+u}\Bigg] + [r_{\gamma s,u(i)+s-1} d_{1,j+u}].$$

The column minima of B' are the column minima of B. Moreover, the first term in the sum that defines $b'[i, j]$ depends only on i, the second term depends only on j, and the third term is the product $r_{\gamma s,u(i)+s-1}d_{1,j+u}$, where

$$r_{\gamma s,u(1)+s-1} \geq r_{\gamma s,u(2)+s-1} \geq \ldots \geq r_{\gamma s,u(u-s)+2-1}$$

and

$$d_{1,u+1} \leq d_{1,u+2} \leq \ldots \leq d_{1,t}.$$

Thus, by Corollary 1, B' is Monge. Furthermore, using the $O(n)$-time preprocessing described in the previous subsection, we can compute any entry of B' in constant time, because $E(s+1)$ through $E(u)$ are known. Thus, we can apply the off-line array-searching algorithm of Aggarwal et al. directly and obtain the column minima of B in $O(t-s)$ time.

Given the column minima of B, we can now compute $F_u(u+1), \ldots, F_u(t)$, because for $u < j \leq t$,

$$F_u(j) = \min\left\{F_s(j), \min_{s \leq i < u} a[i, j]\right\}$$

and $\min_{s \leq i < u} a[i, j]$ is the minimum entry in the $(j-u)$th column of B. This computation requires only $O(t-u) = O(t-s)$ additional time.

Once $F_u(u+1)$ through $F_u(t)$ are known, we recursively solve the subproblem corresponding to periods u through $t-1$ using $E(u)$ and $F_u(u+1), \ldots, F_u(t)$. This recursive computation gives $E(u+1)$ through $E(t)$ and the permutation $\gamma_{u,t}$.

As the final step of our algorithm, we compute the permutation $\gamma_{s,t}$ from the permutations $\gamma_{s,u}$ and $\gamma_{u,t}$. This computation can be accomplished in $O(t-s)$ time by merging the two sorted lists of r_i's corresponding to $\gamma_{s,u}$ and $\gamma_{u,t}$. (We assume that for $1 \leq i \leq n$, $\sum_{m=1}^{i} h_m^1$ has been precomputed, so that any r_i can be computed in constant time; this preprocessing requires only $O(n)$ time.)

The running time $T(s, t)$ of this algorithm for the subproblem corresponding to periods s through $t-1$ is governed by the recurrence

$$T(s,t) = \begin{cases} T(s, \lfloor (s+t)/2 \rfloor) + T(\lfloor (s+t)/2 \rfloor, t) + O(t-s) & \text{if } t-s > 1, \\ O(1) & \text{if } t-s = 1, \end{cases}$$

which has as its solution $T(s, t) = O((t-s)\lg(t-s))$. Thus, $T(1, n+1) = O(n \lg n)$, which gives the following theorem.

Theorem 3. *Given an n-period instance of the basic economic lot size problem such that*

- *for* $1 \leq i \leq n$,

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 *and* c_i^1 *are constants and* $c_i^0 \geq 0$, *and*

- *for* $1 \leq i \leq n$, $h_i(y) = h_i^1 y$, *where* h_i^1 *is a constant,*

we can find an optimal production schedule in $O(n \lg n)$ *time.*

3.2. Other Cost Structures

In the previous subsection, we assumed nearly linear production costs and linear inventory costs. These assumptions allowed us to prove that certain arrays arising in the context of the basic economic lot size problem were Monge, and it was the Mongité of these arrays that allowed us to give improved algorithms for the basic problem with nearly linear costs. If we try to generalize this approach to arbitrary concave production and inventory cost functions (and improve upon the $O(n^2)$-time algorithm of Veinott 1963), however, note that the corresponding arrays need not be Monge. Consequently, the question of whether it is possible to obtain a subquadratic algorithm for the basic economic lot size problem with arbitrary concave costs remains open (see Section 6).

Note that even if the array $A = \{a[i, j]\}$ defined in the last subsection were Monge under less restrictive assumptions about the production and inventory cost functions, the computation of its column minima might still take $\Omega(n^2)$ time. This possibility stems from our need to be able to compute any entry $a[i, j]$ in constant time, given the minimum entries in columns 1 through i of A. In fact, there are cost structures where this entry computation time turns out to be a time bottleneck. Specifically, consider the cost structure studied by Zangwill (1969) (see also subsection 4.3 and Section 6). Zangwill assumed that for $1 \leq i \leq n$,

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 and c^1 are constants, and $h_i(\cdot)$ is a nondecreasing concave function. (Note that the marginal cost of production c^1 is the same for all time periods; this assumption is needed to ensure that the array A defined below is Monge.) If we consider the $n \times (n+1)$ array $A = \{a[i, j]\}$ where

$$a[i,j] = \begin{cases} E(i) + c_i^0 + c^1 d_{i,j} + \sum_{m=i+1}^{j-1} h_m(d_{m,j}) & \text{if } i < j, \\ +\infty & \text{if } i \geq j, \end{cases}$$

then A is Monge. This claim follows because

$$a[i, j] + a[i+1, j+1] - a[i, j+1] - a[i+1, j] = h_{i+1}(d_{i+1,j}) - h_{i+1}(d_{i+1,j+1})$$

for $1 < i + 1 < j < n + 1$, and the right-hand side of this equation is nonpositive so long as $h_{i+1}(\cdot)$ is a nondecreasing function. However, it is unclear how to compute $a[i, j]$ in constant time, given the minimum entries in columns 1 through i of A, as $a[i, j]$ depends on $\sum_{m=i+1}^{j-1} h_m^1(d_{m,j})$ and we do not know of any $O(n^2)$-time preprocessing that would allow us to compute this sum for any i and j in constant time.

4. THE BACKLOGGING PROBLEM

This section investigates the time complexity of the backlogging economic lot size problem under several different assumptions about the production, inventory, and backlogging cost functions. In subsection 4.1, we consider nearly linear production costs and linear inventory and backlogging costs. In subsection 4.2, we focus on arbitrary concave production, inventory, and backlogging cost functions. Finally, in subsection 4.3, we discuss arbitrary concave inventory and backlogging cost functions together with nearly linear production cost functions such that the marginal cost of production is the same for all periods.

4.1. Nearly Linear Costs

In this subsection, we give results for instances of the backlogging economic lot size problem with nearly linear costs. Specifically, for $1 \leq i \leq n$, we assume

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

$h_i(y) = h_i^1 y$, and $g_i(z) = g_i^1 z$, where the c_i^0, c_i^1, h_i^1 and g_i^1 are constants and c_i^0 is restricted to be nonnegative for $1 \leq i \leq n$. This problem is similar to the basic problem with nearly linear costs considered in subsection 3.1, except that here we are faced with a three-dimensional Monge array rather than a two-dimensional Monge array.

