A New Dynamic Programming Algorithm for the Single Item Capacitated Dynamic Lot Size Model*

HSIN-DER CHEN, DONALD W. HEARN, and CHUNG-YEE LEE
Industrial and Systems Engineering Department, University of Florida, Gainesville, Florida 32611, U.S.A.

(Received: 8 September 1992; accepted: 24 March 1993)

Abstract. We develop a new dynamic programming method for the single item capacitated dynamic lot size model with non-negative demands and no backlogging. This approach builds the optimal value function in piecewise linear segments. It works very well on the test problems, requiring less than 0.3 seconds to solve problems with 48 periods on a VAX 8600. Problems with the time horizon up to 768 periods are solved. Empirically, the computing effort increases only at a quadratic rate relative to the number of periods in the time horizon.

Key words. Capacitated lot size model, dynamic programming, production planning.

1. Introduction

The single item capacitated dynamic lot size model (CLSM) can be described as follows. For a finite time horizon T, there is demand for a single item in each production period. This demand must be satisfied by the production in that period or by inventory from previous periods, that is, no backlogging is allowed. The production level cannot exceed a certain capacity limit. Two kinds of costs are considered, production cost and holding cost. All the data, demands and capacities, as well as cost functions may differ from period to period. Assume that all data are non-negative and cost functions are non-decreasing. We are trying to find a feasible production plan with the total cost as small as possible.

This model was first studied by Wagner and Whitin (1958). They assumed that there is infinite capacity at each period and provided an $O(T^2)$ dynamic programming method. Since then, this uncapacitated model and its variations have been extensively studied by many researchers. We cite Veinott (1968), Zangwill (1966), Love (1973), Johnson and Montgomery (1974), Swoveland (1975), Florian *et al.* (1980) and Silver and Peterson (1985). Recently, Wagelmans *et al.* (1992), Federgruen and Tzur (1991), as well as Aggarwal and Park (1990) have independently provided algorithms to solve the problem in $O(T \log T)$ effort for the general problems and in $O(T)$ effort for problems with special cost structure.

* This research was supported in part by NSF grants DDM-8814075 and DMC-8504786.

The computational complexity of the CLSM has been investigated by Florian *et al.* (1980) and later by Bitran and Yanasse (1982). They showed that the CLSM is NP-hard even in many special cases. Therefore, one avenue of research has been to define some special cases and find polynomial algorithms for those cases. The notation $\alpha/\beta/\gamma/\delta$, which was introduced by Bitran and Yanasse (1982), where α, β, γ and δ represent set up cost, holding cost, production cost and capacity type respectively, describes families of the CLSM. The values of these parameters may be G, C, ND, NI and Z for arbitrary pattern, constant, non-decreasing, non-increasing and zero. Florian and Klein (1971) gave an $O(T^4)$ algorithm for $G/G/G/C$. Jagannathan and Rao (1973) extended this result to a general (in-period) cost function, neither concave nor convex. Swoveland (1975) adapted Florian and Klein's algorithm for piecewise concave cost functions. Love (1973) solved the same model with constant inventory capacity instead of production capacity in $O(T^3)$. Bitran and Yanasse showed that the time complexity of $NI/G/NI/ND$, $NI/G/NI/C$, $C/Z/C/G$ and $ND/Z/ND/NI$ are $O(T^4)$, $O(T^3)$, $O(T \log T)$ and $O(T)$. Chung and Lin (1988) improved the time complexity of $NI/G/NI/ND$ to $O(T^2)$.

For the general CLSM, Florian *et al.* (1980) suggested a pseudo-polynomial time dynamic programming approach with complexity $O(D_T C_T)$, where D_T is the total demand and C_T is the total capacity. Baker *et al.* (1978) provided a tree-search solution algorithm for the $G/G/C/G$ cases. For problems with $G/G/G/G$, there are two papers, by Lambrecht and Vander Eechen (1978) and by Kirca (1990). The former implemented the optimal property from Florian and Klein (1971) and designed a dynamic programming method. The latter refined the idea of pseudo-polynomial time dynamic program. Chung *et al.* (1990) provided an algorithm which combines dynamic programming and branch-and-bound. This algorithm works only for the special case without speculative motive for carrying inventory, that is, where the unit production cost increases no more than the cost of carrying a unit of inventory in each period.

Finally, the approach of defining facet cuts for the dynamic lot size model has been investigated by Barany *et al.* (1984a,b), Leung *et al.* (1989), and Pochet and Wolsey (1991).

This paper refines the pseudo-polynomial time dynamic programming approach by a geometric argument. The geometric argument to solve the uncapacitated dynamic lot size model has been used by Wagelmans *et al.* (1992) and by Chen and Lee (1991). Our work was motivated by these two papers. Although the result is an exponential time algorithm theoretically, it runs very well on test problems. In our computational experiments, it took only about 40 seconds on average to solve the CLSM with a time horizon up to 768 periods.

This paper is organized as follows. Section 2 gives the notation and formulations for the capacitated single item dynamic lot size model. Section 3 introduces our approach, Section 4 gives the algorithm and a numerical example, and Section 5 contains the computational experience.

2. Notation and Formulation

We need the following notation to describe the single item capacitated dynamic lot size model:

d_t	= demand in period t.
c_t	= production capacity in period t, where $c_t \geqslant 0$.
x_t	= production level in period t, where $0 \leqslant x_t \leqslant c_t$.
I_t	= inventory level at the end of period t.
$p_t(x_t)$	= function of production cost in period t.
K_t	= set-up cost in period t.
p'_t	= unit production cost in period t.
h'_t	= unit holding cost in period t.
D_t	= accumulated demand, that is, $D_t = \Sigma_{i=1}^{t} d_i$.
C_t	= accumulated capacity, that is, $C_t = \Sigma_{i=1}^{t} c_i$.

