Fully Polynomial Approximation Schemes for Single–Item Capacitated Economic Lot–Sizing Problems

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

July 1997

Econometric Institute Report 9735/A

Abstract

NP–hard cases of the single–item capacitated lot–sizing problem have been the topic of extensive research and continue to receive considerable attention. However, surprisingly few theoretical results have been published on approximation methods for these problems. To the best of our knowledge, until now no polynomial approximation method is known which produces solutions with a relative deviation from optimality that is bounded by a constant. In this paper we show that such methods do exist, by presenting an even stronger result: the existence of fully polynomial approximation schemes. The approximation scheme is first developed for a quite general model, which has concave backlogging and production cost functions and arbitrary (monotone) holding cost functions. Subsequently we discuss important special cases of the model and extensions of the approximation scheme to even more general models.

Subject classification: Analysis of algorithms, suboptimal algorithms: fully polynomial approximation schemes. Dynamic programming/optimal control: lot–sizing models. Inventory/production: single–item capacitated lot–sizing.

In the single–item capacitated economic lot–sizing problem we consider a production facility which manufactures a single product to satisfy known integer demands over a finite planning horizon of T periods. At each period, the production and holding–backlogging cost functions are given, and the amount of production is subject to a

[*] Department of Quantitative Economics, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; e–mail: `s.vanhoesel@ke.unimaas.nl`

[†] Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands; e–mail: `wagelmans@few.eur.nl`

capacity limit. The problem is that of determining the amounts to be produced in each period such that all demand is satisfied and the total cost is minimized.

Florian, Lenstra and Rinnooy Kan (1980) and Bitran and Yanasse (1982) have shown that the single–item capacitated lot–sizing problem is NP–hard, even for many special cases. For notable exceptions, we refer to Florian and Klein (1971), Bitran and Yanasse (1982), Rosling (1993), Chung and Lin (1988) and Van Hoesel and Wagelmans (1996).

NP–hard cases of the problem have been the topic of extensive research and continue to receive considerable attention. The proposed solution methods are typically based on dynamic programming (for instance, Kirca 1990; Chen, Hearn and Lee 1994a, 1994b; Shaw and Wagelmans 1995), branch–and–bound (for instance, Baker et al. 1978; Erenguc and Aksoy 1990), or a combination of the two (for instance, Chung, Flynn and Lin 1994; Lofti and Yoon 1994).

It should also be mentioned that a lot of research has been devoted to finding a (partial) polyhedral description of the set of feasible solutions of lot–sizing problems; see, for example, Pochet (1988) and Leung et al. (1989), Pochet and Wolsey (1993, 1995) and Constantino (1995). The main motivation for studying the polyhedral structure of capacitated single–item models is to use the results to develop branch–and–cut methods for more complicated problems, such as multi–item problems, that contain this model as a substructure. However, the branch–and–cut approach has not (yet) resulted in competitive algorithms for the capacitated single–item problems themselves.

Surprisingly, very few theoretical results have been published on approximation methods for capacitated single–item problems. The only notable exceptions are Bitran and Matsuo (1986) and Gavish and Johnson (1990). The first article considers approximation formulations which are solvable in pseudo–polynomial time. The optimal solution of an approximation formulation can be used as an approximate solution of the actual problem. For special cases of the problem, it can be shown that the relative error of the approximate solution value can be bounded by an expression which depends on the input data. The authors argue that this bound will be satisfactory for practical purposes. Gavish and Johnson present a fully polynomial approximation scheme which is applicable to a large class of capacitated single–item scheduling problems. Their approach, however, appears to be more suitable for continuous time models, than for discrete times models, such as those considered in this paper. The reason is that in calculating an approximate solution, the discrete nature of the problem is ignored. Therefore, "the translation from this solution back to an equivalent discrete–time model may be difficult" (p. 74). Another drawback of the approach is that the error of the approximate solution is not measured as the usual relative error with respect to the optimal value, but as ratio of the value of the approximate solution and an upperbound on value of any feasible solution. (The ratio of this upperbound and the optimal value may be arbitrarily large.) Gavish and Johnson justify the alternative error measure by pointing out that the usual relative error is inadequate for a minimization problem if there is a possibility that the optimal value is zero. Although this is true in general, we will explain in Section 1 why this is not a relevant argument for the lot–sizing problems considered in this paper.

To summarize the above discussion: to the best of our knowledge, until now no polynomial approximation method is known for the single–item capacitated lot–sizing problem which produces solutions with a relative deviation from optimality that is bounded by a constant. In this paper we will show that such methods do exist, by presenting an even stronger result: the existence of fully polynomial approximations schemes. Recall that such algorithms determine for *any* $\epsilon > 0$ and any problem instance, a solution of which the relative deviation from optimality is at most ϵ, in a running time which is polynomial in both $1/\epsilon$ and the size of the problem instance.