We begin with a special case in subsection 4.1.1: For $1 \leq i < n$, we assume that $c_i^1 \leq c_{i+1}^1 + g_i^1$ for $1 \leq i < n$ and $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$. For this special case of the backlogging problem, we give an $O(n)$-time algorithm for computing the optimal production schedule. Then, in subsection 4.1.2, we remove the constraint on the coefficients of the cost functions and give an $O(n \lg n)$-time algorithm for the backlogging problem with nearly linear costs.

4.1.1. Restricted Coefficients

In this subsection, we consider a nearly linear cost structure where the cost coefficients satisfy $c_i^1 \leq c_{i+1}^1 + g_i^1$ for $1 \leq i < n$ and $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$. This particular cost structure subsumes the cost structure considered by Morton. Morton gave an $O(n^2)$-time algorithm for his problem; we improve this time bound to $O(n)$ for our more general cost structure.

Recall the dynamic programming formulation of the backlogging economic lot size problem given in subsection 2.1: If we let $E(j)$ denote the minimum cost of satisfying the demands of periods 1 through $j - 1$ such that $y_j = 0$ (i.e., no inventory is stored from period $j - 1$ to period j, nor is any demand backlogged from period $j - 1$ to period j), then $E(1) = 0$ and for $2 \leq j \leq n + 1$,

$$E(j) = \min_{i \leq i \leq k < j} \left\{ E(i) + c_k(d_{i,j}) + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}) \right\},$$

where $d_{i,j} = d_i + d_{i+1} + \ldots + d_{j-1}$.

To compute $E(2), \ldots, E(n + 1)$ in $O(n)$ time, we consider the $n \times (n + 1) \times n$ array $A = \{a[i, j, k]\}$ where

$$a[i, j, k] = \begin{cases} E(i) + c_k^0 + c_k^1 d_{i,j} + \sum_{m=k+1}^{j-1} h_m^1 d_{m,j} & \text{if } i \leq k < j, \\ +\infty & \text{otherwise.} \end{cases}$$

Now if $d_{j-1} = 0$, then either some optimal production schedule produces during period $j - 1$, in which case

$$E(i) + c_{j-1}(d_{i,j}) + \sum_{m=i}^{j-2} g_m(d_{i,m+1}) + \sum_{m=j}^{j-1} h_m(d_{m,j}) = a[i, j, j - 1]$$

for $i < j$ and

$$E(j) = \min_{1 \leq i \leq n} a[i, j, j - 1] \leq E(j - 1),$$

or some optimal schedule does not produce during period j, in which case

$$E(j) = E(j - 1) \leq \min_{1 \leq i \leq n} a[i, j, j - 1].$$

On the other hand, if $d_{j-1} > 0$, then $d_{i,j} > 0$ for all $i < j$, which implies

$$E(i) + c_k(d_{i,j}) + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}) = a[i, j, k]$$

for $i \leq k < j$ and

$$E(j) = \min_{\substack{1 \leq i \leq n \\ 1 \leq k \leq n}} a[i, j, k].$$

These observations give the following recurrence for $E(j)$ when $2 \leq j \leq n+1$:

$$E(j) = \begin{cases} \min\{E(j-1), \min_{1 \leq i \leq n} a[i,j,j-1]\} & \text{if } d_{j-1}=0, \\ \min_{\substack{1 \leq i \leq n \\ 1 \leq k \leq n}} a[i,j,k] & \text{if } d_{j-1}>0. \end{cases}$$

Now observe that the three-dimensional array A can be decomposed into two two-dimensional arrays S and T. (Zangwill 1969 uses essentially this same decomposition to obtain an $O(n^2)$-time algorithm for a variant of this problem.) Specifically, let $S = \{s[i, l]\}$ denote the $n \times (n+1)$ array given by the equation

$$s[i,l] = \begin{cases} E(i) + c^0_{l-1} + c^1_{l-1} d_{i,l} + \sum_{m=i}^{l-2} g^1_m d_{i,m+1} & \text{if } i<l, \\ +\infty & \text{if } i \geq l, \end{cases}$$

and let $T = \{t[k, j]\}$ denote the $n \times (n+1)$ array given by

$$t[k,j] = \begin{cases} F(k) + c^1_k d_{k+1,j} + \sum_{m=k+1}^{j-1} h^1_m d_{m,j} & \text{if } k<j, \\ +\infty & \text{if } k \geq j. \end{cases}$$

where for $1 \leq k \leq n$,

$$F(k) = \min_{1 \leq i \leq n} s[i, k+1].$$

Since

$$\begin{aligned} \min_{\substack{1 \leq i \leq n \\ 1 \leq k \leq n}} a[i,j,k] &= \min_{1 \leq k<j} \left\{ \min_{1 \leq i \leq k} \left\{ E(i) + c^0_k + c^1_k d_{i,k+1} + \sum_{m=i}^{k-1} g^1_m d_{i,m+1} \right\} \right. \\ &\qquad \left. + c^1_k d_{k+1,j} + \sum_{m=k+1}^{j-1} h^1_m d_{m,j} \right\} \\ &= \min_{1 \leq k<j} \left\{ F(k) + c^1_k d_{k+1,j} + \sum_{m=k+1}^{j-1} h^1_m d_{m,j} \right\} \\ &= \min_{1 \leq k \leq n} t[k,j], \end{aligned}$$

and

$$\min_{1 \leq i \leq n} a[i,j,j-1] = \min_{1 \leq i \leq n} s[i,j],$$

we have

$$E(j) = \begin{cases} \min\{E(j-1), \min_{1 \leq i \leq n} s[i,j]\} & \text{if } d_{j-1}=0, \\ \min_{1 \leq k \leq n} t[k,j] & \text{if } d_{j-1}>0. \end{cases}$$

Thus, to compute $E(2), \ldots, E(n+1)$, we need merely compute the column minima of S and T. (For the reader familiar with Aggarwal and Park 1989, the three-dimensional array A is *path decomposable*; this structure is what allows the plane-minima problem for A to be decomposed into two column-minima problems for two-dimensional arrays.)

Using arguments similar to those used in proving Lemma 4, it is easy to verify that, after linear preprocessing time, any entry $s[i, k]$ of S can be computed in constant time given $E(i)$, and any entry $t[k, j]$ of T can be computed in constant time given $F(k)$. Furthermore, both S and T are Monge, as the following two lemmas show.