Without loss of generality, we may assume that $I_0 = 0$. The CLSM can be formulated as follows.

$$\text{Minimize} \sum_{t=1}^{T} p_t(x_t) + h'_t I_t$$

$$\text{subject to } I_{t-1} + x_t = d_t + I_t\,, \quad t = 1, \dots, T \tag{1a}$$

$$x_t \leqslant c_t\,, \quad t = 1, \dots, T \tag{1b}$$

$$I_0 = 0\,, \tag{1c}$$

$$x_t, I_t \geqslant 0\,, \quad t = 1, \dots, T \tag{1d}$$

where

$$p_t(x_t) = \begin{cases} 0 & \text{if } x_t = 0\,, \\ K_t + p'_t x_t & \text{if } 0 < x_t\,. \end{cases}$$

Substituting for the inventory variables $I_t = \Sigma_{i=1}^{t} x_i - \Sigma_{i=1}^{t} d_i$ and redefining $p_t(x_t)$, the above model can be rewritten as

$$\text{Minimize} \sum_{t=1}^{T} p_t(x_t) - \sum_{t=1}^{T} h'_t D_t$$

$$\text{subject to} \sum_{i=1}^{t} x_i \geqslant D_t\,, \quad t = 1, \dots, T \tag{2a}$$

$$x_t \geqslant 0\,, \quad t = 1, \dots, T \tag{2b}$$

where

$$p_t(x_t) = \begin{cases} 0 & \text{if } x_t = 0\,, \\ K_t + \left(p'_t + \sum_{i=t}^{T} h'_i\right) x_t & \text{if } 0 < x_t \leqslant c_t\,, \\ \infty & \text{otherwise}\,. \end{cases} \tag{3}$$

The above simplification shows that a CLSM model with linear holding costs can easily be transformed into an equivalent model with no holding cost. For notational convenience, we assume that $h_t' = 0$ for all t in the following discussion.

3. A Continuous Dynamic Programming Approach

For $t \geqslant 1$, we define the optimal value function to be

$$F_t(X) = \begin{cases} \begin{array}{l} \text{minimum cost of producing} \\ X = x_1 + \cdots + x_t \text{ units and} \\ \text{satisfying constraints 1} \\ \text{through } t \text{ of (2a) and (2b)} \end{array} & \text{for } X \in R_t \\ \infty & \text{for } X \not\in R_t \end{cases} \tag{4}$$

where

$$R_t = \{X \mid D_t \leqslant X \leqslant \min\{C_t, D_T\}\} ; \tag{5}$$

and let

$$F_0(X) = \begin{cases} 0 & \text{if } X = 0 \\ \infty & \text{if } X > 0 \end{cases}, \tag{6}$$

$$R_0 = \{0\} . \tag{7}$$

The optimal value function is then defined recursively as

$$F_t(X) = \min_{\substack{Y \in R_{t-1} \\ X - c_t \leqslant Y \leqslant X}} \{F_{t-1}(Y) + p_t(X - Y)\} \tag{8}$$

for $t \geqslant 1$ and $X \in R_t$. Then the optimal value of the objective function is

$$Z^* = F_T(D_T) . \tag{9}$$

The conventional interpretation for (8) is as follows. Given an $X \in R_t$, $F_t(X)$ is the minimal value of $F_{t-1}(Y) + p_t(X - Y)$ for all $Y \in R_{t-1} \cap \{Y \mid X - c_t \leqslant Y \leqslant X\}$. Typically X and Y are treated as discrete variables, and this approach implies the storing of tabular data at each stage of the dynamic program to solve for $F_T(D_T)$.

Here we treat X and Y as continuous variables. Hence $F_t(X)$ and $P_t(Y, X)$ are functions of these continuous variables, where

$$P_t(Y, X) = \begin{cases} F_{t-1}(T) + p_t(X - Y) & \text{if } Y \leqslant X \leqslant Y + c_t \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

for $Y \in R_{t-1}$, $X \in R_t$ and $t = 1, \ldots, T$. Then (8) can be rewritten as

$$F_t(X) = \min_{Y \in R_{t-1}} \{P_t(Y, X)\} . \tag{11}$$

Note that the restriction, $X - c_t \leq Y \leq X$, for Y in (8) is moved to the definition of $P_t(Y, X)$. Now the constraint on Y in (11) is independent of X. In our approach, (11) is interpreted as

> "Find all of the $P_t(Y, X)$ for every $X \in R_{t-1}$, then $F_t(X)$ is the lower envelope of all the $P_t(Y, X)$ functions."

One difficulty is left. Since Y is a continuous variable, there will be an infinite number of Y as well as $P_t(Y, X)$. Fortunately, if F_{t-1} is a piecewise linear function, $F_t(X)$ can be found without computing all $P_t(Y, X)$ over $Y \in R_{t-1}$. We will consider such Y in R_{t-1} segment by segment instead of point by point.

THEOREM 1. *For any $t = 1, \ldots, T$, $F_t(X)$ is a non-decreasing piecewise linear function.*

This theorem will be proven by induction. It is easy to see that $F_1(X) = p_1(X)$ is a non-decreasing piecewise linear function. Now suppose $F_{t-1}(X)$ is non-decreasing and piecewise linear for some $t > 1$. $F_t(X)$ will be shown to be non-decreasing and piecewise linear in the remainder of this proof.