This paper is organized as follows. In Section 1, we first define the model for which the approximation scheme will initially be developed. It assumes concave backlogging and production functions. The holding cost functions are only assumed to be non–decreasing. In Section 2 we present an exact dynamic programming procedure for this model. This algorithm differs from — and is more complicated than — the standard DP approach presented by Florian, Lenstra and Rinnooy Kan (1980). Two approximation methods, one of which is based on the DP algorithm, are described in Section 3, and in Section 4 we show how these methods can be combined to yield a fully polynomial approximation scheme. In Section 5 we discuss two important special cases, namely the model without backlogging (which allows the concavity assumption on the production cost functions to be dropped) and the model in which all cost functions are pseudo–linear (which allows an improved complexity). Furthermore, we will show in this section that our results can be extended to models with features such as bounds on the inventory levels, piecewise concave cost functions and start–up and reservation costs. Section 6 contains concluding remarks.

1 Problem definition

In this section we define the model for which the approximation scheme will initially be developed. Let T denote the length of the planning horizon. For each period $t \in \{1, \ldots, T\}$ we define:

d_t: demand in t;
x_t: production level in t;
c_t: production capacity in period t;

I_t: inventory level at the end of t;
$p_t(x_t)$: production costs in t, a function of x_t;
$h_t(I_t)$: holding–backlogging costs in t, a function of I_t.

Furthermore, I_0 is defined to be 0 and we make the following assumptions:

> All demands, capacities, production and inventory levels are integer.
>
> The production cost function p_t is non–decreasing and concave in the integers of the interval $[0, c_t]$, $t \in \{1, 2, \ldots, T\}$. Furthermore, $p_t(0) = 0$.
>
> The holding–backlogging cost functions are non–decreasing on $[0, \infty)$ and non–increasing and concave on $(-\infty, 0]$. If backlogging is not allowed, then the costs are equal to ∞ for all negative inventory levels. Furthermore, $h_t(0) = 0$ for all $t \in \{1, 2, \ldots, T\}$.
>
> All cost functions can be evaluated in polynomial time at any value in their domain and are scaled such that they are integer valued.

The objective is to satisfy all demand at minimal cost, subject to the capacity constraints. Hence, the problem can be formulated as

$$\begin{array}{rlll} z^* = & \min & \sum_{t=1}^{T}(p_t(x_t) + h_t(I_t)) & \\ & \text{s.t.} & I_t = I_{t-1} + x_t - d_t & t = 1, 2, \ldots, T \\ & & x_t \leq c_t & t = 1, 2, \ldots, T \\ & & I_0 = 0 & \\ & & x_t \geq 0 \quad \text{integer} & t = 1, 2, \ldots, T \\ & & I_t \qquad \text{integer} & t = 1, 2, \ldots, T \end{array}$$

The assumptions $h_t(0) = 0$ and $p_t(0) = 0$ for all periods t, imply that we are only considering the costs which depend on the production plan, i.e., constant costs are ignored. Although adding the same positive constant to the cost of every feasible solution does not change the cost ordering of the solutions, it would decrease the relative error of every solution. Hence, the assumptions can be viewed as a normalization of the problem. With respect to the issue of zero cost solutions, as raised by Gavish and Johnson (1990), we note the following. In Subsection 3.2, it will be shown that under very mild conditions (monotonicity of the cost functions) it is possible to determine in polynomial time whether or not there exists a zero cost solution of a given instance of the single–item capacitated lot–sizing problem. Moreover, if it exists, such a solution is found. Hence, the issue of polynomial approximation is only relevant for those problem instances for which we do not find a zero cost solution. Of course, for these problem instances the relative error with respect to the optimal value is a meaningful measure for the quality of approximate solutions.

In the next section, we will describe an exact solution method for the above problem.

2 A dynamic programming algorithm

In the standard dynamic programming approach to the capacitated dynamic lot sizing problem, one computes (in a forward or backward fashion) for every period $t \in \{1, 2, \ldots, T\}$ and all possible inventory levels in t the minimal cost of achieving that level. The running time of this approach is proportional to $\sum_{t=1}^{T} c_t \times \sum_{t=1}^{T} d_t$ (see Florian, Lenstra and Rinnooy Kan, 1980). It is not easy to base an approximation scheme on this DP approach, since the running time can only be decreased if both cumulative capacity and cumulative demand are rescaled, which means that the set of feasible solutions is changed. As a consequence, it may not be trivial to translate an optimal solution of a rescaled problem instance into a feasible solution of the original instance. It may even be possible that one instance is feasible while the other is infeasible. Therefore, we will present a different, more complicated, dynamic programming approach of which the running time mainly depends on an upperbound on the optimal value z^*. This approach can be viewed as being "dual" to the standard dynamic programming approach. i.e., the ending inventory is maximized subject to a budget constraint.