Lemma 5. *S is Monge.*

Proof. For $1 \leq i < l \leq n$,

$$\begin{aligned} s[i,l] &= E(i) + c^0_{l-1} + c^1_{l-1} d_{i,l} + \sum_{m=i}^{l-2} g^1_m d_{i,m+1} \\ &= E(i) + c^0_{l-1} + c^1_{l-1}(d_{1,l} - d_{1,i}) \\ &\quad + \sum_{m=1}^{l-2} g^1_m (d_{1,m+1} - d_{1,i}) - \sum_{m=1}^{i-1} g^1_m d_{i,m+1} \\ &= \left[E(i) - \sum_{m=1}^{i-1} g^1_m d_{i,m+1} \right] + \left[c^0_{l-1} + c^1_{l-1} d_{1,l} \right. \\ &\quad \left. + \sum_{m=1}^{l-2} g^1_m d_{1,m+1} \right] + \left[-\left(c^1_{l-1} + \sum_{m=1}^{l-2} g^1_m \right) d_{1,i} \right]. \end{aligned}$$

Now consider the $n \times n$ array $S' = \{s'[i, j]\}$, where

$$\begin{aligned} s'[i,j] &= \left[E(i) - \sum_{m=1}^{i-1} g^1_m \, d_{i,m+1} \right] \\ &\quad + \left[c^0_j + c^1_j \, d_{1,j+1} + \sum_{m=1}^{j-1} g^1_m \, d_{1,m+1} \right] \\ &\quad + \left[-\left(c^1_j + \sum_{m=1}^{j-1} g^1_m \right) d_{1,i} \right] \end{aligned}$$

for $1 \leq i \leq n$ and $1 \leq j \leq n$. (The array S' consists of columns 2 through $n+1$ of S with all the infinite entries in these columns replaced by finite values.) If we can show that S' is Monge, then S must also be Monge, because every 2×2 subarray of S is either a 2×2 subarray of S' or its left- and bottommost entry is a $+\infty$.

To show that S' is Monge, note that the first bracketed term in its definition depends only on i, the second bracketed term depends only on j, and the third bracketed term is the product of $d_{1,i}$, which depends only on i, and

$$-\left(c_j^1 + \sum_{m=1}^{j-1} g_m^1\right),$$

which depends only on j. Furthermore, $0 = d_{1,1} \leq d_{1,2} \leq \ldots \leq d_{1,n}$ (since, by assumption, $d_i \geq 0$ for $1 \leq i \leq n$), and

$$c_1^1 + \sum_{m=1}^{0} g_m^1 \leq c_2^1 + \sum_{m=1}^{1} g_m^1 \leq \ldots \leq c_n^1 + \sum_{m=1}^{n-1} g_m^1$$

(since, by assumption, $c_i^1 \leq c_{i+1}^1 + g_i^1$ for $1 \leq i < n$). Thus, by Corollary 1, S' is Monge.

Lemma 6. *T is Monge.*

Proof. The proof for this lemma is similar to that for Lemma 3.

At this point, we would like to apply one of the on-line array-searching algorithms mentioned in subsection 2.2. However, none of these algorithms can be applied directly, as $s[i, k]$ cannot be computed in constant time from the first through ith column minima of S, nor can $t[k, j]$ be computed in constant time from the first through kth column minima of T. To get around this difficulty, we use two interleaved processes, both running Eppstein's linear time, on-line array-searching algorithm. (Eppstein uses a similar approach to solve the modified string-editing problem.) The first process computes $E(2), \ldots, E(n+1)$ and the column minima of S, while the second process computes $F(1), \ldots, F(n)$ and the column minima of T. These computations occur in $2n - 1$ stages, where the first process is active only during the odd stages and the second process is active only during the even stages. In the first stage, the first process computes the second column minimum of S. In the second stage, the second process computes $F(1)$ and the second column minimum of T. Then for $2 \leq i \leq n$, the first process spends stage $2i - 1$ computing $E(i)$ and the $(i + 1)$st column minimum of S, and the second process spends stage $2i$ computing $F(i)$ and the $(i + 1)$st column minimum of T. Finally, in stage $2n - 1$, the first process computes $E(n + 1)$.

To bound the running time of the above procedure, note that since the ith column minimum of T, the ith column minimum of S, and $E(i - 1)$ are all known at the beginning of stage $2i - 1$, the first process can always compute $E(i)$ in constant time. Similarly, since the $(i + 1)$st column minimum of S is known at the beginning of stage $2i$, the second process can always compute $F(i)$ in constant time. Thus, both instances of Eppstein's on-line array-searching algorithm run in $O(n)$ time, which gives the following theorem.

Theorem 4. *Given an n-period instance of the backlogging economic lot size problem such that*

a. *for $1 \leq i \leq n$,*

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 and c_i^1 are constants and $c_i^0 \geq 0$,
b. *for $1 \leq i \leq n$, $h_i(y) = h_i^1 y$, where h_i^1 is a constant,*
c. *for $1 \leq i \leq n$, $g_i(z) = g_i^1 z$, where g_i^1 is a constant,*
d. *for $1 < i \leq n$, $c_i^1 \leq c_{i-1}^1 + h_i^1$, and*
e. *for $1 \leq i < n$, $c_i^1 \leq c_{i+1}^1 + g_i^1$,*

we can find an optimal production schedule in $O(n)$ time.

4.1.2. Arbitrary Coefficients

In this subsection, we allow c_i^1, g_i^1, and h_i^1 to be arbitrary constants, i.e., we no longer assume that $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$ and that $c_i^1 \leq c_{i+1}^1 + g_i^1$ for $1 \leq i < n$. This cost structure was considered by Blackburn and Kunreuther (1974) and by Lundin and Morton (1975). Both papers gave $O(n^2)$-time algorithms for this variant of the backlogging economic lot size problem; we improve this time bound to $O(n \log n)$.

As in subsection 3.1.2, if we allow arbitrary coefficients c_i^1, g_i^1, and h_i^1, then it is easy to verify that the arrays S and T defined in the previous subsection no longer satisfy the Monge condition. However, we can circumvent this difficulty by reordering the rows of S and T, just as we reordered the rows of A in subsection 3.1.2. Specifically, let

$$q_k = -\left(c_k^1 + \sum_{m=1}^{k} g_m^1\right)$$

for $1 \leq k \leq n$, and let

$$r_i = c_i^1 - \sum_{m=1}^{i} h_m^1$$

for $1 \leq i \leq n$. If we sort $q_1, \ldots, q_n$ and $r_1, \ldots, r_n$, obtaining permutations β and γ such that $q_{\beta(1)} \geq q_{\beta(2)} \geq \ldots \geq q_{\beta(n)}$ and $r_{\gamma(1)} \geq r_{\gamma(2)} \geq \ldots \geq r_{\gamma(n)}$, then the finite entries of $S' = \{s'[i, k]\}$ where $s'[i, k] = s[i, \beta(k)]$ and $T' = \{t'[k, j]\}$ where $t'[k, j] = t[\gamma(k), j]$ satisfy the Monge condition. Combining this observation with the divide-and-conquer approach of subsection 3.1.2, it is straightforward to obtain the following theorem.