Consider $p_t(x_t)$ in (3) for some t. There are two segments of $p_t(x_t)$ if it is treated as a piecewise linear function. When $x_t = 0$, that is, $X = Y$, it is easy to verify that

$$P_t(X, X) = F_{t-1}(X) + p_t(0) = F_{t-1}(X) \quad \text{for } X \in R_t , \tag{12}$$

and therefore (11) may be written as

$$F_t(X) = \min\{F_{t-1}(X), \min_{Y \in R_{t-1}, Y \neq X} \{P_t(Y, X)\}\} . \tag{13}$$

Since $F_{t-1}(Y)$ is a piecewise linear function by assumption, suppose there are m_{t-1} segments of $F_{t-1}(Y)$. Let

$$R_{t-1} = \bigcup_{1 \leq i \leq m_{t-1}} R^i_{t-1} \quad \text{for } t = 2, \ldots, T \tag{14}$$

where R^i_{t-1} is the domain corresponding to the ith segment of $F_{t-1}(Y)$ for $1 \leq i \leq m_{t-1}$. Then, (13) can be rewritten as

$$F_t(X) = \min\{F_{t-1}(X), \min_{1 \leq i \leq m_{t-1}} \{\bar{P}_t(i, X)\}\} , \tag{15}$$

where

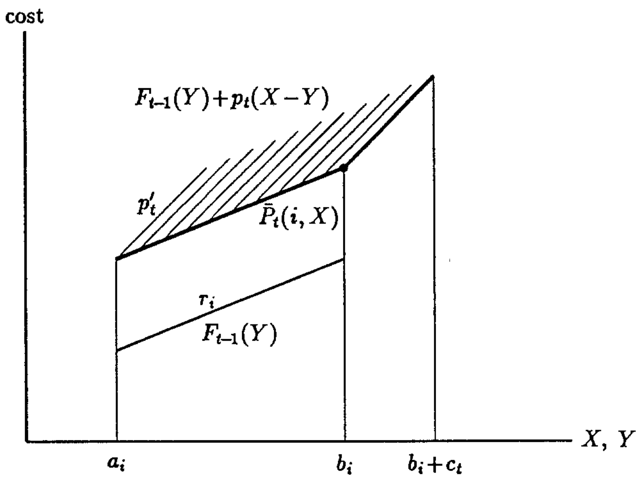

Fig. 1. Lower envelope of $F_{t-1}(Y)+p_t(X-Y)$ $(r_i \leqslant p'_t)$.

$$\bar{P}_t(i,X)=\min_{Y\in R^i_{t-1},\,Y\neq X}\{P_t(Y,X)\}\quad \text{for } 1\leqslant i\leqslant m_{t-1}\,. \tag{16}$$

Note that if there is a point of discontinuity in the domain of $F_{t-1}(Y)$, it must be left continuous at this point, since $F_{t-1}(Y)$ for any $t>1$ is a lower envelope of a set of non-decreasing functions. Let $R^1_{t-1}=\{Y\mid a_1\leqslant Y\leqslant b_1\}$ and $R^i_{t-1}=\{Y\mid a_i< Y\leqslant b_i\}$ for $i>1$. Denote the slope of $F_{t-1}(Y)$ in region R^i_{t-1} by r_i. See Figures 1 and 2. The thin lines with the slope of p'_t in these figures are $F_{t-1}(Y)+p_t(X-Y)$ over $Y<X\leqslant Y+c_t$ for $Y\in R^i_{t-1}$. The thick lines are the lower envelopes of such thin lines. Below are the algebraic expressions for these lower envelopes.

For any $t=2,\ldots,T$, and for any $1\leqslant i\leqslant m_{t-1}$, if $r_i\leqslant p'_t$, then

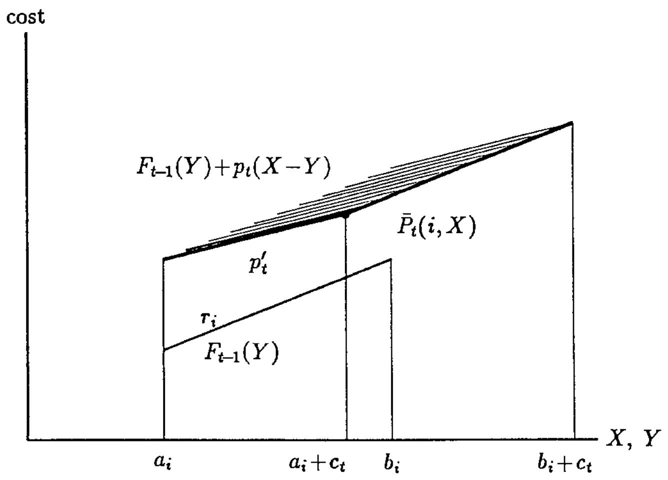

Fig. 2. Lower envelope of $F_{t-1}(Y)+p_t(X-Y)$ $(r_i>p'_t)$.

$$\bar{P}_t(i,X)=\begin{cases}F_{t-1}(X)+K_t & \text{for } a_i<X\leqslant b_i\\ F_{t-1}(b_i)+K_t+p_t'(X-b_i) & \text{for } b_i<X\leqslant b_i+c_t\,,\end{cases}\tag{17}$$

otherwise,

$$\bar{P}_t(i,X)=\begin{cases}F_{t-1}(a_i)+K_t+p_t'(X-a_i) & \text{for } a_i<X\leqslant a_i+c_t\,,\\ F_{t-1}(X-c_t)+K_t+p_t'c_t & \text{for } a_i+c_t<X\leqslant b_i+c_t\,.\end{cases}\tag{18}$$

Note that $\bar{P}_t(i,X)$ must also be defined on $X\in R_t$. In (17) and (18) it is implicit that $X\in R_t$.

Since $\bar{P}_t(i,X)$ is non-decreasing and piecewise linear for any $1\leqslant i\leqslant m_{t-1}$ as shown in (17) and (18) and $F_{t-1}(X)$ is non-decreasing piecewise linear function by assumption, $F_t(X)$, which is computed from (15), must be piecewise linear and non-decreasing. This proves Theorem 1.

4. Algorithm and Numerical Example

This section provides a pseudocode statement of the algorithm and a detailed numerical example. The algorithmic statement is based on preliminary observations regarding (17) and (18), and on a way to recover the optimal production plan after $F_T(X)$ is obtained.

For any $t=1,\ldots,T$, and for any $1\leqslant i\leqslant m_{t-1}$, let

$$\begin{aligned}
&\bar{P}_t^{(1)}(i,X)=F_{t-1}(X)+K_t && \text{for } a_i<X\leqslant b_i\\
&\bar{P}_t^{(2)}(i,X)=F_{t-1}(b_i)+K_t+p_t'(X-b_i) && \text{for } b_i<X\leqslant b_i+c_t\\
&\bar{P}_t^{(3)}(i,X)=F_{t-1}(a_i)+K_t+p_t'(X-a_i) && \text{for } a_i<X\leqslant a_i+c_t\\
&\bar{P}_t^{(4)}(i,X)=F_{t-1}(X-c_t)+K_t+p_t'c_t && \text{for } a_i+c_t<X\leqslant b_i+c_t\,.
\end{aligned}$$

These are segments of $\bar{P}_t(i,X)$ in (17) and (18). For $t\geqslant 1$, it is easy to see that $\bar{P}_t^{(1)}(i,X)=F_{t-1}(X)+K_t\geqslant F_{t-1}(X)$. Then $\bar{P}_t^{(1)}(i,X)$ will not contribute to $F_t(X)$ according to (15). For any $i>1$, since $F_t(X)$ is non-decreasing, $\bar{P}_t^{(3)}(i,X)\geqslant \bar{P}^{(2)}(i-1,b_{i-1})\geqslant F_t(X)$ for $a_i<X\leqslant a_i+c_t$ (note that $b_{i-1}=a_i$). Hence, $\bar{P}^{(3)}(i,X)$ will not contribute to $F_t(X)$ when $i>1$. Therefore, $\bar{P}_t(i,X)$ can be simplified as follows.

$$\bar{P}_t(i,X)=\begin{cases}\bar{P}_t^{(2)}(i,X) & \text{if } r_i\leqslant p_t'\\ \bar{P}_t^{(3)}(i,X)\cup\bar{P}_t^{(4)}(i,X) & \text{if } r_i>p_t' \text{ and } i=1\\ \bar{P}_t^{(4)}(i,X) & \text{if } r_i>p_t' \text{ and } i>1\end{cases}$$

for any $t\geqslant 1$ because this does not affect the recurrence in (15).

For convenience, we refer to $\bar{P}_t(i,X)$ as a *production line* of period t and refer to b_i when $r_i\leqslant p_t'$ or a_i when $r_i>p_t'$ as the *re-production point* of $\bar{P}_t(i,X)$.

In order to recover the optimal plan, we must keep the re-production points of each segment of the optimal value function for each period since they are used to determine the production level of the corresponding period. There are three possible production levels: (1) no production, (2) producing at full capacity, and (3) producing under capacity. To record these it is convenient to define another function, $G_t(X)$, from $F_t(X)$ as follows: if the segment containing X is from $\bar{P}_t^{(2)}(i, X)$ or $\bar{P}_t^{(3)}(i, X)$ (producing under capacity), let $G_t(X)$ be equal to the corresponding re-production point; if that segment is from $\bar{P}_t^{(4)}(i, X)$ (producing at full capacity), let $G_t(X)$ be equal to a large positive value; otherwise, let $G_t(X)$ be equal to a negative value. Accordingly, in the algorithm statement, let $G_t(X) = -D_T$ if no production is necessary in period t, $G_t(X) = D_T + c_t$ if producing at full capacity, and $G_t(X) =$ corresponding re-production point, otherwise.

Since X is a continuous variable, we assume there is a procedure, *lowerenv* which determines the lower envelope of a collection of functions. In our computer program, this procedure determines the lower envelope of two functions only, the $F_t(X)$ and a $\bar{P}_t^{(2)}(i, X)$ like function (a line segment). That is, we split the domain of these functions into several intervals in which both functions are only a line segment and compute the lower envelope in each interval. We also assume there is a procedure *findGt* which gives G_t as described above. Furthermore, $F_t(X)$ is implicitly infinite for $X \not\in R_t$, $|F_t(X)|$ represents the number of segments in $F_t(X)$, and a segment of one point is assumed to have slope zero.