Theorem 5. *Given an n-period instance of the backlogging economic lot size problem such that*

a. *for* $1 \leqslant i \leqslant n$,

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c_i^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 *and* c_i^1 *are constants and* $c_i^0 \geqslant 0$,
b. *for* $1 \leqslant i \leqslant n$, $g_i(y) = g_i^1 y$, *where* g_i^1 *is a constant, and*
c. *for* $1 \leqslant i \leqslant n$, $h_i(y) = h_i^1 y$,, *where* h_i^1 *is a constant,*

4.2. Concave Costs

In this subsection, we consider the backlogging economic lot size problem with arbitrary concave costs, i.e., we assume only that the cost functions $c_i(\cdot)$, $g_i(\cdot)$, and $h_i(\cdot)$ are concave. Zangwill (1966) gave an $O(n^3)$-time algorithm for this problem; we reduce this time bound to $O(n^2)$.

Let $A = \{a[i, j, k]\}$ denote the $n \times (n+1) \times n$ array where

$$a[i, j, k] = \begin{cases} E(i) + c_k(d_{i,j}) + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}) & \text{if } i \leqslant k < j, \\ +\infty & \text{otherwise.} \end{cases}$$

Furthermore, for $1 \leqslant k \leqslant n$, let $A_k = \{a_k[i, j]\}$ denote the $n \times (n+1)$ two-dimensional plane of A corresponding to those entries whose third coordinate is k, and let $B_k = \{b_k[s, t]\}$ denote the $k \times (n - k + 1)$ subarray of A_k consisting of rows 1 through k and columns $k+1$ through $n+1$ of A_k, so that $b_k[s, t] = a_k[s, t+k]$. (One such plane A_k and its subarray B_k are depicted in Figure 6.) Finally, for $1 \leqslant k < j \leqslant n+1$, let $F(j, k) = \min_{1 \leqslant i \leqslant n} a[i, j, k]$. The values $F(k+1, k)$, $F(k+2, k)$, ..., $F(n+1, k)$ are simply the column minima of B_k, and for $2 \leqslant j \leqslant n+1$, $E(j) = \min_{1 \leqslant k < j}\{F(j, k)\}$.

Our algorithm consists of n stages, each requiring $O(n)$ time. In the kth stage, we compute $F(k+1, k)$, $F(k+2, k)$, ..., $F(n+1, k)$ and then $E(k+1)$. For computing $E(k+1)$, $O(n)$ time clearly suffices, because $E(k+1)$ depends only on $F(k+1, 1)$, $F(k+1, 2)$, ..., $F(k+1, k)$ and we have already computed these values. Thus, all that remains to be shown is that we can compute the column minima of B_k in $O(n)$ time given $E(1)$, ..., $E(k)$. For such an argument, we need the following two lemmas.

Lemma 7. *B_k is inverse-Monge for all k in the range* $1 \leqslant k \leqslant n$.

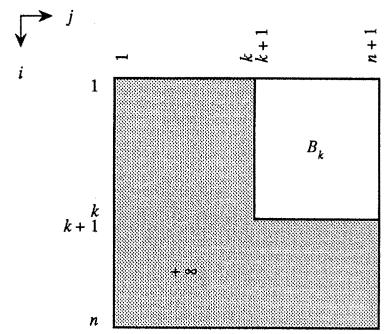

Figure 6. For any k in the range $l \leqslant k \leqslant n$, the only finite entries in the plane A_k lie in the subarray D_k consisting of rows 1 through k and columns $k+1$ through $n+1$ or A_k; moreover, all the entries in B_k are finite.

Proof. Consider any entry $b_k[s, t] = a_k[s, t+k]$ of B_k. Since $s \leqslant k$ and $t + k > k$, this entry is finite. In particular,

$$b_k[s, t] = E(s) + c_k(d_{s,t+k}) + \sum_{m=s}^{k-1} g_m(d_{s,m+1}) + \sum_{m=k+1}^{t+k-1} h_m(d_{m,t+k}).$$

Now observe that the terms $E(s)$ and $\sum_{m=s}^{k-1} g_m(d_{s,m+1})$ in the above depend only on s, and the term $\sum_{m=k+1}^{t+k-1} h_m(d_{m,t+k})$ depends only on t. Furthermore, $c_k(\cdot)$ is a concave function, $d_{s,t+k} = d_{1,t+k} - d_{1,s}$, and $0 = d_{1,1} \leqslant d_{1,2} \leqslant \ldots d_{1,n+1}$. Thus, by Lemmas 1 and 2, B_k is inverse Monge.

Lemma 8. *Given $O(n^2)$ preprocessing time, we can compute any entry of B_k in constant time for all k in the range* $1 \leqslant k \leqslant n$.

Proof. As we observed in the proof of the previous lemma,

$$b_k[s, t] = E(s) + c_k(d_{s,t+k}) + \sum_{m=s}^{k-1} g_m(d_{s,m+1}) + \sum_{m=k-1}^{t+k-1} h_m(d_{m,t+k})$$

for all s in the range $1 \leqslant s \leqslant k$ and all t in the range $1 \leqslant t \leqslant n - k + 1$. Now suppose that we precompute $d_{1,i}$ for all i in the range $1 \leqslant i \leqslant n$, which takes $O(n)$ time. This preprocessing gives us any $d_{i,j}$ in constant time, since $d_{i,j} = d_{1,j} - d_{1,i}$. Suppose that we also precompute

$$\sum_{m=i}^{k-1} g_m(d_{i,m+1})$$

for all i and k satisfying $1 \leqslant i \leqslant k \leqslant n$ and

$$\sum_{m=k+1}^{j-1} h_m(d_{m,j})$$

for all k and j satisfying $1 \leqslant k < j \leqslant n + 1$. This preprocessing takes an additional $O(n^2)$ time, since

$$\sum_{m=i}^{k-1} g_m(d_{i,m+1}) = g_{k-1}(d_{i,k}) + \sum_{m=i}^{k-2} g_m(d_{i,m+1})$$

and

$$\sum_{m=k+1}^{j-1} h_m(d_{m,j}) = h_{k+1}(d_{k+1,j}) + \sum_{m=k+2}^{j-1} h_m(d_{m,j}).$$

Moreover, these precomputations allow us to compute $b_k[s, t]$ in constant time, since by the time we consider B_k, $E(s)$ is known for all $s \leqslant k$.