CLSM ALGORITHM PSEUDOCODE

```
(Initialization)
$F_0(X) \leftarrow 0 \quad X \in R_0$
For $t = 1$ to $T$ do
  (Updating $F_t(X)$ for $X \in R_t$)
  $m_{t-1} \leftarrow |F_{t-1}(X)|$
  $F_t(X) \leftarrow F_{t-1}(X) \; X \in R_t$
  For $i = m_{t-1}$ down to 1 do
    If $r_i \leq p_t'$ then
      $F_t(X) \leftarrow lowerenv(F_t(X), \bar{P}_t^{(2)}(i, X))$
    else if $i = 1$ then
      $F_t(X) \leftarrow lowerenv(F_t(X), \bar{P}_t^{(3)}(i, X), \bar{P}_t^{(4)}(i, X))$
    else
      $F_t(X) \leftarrow lowerenv(F_t(X), \bar{P}_t^{(4)}(i, X))$
    endif
    $G_t(X) \leftarrow findGt(F_t(X))$
  enddo
enddo
```

```
(Recovery of the optimal plan)
$Z^* \leftarrow F_T(D_T)$
$Y \leftarrow D_T$
$t \leftarrow T$
repeat
    if $G_t(Y) < 0$ then
        $x_t \leftarrow 0$
        $Y \leftarrow Y$
    else if $G_t(Y) > D_T$ then
        $x_t \leftarrow G_t(Y) - D_T$
        $Y \leftarrow Y - x_t$
    else
        $x_t \leftarrow Y - G_t(Y)$
        $Y \leftarrow G_t(Y)$
    endif
    $t \leftarrow t - 1$
until $Y = 0$
```

A NUMERICAL EXAMPLE

t	d_t	D_t	K_t	p'_t	c_t
1	50	50	80	6	100
2	30	80	80	4	50
3	60	140	50	6	100
4	40	180	50	4	50

(initialization)

$$F_0(X) = 0 \quad \text{for } X = 0$$

$t = 1$

$$m_0 \leftarrow |F_0(X)| = 1$$

$$F_1(X) \leftarrow F_0(X) = 0 \quad \text{for } X = 0$$

$i = 1 \ (r_1 = 0 \leq p'_1 = 6)$

$$\bar{P}_1^{(2)}(1, X) \leftarrow 80 + 6X \quad \text{for } 50 < X \leq 100$$

$$F_1(X) \leftarrow \mathit{lowerenv}(F_1(X), \bar{P}_1^{(2)}(1, X)) = 80 + 6X \quad \text{for } 50 \leq X \leq 100$$

$$G_1(X) \leftarrow \mathit{findGt}(F_1(X)) = 0 \quad \text{for } 50 \leq X \leq 100$$

$t=2$

$$m_1 \leftarrow |F_1(X)| = 1$$

$$F_2(X) \leftarrow F_1(X) = 80 + 6X \quad \text{for } 8 \leqslant X \leqslant 100$$

$i=1$ $(r_1 = 6 > p_2' = 4)$

$$\bar{P}_2^{(3)}(1, X) \leftarrow 260 + 4X \quad \text{for } 50 < X \leqslant 100$$

$$\bar{P}_2^{(4)}(1, X) \leftarrow 60 + 6X \quad \text{for } 100 < X \leqslant 150$$

$$F_2(X) \leftarrow \mathit{lowerenv}(F_2(X), \bar{P}_1^{(3)}(1, X), \bar{P}_2^{(4)}(1, X))$$

$$= \begin{cases} 80 + 6X & \text{for } 80 \leqslant X \leqslant 90 \\ 260 + 4X & \text{for } 90 < X \leqslant 100 \\ 60 + 6X & \text{for } 100 < X \leqslant 150 \end{cases}$$

$$G_2(X) \leftarrow \mathit{findGt}(F_2(X)) = \begin{cases} -180 & \text{for } 80 \leqslant X \leqslant 90 \\ 50 & \text{for } 90 < X \leqslant 100 \\ 230 & \text{for } 100 < X \leqslant 150 \end{cases}$$

$t=3$

$$m_2 \leftarrow |F_2(X)| = 3$$

$$F_3(X) \leftarrow F_2(X) = 60 + 6X \quad \text{for } 140 \leqslant X \leqslant 150$$

$i=3$ $(r_3 = 6 \leqslant p_3' = 6)$

$$\bar{P}_3^{(2)}(3, X) \leftarrow 110 = 6X \quad \text{for } 150 < X \leqslant 250$$

$$F_3(X) \leftarrow \mathit{lowerenv}(F_3(X), \bar{P}_3^{(2)}(3, X)) = \begin{cases} 60 + 6X & \text{for } 140 \leqslant X \leqslant 150 \\ 110 + 6X & \text{for } 150 < X \leqslant 250 \end{cases}$$

$$G_3(X) \leftarrow \mathit{findGt}(F_3(X)) = \begin{cases} -180 & \text{for } 140 \leqslant X \leqslant 150 \\ 150 & \text{for } 150 < X \leqslant 250 \end{cases}$$

$i=2$ $(r_2 = 4 \leqslant p_3' = 6)$

$$\bar{P}_3^{(2)}(2, X) \leftarrow 110 + 6X \quad \text{for } 100 < X \leqslant 200$$

$$F_3(X) \leftarrow \mathit{lowerenv}(F_3(X), \bar{P}_3^{(2)}(2, X)) = \begin{cases} 60 + 6X & \text{for } 140 \leqslant X \leqslant 150 \\ 110 + 6X & \text{for } 150 < X \leqslant 250 \end{cases}$$

$$G_3(X) \leftarrow \mathit{findGt}(F_3(X)) = \begin{cases} -180 & \text{for } 140 \leqslant X \leqslant 150 \\ 150 & \text{for } 150 < X \leqslant 250 \end{cases}$$

$i=1$ $(r_1 = 6 \leqslant p_3' = 6)$

$$\bar{P}_3^{(2)}(1, X) \leftarrow 130 + 6X \quad \text{for } 90 < X \leqslant 190$$

$$F_3(X) \leftarrow \mathit{lowerenv}(F_2(X), \bar{P}_3^{(2)}(1, X)) = \begin{cases} 60 + 6X & \text{for } 140 \leqslant X \leqslant 150 \\ 110 + 6X & \text{for } 150 < X \leqslant 250 \end{cases}$$

$$G_3(X) \leftarrow \mathit{findGt}(F_3(X)) = \begin{cases} -180 & \text{for } 140 \leqslant X \leqslant 150 \\ 150 & \text{for } 150 < X \leqslant 250 \end{cases}$$