Lemmas 7 and 8 allow us to apply the off-line array-searching algorithm of Aggarwal et al. to obtain the column minima of B_k from $E(1), \ldots, E(k)$ in $O(n)$ time. Thus, we have shown that each stage of our algorithm requires only $O(n)$ time, which gives the entire algorithm a running time of $O(n^2)$, including the preprocessing time required for Lemma 8.

Theorem 6. *Given an n-period instance of the backlogging economic lot size problem such that the $c_i(\cdot)$, $g_i(\cdot)$, and $h_i(\cdot)$ are concave functions, we can find an optimal production schedule in $O(n^2)$ time.*

4.3. Other Cost Structures

Zangwill (1969) considered yet another cost structure for the backlogging economic lot size problem: He assumed that for $1 \leqslant i \leqslant n$,

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 and c^1 are constants and $c_i^0 \geqslant 0$, and the $h_i(\cdot)$ and $g_i(\cdot)$ are nondecreasing concave functions. (Note that the marginal cost of production c^1 is the same for all time periods; this assumption is again needed to ensure that the arrays defined below are Monge.) If we consider the $n \times (n + 1) \times n$ array $A = \{a[i, j, k]\}$ where

$$a[i, j, k] = \begin{cases} E(i) + c_k^0 + c^1 d_{i,j} + \sum_{m=i}^{k-1} g_m(d_{i,m+1}) + \sum_{m=k+1}^{j-1} h_m(d_{m,j}) & \text{if } i \leqslant k < j, \\ +\infty & \text{otherwise.} \end{cases}$$

then A can be decomposed into two two-dimensional Monge arrays S and T as in subsection 4.1.1. These arrays are Monge because

1. $s[i, k] + s[i + 1, k + 1] - s([i, k + 1] - s[i + 1, k] = g_k(d_{i+1,k+1}) - g_k(d_{i,k+1})$ for $1 < i + 1 < k < n$, and the right-hand side of this equation is nonpositive so long as $g_k(\cdot)$ is a nondecreasing function, and

2. $t[k, j] + t[k + 1, j + 1] - t[k, j + 1] - t[k + 1, j] = h_{k+1}(d_{k+1,j}) - h_{k+1}(d_{k+1,j+1})$ for $1 < k + 1 < j < n + 1$, and the right-hand side of this equation is nonpositive so long as $h_{i+1}(\cdot)$ is a nondecreasing function.

However, it is unclear how to compute $s[i, k]$ and $t[k, j]$ in constant time given $o(n^2)$-time preprocessing; thus, we are unable to improve the running time of Zangwill's $O(n^2)$-time algorithm for the problem.

As a final remark, suppose that for all i and k such that $1 \leqslant i < k < n$, we knew

$$\sum_{m=i}^{k-1} g_m(d_{i,m+1}),$$

and similarly, for all k and j such that $1 \leqslant k < j \leqslant n + 1$, we knew

$$\sum_{m=k+1}^{j-1} h_m(d_{m,j}).$$

In this case, it is easy to see that any entry in row i of S and T could then be computed in constant time, given the minimum entries in columns 1 through i of S and T. Consequently, the column minima of S and T could then be computed in linear additional time using the approach of subsection 4.1.1. We will use this observation in subsection 5.1, as it helps us to obtain an improved algorithm for the periodic variant of the backlogging economic lot size problem considered by Erickson, Monma and Veinott.

5. TWO PERIODIC PROBLEMS

In this section, we present algorithms for two periodic variants of the backlogging economic lot size problem. The first was proposed by Erickson, Monma and Veinott, whereas the second was given by Graves and Orlin. Both problems assume that the planning horizon is infinite (i.e., we are planning for an infinite number of periods) but that demands and costs vary periodically over time with periodicity n, so that

$$d_{i+rn} = d_i,$$

$$c_{i+rn}(\cdot) = c_i(\cdot),$$

$$h_{i+rn}(\cdot) = g_i(\cdot), \text{ and}$$

$$g_{i+rn}(\cdot) = h_i(\cdot),$$

for $1 \leq i \leq n$ and all positive integers r. Erickson, Monma and Veinott consider the problem of finding a production schedule with periodicity n whose cost is minimum among all such schedules, whereas Graves and Orlin tackle the more difficult problem of finding a semi-infinite production schedule (starting with period 1, where the initial inventory is assumed to be 0) with minimum average cost per period.

5.1. Erickson, Monma and Veinott's Problem

Given an infinite planning horizon and periodic demands and costs, Erickson, Monma and Veinott considered the problem of finding an infinite production schedule with minimum average cost per period, subject to the restriction that the production schedule must have periodicity n, i.e., we must have $x_{i+rn} = x_i$ and $y_{i+rn} = y_i$ for $1 \leq i \leq n$ and all positive integers r. This problem is equivalent to finding the minimum cost, n-period production schedule for periods i through $n + i - 1$, where i is allowed to vary between 1 and n. In terms of network flows, this new problem is obtained from the backlogging economic lot size problem by adding two edges to the graph depicted in Figure 2, one from the nth sink to the first sink with concave cost function $h_1(\cdot)$ and the second from the first sink to the nth sink with concave cost function $g_n(\cdot)$.

For arbitrary concave costs $c_i(\cdot)$, $g_i(\cdot)$, and $h_i(\cdot)$, Erickson, Monma and Veinott gave an $O(n^3)$-time algorithm for their problem, which they obtained by solving n instances of the n-period backlogging economic lot size problem. For the special case where

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

and the $g_i(\cdot)$ and $h_i(\cdot)$ are nondecreasing, we can improve this bound to $O(n^2)$ time using the techniques of subsection 4.3: We merely spend $O(n^2)$ time to precompute

$$\sum_{m=i}^{k-1} g_m(d_{i,m+1})$$

for all i and k such that $1 \leq i < k \leq 2n$ and

$$\sum_{m=k+1}^{j-1} h_m(d_{m,j})$$

for all k and j such that $1 \leq k < j \leq 2n$, and then solve n instances of the n-period backlogging economic lot size problem in $O(n)$ time each.

Theorem 7. *Given an instance of Erickson, Monma and Veinott's economic lot size problem with periodicity n such that*

- *for $1 < i \leq n$,*

$$c_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ c_i^0 + c^1 x & \text{if } x > 0, \end{cases}$$

where c_i^0 and c^1 are constants and $c_i^0 \geq 0$, and

- *for $1 < i \leq n$, $g_i(\cdot)$ and $h_i(\cdot)$ are nondecreasing concave functions,*

we can find an optimal infinite production schedule with periodicity n in $O(n^2)$ time.