$t = 4$

$$m_3 \leftarrow |F_3(X)| = 2$$

$$F_4(X) \leftarrow F_3(X) = 110 + 6X \quad \text{for } 180 \leqslant X \leqslant 250$$

$i = 2$ $(r_2 = 6 > p'_4 = 4)$

$$\bar{P}_4^{(4)}(2, X) \leftarrow 60 + 6X \quad \text{for } 200 \leqslant X \leqslant 300$$

$$F_4(X) \leftarrow \mathit{lowerenv}(F_4(X),\ \bar{P}_4^{(4)}(2, X)) = \begin{cases} 110 + 6X & \text{for } 180 \leqslant X \leqslant 200 \\ \ 60 + 6X & \text{for } 200 < X \leqslant 300 \end{cases}$$

$$G_4(X) \leftarrow \mathit{findGt}(F_4(X)) = \begin{cases} -180 & \text{for } 180 \leqslant X \leqslant 200 \\ \ 150 & \text{for } 200 < X \leqslant 300 \end{cases}$$

$i = 1$ $(r_1 = 6 > p'_4 = 4)$

$$\bar{P}_4^{(3)}(1, X) \leftarrow 390 + 4X \quad \text{for } 140 < X \leqslant 190$$

$$\bar{P}_4^{(4)}(1, X) \leftarrow 10 + 6X \quad \text{for } 190 < X \leqslant 200$$

$$F_4(X) \leftarrow \mathit{lowerenv}(F_4(X),\ \bar{P}_4^{(3)}(1, X), \bar{P}_4^{(4)}(1, X))$$

$$= \begin{cases} 390 + 4X & \text{for } 180 \leqslant X \leqslant 190 \\ \ 10 + 6X & \text{for } 190 < X \leqslant 200 \\ \ 60 + 6X & \text{for } 200 < X \leqslant 300 \end{cases}$$

$$G_4(X) \leftarrow \mathit{findGt}(F_4(X)) = \begin{cases} 140 & \text{for } 180 \leqslant X \leqslant 190 \\ 230 & \text{for } 190 < X \leqslant 200 \\ 150 & \text{for } 200 < X \leqslant 300 \end{cases}$$

(Recovery of the optimal plan)

$$\begin{cases} Z^* \leftarrow F_4(D_4) = F_4(180) = 1110 \\ \ Y \leftarrow D_4 = 180 \end{cases}$$

$$\begin{cases} \qquad t = 4 \\ G_t(Y) = G_4(180) = 140 < D_T \\ \qquad x_4 \leftarrow Y - G_t(Y) = 180 - 140 = 40 \\ \qquad Y \leftarrow G_t(Y) = 140 \end{cases}$$

$$\begin{cases} \qquad t = 3 \\ G_t(Y) = G_3(140) = -180 < 0 \\ \qquad x_3 \leftarrow 0 \\ \qquad Y \leftarrow Y = 140 \end{cases}$$

$$\begin{cases} \qquad t = 2 \\ G_t(Y) = G_2(140) = 230 > D_T \\ \qquad x_2 \rightarrow G_t(Y) - D_T = 230 - 180 = 50 \\ \qquad Y \leftarrow Y - x_t = 140 - 50 = 90 \end{cases}$$

$$\begin{cases} \quad t=1 \\ G_t(Y)=G_1(90)=0<D_T \\ \quad x_1 \leftarrow Y-G_t(Y)=90-0=90 \\ \quad Y \leftarrow G_t(Y)=0 \end{cases}$$

The optimal plan is:

$$\begin{cases} x_1=90 \\ x_2=50 \\ x_3=0 \\ x_4=40\,. \end{cases}$$

5. Computational Experience

We have coded the dynamic programming algorithm given in the previous section. The storage of the data at each stage is accomplished by using (17) and (18). That is, we only keep the function parameters and the upper and lower bounds. This offers considerable savings over a standard dynamic programming approach.

We used the same problem pattern as that in Baker *et al.* (1978) to test our algorithm.

The demand pattern is given by

$$d_t = 200 + \sigma z_t + a \sin\left[\frac{2\pi}{b}(t+b/4)\right] \tag{19}$$

where

$\sigma =$ standard error of demand,
$z_t =$ i.i.d. standard normal random variable,
$a =$ amplitude of the seasonality component,
$b =$ length of seasonal cycle in periods.

Five problems were created in each of the following four combinations: (1) $\sigma=67$, $a=0$, (2) $\sigma=237$, $a=0$, (3) $\sigma=67$, $a=125$, $b=T$ and (4) $\sigma=67$, $a=125$, $b=12$.

In the first run, unit production cost is zero for each period, $p'_t=0$; and unit holding cost is one, $h'_t=1$. Three different set up costs K are chosen, which are 100, 900 and 3600. Three constant capacity levels C are selected, that is, 250, 700 and 1200. We tested 6 time horizons, $T=24$, 48, 96, 192, 384 and 768. These problems are $C/C/C/C$.

After the first run, we created another set of problems with general capacity. Capacity c_t is generated from an independent uniform distribution with range $(0.5C,\ 1.5C)$ for each capacity category. These problems are $C/C/C/G$. In addition, we generate p_t in the range of $(4, 6)$, h_t in the range of $(0.5, 1.5)$ and K_t

in the range of $(0.5K, 1.5K)$ for each set up cost category. Thus, we have test problems with $G/G/G/G$. We only tested the case of $T = 192$ in the second run.