5.2. Graves and Orlin's Problem

Graves and Orlin consider another periodic variant of the backlogging economic lot size problem. They assume demands and costs are periodic, as do Erickson, Monma and Veinott, and seek an infinite production schedule (starting in period 1, where the initial inventory is assumed to be 0) with minimum average cost per period. Unlike Erickson, Monma and Veinott, however, they do not restrict the production schedule to have periodicity n. Instead, they assume that

$$\lim_{y \to \infty} G(y) = \lim_{y \to \infty} H(y) = \infty,$$

where

$$G(y) = g_1(y) + g_2(y) + \ldots + g_n(y)$$

and

$$H(y) = h_1(y) + h_2(y) + \ldots + h_n(y).$$

This assumption allows them to prove the following lemma.

Lemma 9. (Graves and Orlin 1985) *Let*

$$C = c_1(d_1) + c_2(d_2) + \ldots + c_n(d_n),$$

and let

$$D = d_1 + d_2 + \ldots + d_n.$$

Furthermore, let p denote the minimum integer such that $C < G(pD)$ and $C < H(pD)$. (Such a p exists by our assumption about the unboundedness of $G(\cdot)$ and $H(\cdot)$.) There is an optimal production schedule (i.e., a production schedule of minimum average cost per period) such that every interval of $2(p + 1)n$ consecutive periods contains at least one period with nonzero production.

This lemma and Theorem 1 (which also applies to infinite graphs; see Graves and Olin) together imply every production schedule must repeat after at most $2(p + 1)n^2$ periods, because an optimal production schedule starting from period i must be an optimal production schedule starting from period $i + rn$, for all integers r. Thus, there exists an optimal semi-infinite production schedule consisting of a finite production schedule with length at most $2(p + 1)n^2$ followed by an infinite periodic production schedule with period at most $2(p + 1)n^2$. In other words, there exist integers n_1 and n_2, both between 1 and $2(p + 1)n^2$, such that the optimal production schedule for periods 1 through n_1 (with no initial or final inventory and no initial or final backlogged demand) and the optimal production schedule for periods $n_1 + 1$ through $n_1 + n_2$ (again with no initial or final inventory or backlogging) together characterize the optimal infinite production schedule.

Graves and Orlin argued that such an optimal, semi-infinite production schedule can be computed in $O(p^3n^3)$ time. We reduce this bound to $O(p^2n^3)$ using Monge arrays.

To obtain a faster algorithm, we first compute an optimal production schedule for periods 1 through j, where j is allowed to vary from 1 to $2(p + 1)n^2$ and both the initial and final inventory and backlogging are required to be 0. Such a schedule can be computed in $O(p^2n^4)$ time by applying the techniques of subsection 4.2 directly, i.e., by computing the plane minima of an $O(pn^2) \times O(pn^2) \times O(pn^2)$ Monge array A. However, we can reduce this bound to $O(p^2n^3)$ time if we make use of Lemma 9. Specifically, since production in period j implies production in some period between $j - 2(p + 1)n$ and $j - 1$, we need only consider those entries $a[i, j, k]$ of A such that $j - 2(p + 1)n \leq i \leq k < j$. Roughly speaking, we can distribute these entries among $O(n)$ Monge arrays of size $O(pn) \times O(pn) \times O(pn)$ whose plane minima can be computed in $O(p^2n^2)$ time each.

Once we have an optimal production schedule for periods 1 through j for all j between 1 to $2(p + 1)n^2$, we can find the optimal infinite schedule as follows. For $1 \leq j \leq 2(p + 1)n^2$, we can identify the periodic and nonperiodic portions of the optimal production schedule for periods 1 through j and compute each portion's average cost per period in $O(pn)$ time per value of j, i.e., $O(p^2n^3)$ total time. Then, in $O(pn^2)$ additional time, we can select the value of j giving the best infinite production schedule, which gives us the following theorem.

Theorem 8. *Given an instance of Graves and Orlin's economic lot size problem with periodicity n, such that the $c_i(\cdot)$, $g_i(\cdot$, and $h_i(\cdot)$ are concave functions and p is defined as above, we can find an optimal semi-infinite production schedule in $O(p^2n^3)$ time.*

We remark here that since p depends upon the production, inventory, and backlogging costs and may be exponential in n, both Graves and Orlin's algorithm and our algorithm run in pseudopolynomial time; in fact, obtaining a true polynomial-time algorithm for this problem remains open.

6. SOME FINAL REMARKS

In this paper, we presented efficient dynamic programming algorithms for several variants of the economic lot size problem. These algorithms use properties of Monge arrays to improve the running times of previous algorithms, typically by factors of n and $n/\lg n$, where n denotes the number of periods under consideration. Aside from providing faster algorithms for economic lot size problems, a major contribution of this paper is our identification of the Monge arrays that arise in connection with economic lot size problems.

The algorithms given in this paper are easily extended to many other problems related to economic lot size models. For example, in his paper on Leontif substitution systems, Veinott (1969) showed that several other problems (including the product-assortment problem, the batch-queueing problem, the investment-consumption problem, and the reservoir-control problem) can be transformed into economic lot size problems (with or without backlogging).

Another model related to the economic lot size model that deserves special mention is the *capacity-expansion* model proposed by Manne and Veinott (1967). This model was developed by Manne (1967) during his study of four major industries in India between 1950 and 1965, and many researchers have studied problems formulated in terms of this model (see, for example, Fong and Rao 1975, Luss 1979, 1982, 1986, and Lee and Luss 1987). Manne and Veinott gave an $O(n^3)$-time algorithm for computing an optimal, feasible plan in their capacity-expansion model. Since their dynamic programming algorithm is identical to Zangwill's $O(n^3)$-time algorithm for solving the backlogging economic lot size problem with concave costs, the techniques of subsection 4.2 yield an $O(n^2)$-time algorithm for their problem. In a similar vein, several of the algorithms given in Sections 3 and 4 of this paper can be used to speed up various algorithms given by Luss (1982).

On a different note, this paper considered only production systems involving a single type of item and a single stage of production. However, other researchers (see Graves 1982 and Luss 1982, 1986, for example) have shown that the problem of computing an optimal plan for a multi-item and/or multistage production system can usually be decomposed into simpler problems using Lagrangian relaxation methods or simple heuristics that work fairly well in practice. Furthermore, these resulting problems can be expressed as economic lot size problems. The only difference between the economic lot size problems considered in this paper and those that result when computing optimal production schedules for such complex production systems is that this paper assumes demands are always nonnegative, whereas in the economic lot size problems resulting from Lagrangian relaxation methods or heuristics, the demands may be negative in certain situations. In other words, some of the demand nodes may be supply nodes; this supply has no cost, but it must be used up by any feasible production plan. Now when demands are negative, the arrays that occur in Sections 3–5 are not always Monge. Nevertheless, we show in a different paper (Aggarwal and Park, 1992) that the basic paradigm developed in this paper can still be applied and that the time complexities of the resulting algorithms are quite similar to those given in this paper.