A total of 1440 feasible problems were created and tested. The program of our algorithm is coded in Fortran and run on a VAX 8600.

For any algorithm, the concern is both computing time and working space. In our algorithm, working space is in proportion to the total segment number of $G_t(X)$ over all periods and/or the maximal number of segments of $F_t(X)$. As mentioned, we have to keep every $G_t(X)$ in order to recover the optimal production plan.

The results of the first run are in Tables I, II and III. We do not have a comparison with results from the literature since no other algorithm has been used to solve problems of more than 24 periods with general capacity and cost structure. In these tables, we arrange results in the categories of set up cost, capacity and time horizon. Thus, each cell represents 20 problems. Table I contains CPU time, Table II contains the maximal segment number of $F_t(X)$, and Table III contains the total segments number of $G_t(X)$. In each category, average performance is listed first with worst case performance in parentheses below.

As the tables show, the algorithm is very effective. For example, within 0.3 seconds, the code finds the optimal solution for the case of $T = 48$, which is a plan for a year with weekly lots. Even when the time horizon is 768, it needs less than 3 minutes for the worst case out of 180 tests. Two more things should be mentioned. First, computing time increases only at a quadratic rate and storage space increases at a linear rate relative to T. Secondly, the performance of our

Table I. CPU time (seconds on a VAX 8600)

K	C	$T = 24$	$T = 48$	$T = 96$	$T = 192$	$T = 384$	$T = 768$
100	250	0.02[a]	0.08	0.32	1.15	4.71	19.05
		(0.03)[b]	(0.10)	(0.38)	(1.35)	(5.49)	(22.24)
	700	0.01	0.05	0.17	0.73	3.05	12.37
		(0.03)	(0.06)	(0.21)	(0.95)	(3.90)	(15.44)
	1200	0.01	0.03	0.12	0.46	1.82	7.15
		(0.02)	(0.05)	(0.16)	(0.59)	(2.32)	(9.17)
900	250	0.04	0.16	0.68	2.68	11.00	47.26
		(0.05)	(0.23)	(1.14)	(3.86)	(17.24)	(71.74)
	700	0.03	0.12	0.46	1.89	7.87	32.71
		(0.04)	(0.15)	(0.51)	(2.24)	(8.93)	(35.31)
	1200	0.01	0.06	0.21	0.88	3.57	15.04
		(0.02)	(0.08)	(0.24)	(0.95)	(3.84)	(16.42)
3600	250	0.04	0.19	1.04	5.19	26.16	99.65
		(0.06)	(0.29)	(2.11)	(7.93)	(46.26)	(163.11)
	700	0.03	0.20	0.98	4.45	19.07	84.82
		(0.04)	(0.27)	(1.19)	(5.07)	(21.57)	(95.57)
	1200	0.02	0.10	0.48	2.08	8.94	36.60
		(0.03)	(0.13)	(0.61)	(2.39)	(10.39)	(40.92)

[a] average; [b] worst case.

Table II. Maximal segments of $F_t(X)$ over time horizon

K	C	$T = 24$	$T = 48$	$T = 96$	$T = 192$	$T = 384$	$T = 768$
100	250	19.5[a]	32.3	62.2	115.2	223.9	447.6
		(25)[b]	(39)	(79)	(147)	(293)	(554)
	700	12.4	20.0	36.7	68.4	135.6	264.1
		(18)	(27)	(50)	(91)	(174)	(340)
	1200	9.4	13.6	23.4	41.8	80.8	155.9
		(12)	(18)	(30)	(55)	(105)	(206)
900	250	37.4	73.2	151.4	273.0	550.7	1134.2
		(53)	(108)	(273)	(439)	(977)	(1924)
	700	23.4	49.8	92.4	180.4	363.3	713.0
		(32)	(60)	(112)	(203)	(409)	(783)
	1200	14.1	25.5	44.9	86.2	166.6	329.4
		(18)	(30)	(51)	(96)	(191)	(373)
3600	250	40.8	94.1	249.9	566.6	1343.4	2894.4
		(59)	(134)	(520)	(976)	(2672)	(5332)
	700	29.3	84.3	191.6	418.0	871.0	1805.8
		(38)	(116)	(257)	(509)	(1052)	(2058)
	1200	17.8	44.7	101.3	205.3	428.7	840.9
		(27)	(57)	(140)	(231)	(487)	(955)

[a] average; [b] worst case.

Table III. Total segments of $G_t(X)$ over time horizon

K	C	$T = 24$	$T = 48$	$T = 96$	$T = 192$	$T = 384$	$T = 768$
100	250	41.9[a]	83.4	171.1	342.8	686.5	1376.2
		(46)[b]	(88)	(184)	(351)	(707)	(1400)
	700	49.5	101.3	205.3	412.5	836.0	1666.5
		(54)	(110)	(222)	(440)	(888)	(1778)
	1200	47.3	99.3	203.4	411.8	835.0	1667.9
		(54)	(111)	(219)	(442)	(891)	(1786)
900	250	47.4	95.3	199.9	400.1	801.0	1605.5
		(55)	(107)	(222)	(425)	(854)	(1693)
	700	64.3	134.1	273.7	550.0	1108.2	2223.6
		(67)	(139)	(283)	(571)	(1146)	(2287)
	1200	62.8	133.7	274.9	557.0	1122.6	2255.6
		(66)	(138)	(281)	(570)	(1144)	(2291)
3600	250	48.1	98.9	211.3	425.6	854.1	1719.9
		(56)	(111)	(234)	(460)	(924)	(1840)
	700	64.6	136.4	278.8	562.3	1132.4	2272.4
		(67)	(140)	(283)	(572)	(1148)	(2299)
	1200	62.0	134.6	278.1	564.3	1139.2	2290.3
		(66)	(138)	(281)	(569)	(1145)	(2297)

[a] average; [b] worst case.

algorithm is very stable. The time and space requirement for the worst case in each category is less than twice that of the average case.

The results of the second run are listed in Table IV. Only CPU time is provided. The purpose of the second run is to show that the pattern of problems, whether easy or difficult, does not affect the performance of our algorithm. This is

Table IV. CPU time ($T = 192$)

K	C	$C/C/C/C$	$C/C/C/G$	$G/G/G/G$
100	250	1.15[a]	1.19	1.26
		(1.35)[b]	(1.42)	(1.51)
	700	0.73	0.81	0.89
		(0.95)	(1.02)	(1.09)
	1200	0.46	0.48	0.52
		(0.59)	(0.60)	(0.61)
900	250	2.68	3.17	3.05
		(3.86)	(5.14)	(4.70)
	700	1.89	2.18	2.13
		(2.24)	(2.42)	(2.41)
	1200	0.88	0.97	0.99
		(0.95)	(1.07)	(1.16)
3600	250	5.19	8.31	7.08
		(7.93)	(14.76)	(10.66)
	700	4.45	5.45	4.59
		(5.07)	(6.35)	(5.17)
	1200	2.08	2.38	2.04
		(2.39)	(2.71)	(2.54)

[a] average; [b] worst case.

important because our approach does not require special properties of capacities or of the cost functions.

The computer code used in our experiments is available from the authors. The distribution disk includes a code for generation of the test data.

References

Aggarwal, A. and J. K. Park (1990), Improved Algorithms for Economic Lot-Size Problem, Working paper, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Baker, K. R., P. Dixon, M. J. Magazine, and E. A. Silver (1978), An algorithm for the Dynamic Lot-Size Problem with Time-Varying Production Capacity Constraints, *Management Science* **24**, 1710–1720.

Barany, I., T. Van Roy, and L. A. Wolsey (1984a), Uncapacitated Lot-Sizing: the Convex Hull of Solutions, *Mathematical Programming Study* **22**, 32–43.

Barany, I., T. Van Roy, and L. A. Wolsey (1984b), Strong Formulations for Multi-Item Capacitated Lot Sizing, *Management Science* **30**, 1255–1261.

Bitran, G. R. and H. H. Yanasse (1982), Computational Complexity of the Capacitated Lot Size Problem, *Management Science* **28**, 1174–1186.

Chen, H.-D. and C.-Y. Lee (1991), A Simple Algorithm for the Error Bound of the Dynamic Lot Size Model Allowing Speculative Motive, Research Report 91-5, Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida. (Revised for IIE Transactions.)

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

Chung, C.-S., J. Flynn, and C.-H. M. Lin (1990), An Efficient Algorithm for the Capacitated Lot Size Problem, Working paper, College of Business Administration, Cleveland State University, Cleveland, Ohio.

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, *Management Science* **37**, 909–925.

Florian, M. and M. Klein (1971), Deterministic Production Planning with Concave Costs and Capacity Constraints, *Management Science* **18**, 12–20.

Florian, M., J. K. Lenstra, and A. H. G. (1980), Rinnooy Kan, Deterministic Production Planning Algorithm and Complexity, *Management Science* **26**, 669–679.

Jagannathan, R. and M. R. Rao (1973), A Class of Deterministic Production Planning Problems, *Management Science* **19**, 1295–1300.

Johnson, L. A. and D. C. Montgomery (1974), *Operations Research in Production Planning, Scheduling, and Inventory Control*, John Wiley and Sons, New York.

Kirca, Ö. (1990), An Efficient Algorithm for the Capacitated Single Item Dynamic Lot Size Problem, *European Journal of Operational Research* **45**, 15–24.

Lambrecht M. and J. Vander Eechen (1978), A Capacity Constrainted Single-Facility Dynamic Lot-Size Model, *European Journal of Operational Research* **2**, 132–136.

Leung, J. M. Y., T. L. Magnanti, and R. Vachani (1989), Facets and Algorithms for Capacitated Lot Sizing, *Mathematical Programming* **45**, 331–359.

Love, S. F. (1973), Bounded Production and Inventory Models with Piecewise Concave Costs, *Management Science* **20**, 313–318.

Pochet, Y. and L. A. Wolsey (1991), Solving Multi-Item Lot-Sizing Problems Using Strong Cutting Planes, *Management Science* **37**, 53–67.

Silver, E. A. and R. Peterson (1985), *Decision System for Inventory Management and Production Planning*, John Wiley and Sons, Second Edition.

Sandbothe, R. A. and G. L. Thompson (1990), A Forward Algorithm for the Capacitated Lot Size Model with Stockouts, *Operations Research* **38**, 474–486.

Swoveland, C. (1975), A Deterministic Multi-Period Production Planning Model with Piece-Wise Concave Production and Holding-Backorderer Costs, *Management Science* **21**, 1007–1013.

Veinott, A. F., Jr. (1968), Extreme Points of Leontief Substitution Systems, *Linear Algebra Applications* **1**, 181–194.

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, *Operations Research* **40**, S145–S156.

Wagner, H. M. and T. M. Whitin (1958), Dynamic Version of the Economic Lot Size Model, *Management Science* **5**, 89–96.

Zangwill, W. (1966), A Deterministic Multi-Period Production Scheduling Model with Back-logging, *Management Science* **13**, 105–119.