In the Introduction, we mentioned some recent work by Federgruen and Tzur (1990, 1991) and by Wagelmans, van Hoesel and Kolen (1992), who have independently obtained several of the results that we present in this paper. We will not relate their work to our own.

Wagelmans, van Hoesel and Kolen presented an $O(n \lg n)$-time algorithm for the basic economic lot size problem with nearly linear costs. Their result matches the time bound of our algorithm for this problem which we described in subsection 3.1.2. Wagelmans, van Hoesel and Kolen also gave an $O(n)$-time algorithm for the special case of the basic economic lot size problem with nearly linear costs that we considered in subsection 3.1.1. (For this special case, we assumed that $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $1 < i \leq n$, i.e., the marginal cost of producing in period i is at most the marginal cost of producing in period $i - 1$ plus the marginal cost of storing inventory from period $i - 1$ to period i; this cost structure subsumes those considered by Manne (1958) and Wagner and Whitin (1958).) This result again matches the time bound of our algorithm for the problem.

These same two results—an $O(n \lg n)$-time algorithm for the basic economic lot size problem with nearly linear costs and a linear time algorithm for the special case of subsection 3.1.1—were independently derived by Federgruen and Tzur (1991). Moreover, Federgruen and Tzur also gave an $O(n)$-time algorithm for the basic economic lot size problem with nearly linear costs when setup costs are nondecreasing, i.e., $c_1^0 \leq c_2^0 \leq \ldots \leq c_n^0$ in the notation of subsection 3.1. Furthermore, Federgruen and Tzur (1990) gave an $O(n \lg n)$-time algorithm for the backlogging economic lot size problem with nearly linear costs, matching the result we described in subsection 4.1.2. They also gave a linear time algorithm for the special case of subsection 4.1.1 (again matching our result for this problem), as well as some additional special cases.

Both Federgruen and Tzur and Wagelmans, van Hoesel and Kolen used essentially the same techniques to obtain their results, and these techniques are substantially different from our own. Roughly speaking, they are computing (in an on-line fashion) the convex hull of n points in an appropriate two-dimensional space, whereas we are searching in Monge arrays. Note that neither Federgruen and Tzur nor Wagelmans, van Hoesel and Kolen were able to obtain results for the general backlogging economic lot size problem with arbitrary concave costs comparable to the results that we presented in subsection 4.2, which suggests that our techniques are in some sense more general.

We conclude with a list of open problems:

1. In subsection 3.1.2, we gave an $O(n \lg n)$-time algorithm for the basic economic lot size problem when the production and inventory costs are nearly linear, and in subsection 4.1.2, we gave an $O(n \lg n)$-time algorithm for the backlogging economic lot size problem when the production, inventory, and backlogging costs are nearly linear. It remains open whether there exists a $o(n \lg n)$-time algorithm for either of these problems.

2. Veinott (1963) showed that Wagner and Whitin's algorithm for the basic economic lot size problem can be used even when the production and inventory cost functions are arbitrary concave functions and that the resulting algorithm still takes $O(n^2)$ time. In this paper, we were unable to improve upon this bound (see Table I); thus, obtaining better time bounds for the basic problem with concave costs remains a challenging open problem. As pointed out in subsection 3.2, this problem remains open even for concave inventory costs and nearly linear production costs such that $c_i(0) = 0$ and $c_i(x) = c_i^0 + c^1 x$ for $x > 0$, where $c_i^0 \geq 0$. (This latter cost structure may greatly simplify the problem, since here the resulting array is Monge, as observed in subsection 3.2.)

3. Erickson, Monma and Veinott gave an $O(n^3)$-time algorithm for a periodic variant of the backlogging economic lot size problem. We were unable to

obtain a faster algorithm for this problem when the costs are arbitrary concave functions (see Table III); thus, obtaining a subcubic algorithm for this problem remains unresolved.

4. Graves and Orlin gave an $O(p^3n^3)$-time algorithm for another periodic variant of the backlogging economic lot size problem, and in this paper, we improved this bound to $O(p^2n^3)$. However, as mentioned in subsection 5.2, the parameter p in these running times may be exponential in n. Consequently, obtaining a true polynomial-time algorithm for Graves and Orlin's problem remains open (see Graves and Orlin 1985, for more details).

ACKNOWLEDGMENT

This research was done while the first author was visiting the Massachusetts Institute of Technology's Laboratory for Computer Science during the fall of 1989 and while the second author was a student there. The second author was supported in part by the Defense Advanced Research Projects Agency under contracts N00014-87-K-0825 and N00014-89-J-1988 and the Office of Naval Research under contract N00014-86-K-0593.

The authors thank Professor Eric V. Denardo (School of Organization and Management, Yale University), Professor Anjani Jain (Wharton School of Business, University of Pennsylvania), and Professor Arthur F. Veinott, Jr. (School of Operations Research, Stanford University) for providing valuable references and encouragement. The authors also thank Dr. Clyde Monma and Dr. Subhash Suri (Bell Communications Research), Professor Tom Leighton (Massachusetts Institute of Technology), and Dr. Baruch Schieber (IBM T. J. Watson Research Center) for stimulating discussions. Finally, for their helpful comments on our original submission, the authors thank the three anonymous referees who reviewed this paper.

REFERENCES

AGGARWAL, A., M. M. KLAWE, S. MORAN, P. W. SHOR AND R. WILBER. 1987. Geometric Applications of a Matrix-Searching Algorithm. *Algorithmica* **2,** 195–208.

AGGARWAL, A., AND J. K. PARK. 1989. Sequential Searching in Multidimensional Monotone Arrays. Research Report RC 15128, IBM T. J. Watson Research Center, Yorktown Heights, N.Y. J. Algorithms (submitted). Portions of this paper appear in *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science,* 1988, 497–512.

AGGARWAL, A., AND J. K. PARK. 1992. More Applications of Monge-Array Techniques to Economic Lot-Size Problems. (Manuscript in preparation.)

BAHL, H. C., L. R. RITZMAN AND J. N. D. GUPTA. 1987. Determining Lot Sizes and Resource Requirements: A Review. *Opns. Res.* **35,** 329–345.

BELLMAN, R. E. 1957. *Dynamic Programming.* Princeton University Press, Princeton, N. J.

BITRAN, G. R., AND H. H. YANASSE. 1982. Computational Complexity of the Capacitated Lot Size Problem. *Mgmt. Sci.* **28,** 1174–1186.

BLACKBURN, J. D., AND H. KUNREUTHER. 1974. Planning Horizons for the Dynamic Lot Size Model with Backlogging. *Mgmt. Sci.* **21,** 251–255.

CHUNG, C.-S., AND C.-H. M. LIN. 1988. An $O(T^2)$ Algorithm for NI/G/NI/ND Capacitated Lot Size Problem. *Mgmt. Sci.* **34,** 420–426.

DENARDO, E. V. 1982. *Dynamic Programming: Models and Applications.* Prentice-Hall, Englewood Cliffs, N. J.

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

EPPSTEIN, D. 1990. Sequence Comparison With Mixed Convex and Concave Costs. *J. Algorithms* **11,** 85–101.

ERICKSON, R. E., C. L. MONMA AND A. F. VEINOTT, JR. 1987. Send-and-Split Method for Minimum-Concave-Cost Network Flows. *Math. Opns. Res.* **12,** 634–664.

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, N. Y.

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,** 969–925.

FLORIAN, M., J. K. LENSTRA AND A. H. G. RINNOOY KAN. 1980. Deterministic Production Planning: Algorithms and Complexity. *Mgmt. Sci.* **26,** 669–679.

FONG, C. O., AND M. R. RAO. 1975. Capacity Expansion With Two Producing Regions and Concave Costs. *Mgmt. Sci.* **22,** 331–339.

GALIL, Z., AND K. PARK. 1990. A Linear-Time Algorithm for Concave One-Dimensional Dynamic Programming. *Infor. Proces. Letts.* **33,** 309–311.

GRAVES, S. C. 1982. Using Lagrangian Techniques to Solve Hierarchical Production Planning Problems. *Mgmt. Sci.* **28,** 260–275.

GRAVES, S. C., AND J. B. ORLIN. 1985. A Minimum Concave-Cost Dynamic Network Flow Problem With an Application to Lot-Sizing. *Networks* **15,** 59–71.

HARRIS, F. W. 1915. What Quantity to Make at Once.

In *Operation and Costs: Planning and Filling Orders, Cost-Keeping Methods, Controlling Your Operations, Standardizing Material and Labor Costs*, Vol. 5 of *The Library of Factory Management*, A. W. Shaw Co., Chicago, 47–52.

HAX, A. C., AND D. CANDEA. 1984. *Production and Inventory Management.* Prentice-Hall, Englewood Cliffs, N. J.

HOFFMAN, A. J. 1963. On Simple Linear Programming Problems. In *Convexity: Proceedings of the Seventh Symposium in Pure Mathematics of the AMS*, V. Klee, (ed.), Vol. 7 of *Proceedings of Symposia in Pure Mathematics*, American Mathematical Society, Providence, R. I., 317–327.

JOHNSON, L. A., AND D. C. MONTGOMERY. 1974. *Operations Research in Production Planning, Scheduling, and Inventory Control.* John Wiley, New York.

KLAWE, M. M. 1989. A Simple Linear Time Algorithm for Concave One-Dimensional Dynamic Programming. Technical Report 89-16, Department of Computer Science, University of British Columbia, Vancouver, Canada.

LARMORE, L. L., AND B. SCHIEBER. 1991. On-Line Dynamic Programming With Applications to the Prediction of RNA Secondary Structure. *J. Algorithms* **12,** 490–515.

LEE, C.-Y., AND E. V. DENARDO. 1986. Rolling Planning Horizons: Error Bounds for the Dynamic Lot Size Model. *Math. Opns. Res.* **11,** 423–432.

LEE, S.-B., AND H. LUSS. 1987. Multifacility-Type Capacity Expansion Planning: Algorithms and Complexities. *Opns. Res.* **35,** 249–253.

LOVÁSZ, L. 1983. Submodular Functions and Convexity. In *Mathematical Programming: The State of the Art, Bonn 1982.* A. Bachem, M. Grötschel and B. Korte (eds.). Springer-Verlag, New York, 235–257.

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.

LUSS, H. 1979. A Capacity Expansion Model for Two Facility Types. *Naval Res. Logist. Quart.* **26,** 291–303.

LUSS, H. 1982. Operations Research and Capacity Expansion Problems: A Survey. *Opns. Res.* **30,** 907–947.

LUSS, H. 1986. A Heuristic for Capacity Expansion Planning With Multiple Facility Types. *Naval Res. Logist. Quart.* **33,** 685–701.

MANNE, A. S. 1958. Programming of Economic Lot Sizes. *Mgmt. Sci.* **4,** 115–135.

MANNE, A. S. (ed.). 1967. *Investments for Capacity Expansion; Size, Location, and Time-Phasing.* MIT Press, Cambridge, Mass.

MANNE, A. S., AND A. F. VEINOTT, JR. 1967. Optimal Plant Size With Arbitrary Increasing Time Paths of Demand. In *Investments for Capacity Expansion; Size, Location, and Time-Phasing*, A. S. Manne (ed.). MIT Press, Cambridge, Mass., 178–190.

MORTON, T. E. 1978. An Improved Algorithm for the Stationary Cost Dynamic Lot Size Model With Backlogging. *Mgmt. Sci.* **24,** 869–873.

VEINOTT, JR., A. F. 1963. Unpublished Class Notes. Program in Operations Research, Stanford University, Stanford, Calif.

VEINOTT, JR., A. F. 1969. Minimum Concave-Cost Solution of Leontief Substitution Models of Multifacility Inventory Systems. *Opns. Res.* **17,** 262–291.

WAGELMANS, A., S. VAN HOESEL AND A. KOLEN. 1992. Economic Lot Sizing: An $O(n \log n)$ Algorithm That Runs in Linear Time in the Wagner-Whitin Case. *Opns. Res.* **40,** S145–S156.

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. 1975. *Principles of Operations Research, With Applications to Managerial Decisions*, 2nd ed. Prentice-Hall, Englewood Cliffs, N. J.

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

WILBER, R. 1988. The Concave Least-Weight Subsequence Problem Revisited. *J. Algorithms* **9,** 418–425.

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

ZANGWILL, W. I. 1966. A Deterministic Multi-Period Production Scheduling Model With Backlogging. *Mgmt. Sci.* **13,** 105–119.

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

ZANGWILL, W. I. 1969. A Backlogging Model and a Multiechelon Model of a Dynamic Economic Lot Size Production System—A Network Approach. *Mgmt. Sci.* **15,** 506–527.