2.1 Preliminaries

To facilitate the exposition, we will assume from now on that it takes constant time to evaluate any of the cost functions which we defined in the previous section. The reader will have no problem in verifying that every polynomial running time obtained in this paper, will remain polynomial if the function evaluations take polynomial time instead. Furthermore, we will assume from now on that all capacities are strictly positive. The adaption of our algorithms for zero capacities is straightforward.

The following lemmas are well–known, and will be frequently used in our exposition.

Lemma 1 *If two functions f and g are both non–decreasing, then $f + g$ is also non–decreasing.*

Lemma 2 *If two functions f and g are both concave, then $f + g$ is also concave.*

Lemma 3 *If a function f defined on the interval of integers $[a, b]$, is non–decreasing and can be evaluated in constant time, then we can find for any number y the values*

$$\min_{a \le x \le b}\{f(x) \ge y\} \quad and \quad \max_{a \le x \le b}\{f(x) \le y\},$$

and the corresponding values of x, in $\mathcal{O}(\log(b-a))$ time by applying binary search.

Lemma 4 *If a function f defined on the interval of integers $[a, b]$, is concave and can be evaluated in constant time, then we can find for any number y the values*

$$\min_{a \le x \le b}\{f(x) \ge y\},\ \max_{a \le x \le b}\{f(x) \le y\},\ \min_{a \le x \le b}\{f(x) \le y\} \quad and \quad \max_{a \le x \le b}\{f(x) \ge y\}$$

and the corresponding values of x, in $\mathcal{O}(\log(b-a))$ time by binary search.

2.2 The recursion formulas

Let B be any integer upperbound on z^*. For $t \in \{1, 2, \ldots, T\}$ and $b \in \{0, 1, \ldots, B\}$ we define $F_t(b)$ as the maximum value of I_t which can be achieved by production in the first t periods if the total cost incurred in these periods is at most b. Hence, b can be viewed as the total budget that we are allowed to spend in the first t periods. $F_t(b)$ is defined to be $-\infty$ if there does not exist any value of I_t with a corresponding feasible production plan costing at most b. Note that z^* is equal to the smallest value of b for which $F_T(b) \ge 0$.

By definition, the following holds for $t = 1$:

$$F_1(b) = \max_{0 \le x_1 \le c_1}\{x_1 - d_1 \mid p_1(x_1) + h_1(x_1 - d_1) \le b\} \text{ for } b = 0, \ldots, B \qquad (1)$$

For any $b \in \{0, 1, \ldots, B\}$ the value of $F_1(b)$ can be calculated as follows. Define $m_1 = \min\{d_1, c_1\}$. The function $p_1(x_1) + h_1(x_1 - d_1)$ is concave on the interval $[0, m_1]$, and on the interval $[m_1, c_1]$ it is non–decreasing. Therefore, we can find

$$\max\{x_1 \in \{0, 1, \ldots, m_1\} \mid p_1(x_1) + h_1(x_1 - d_1) \le b\} \qquad (2)$$

and

$$\max\{x_1 \in \{m_1, m_1 + 1, \ldots, c_1\} \mid p_1(x_1) + h_1(x_1 - d_1) \le b\}$$

in time $\mathcal{O}(\log(c_1))$, by Lemmas 3 and 4. If both maxima exist, we take the second, i.e., the maximum of the two; if none exists we set $F_1(b) = -\infty$.

We have shown the following.

Proposition 5 *Determining the values of $F_1(b)$ for all $b \in \{0, 1, \ldots, B\}$ can be done in $\mathcal{O}(B \log c_1)$ time.*

Now consider a period $t \in \{2, 3, \ldots, T\}$ and a fixed budget $b \in \{0, 1, \ldots, B\}$. A correct recursion formula which links $F_t(b)$ to the values $F_{t-1}(a)$, $a \in \{0, 1, \ldots, B\}$, is not trivial. Consider a fixed value of a, $1 \leq a \leq b$, and suppose we want to determine the maximum value of I_t such that the total cost incurred in the first $t-1$ periods is at most a and the cost incurred in period t is limited by $b - a$. We first discuss two situations between which we will distinguish.

By definition, with the given budget the maximum ending inventory of the first $t-1$ periods is $I_{t-1} = F_{t-1}(a)$. The remainder $b-a$ of the budget is available for production and inventory costs in period t. The first situation is the one in which it is possible to extend the production plan corresponding to $I_{t-1} = F_{t-1}(a)$, to a plan also including period t, i.e., there exists an $x_t \in \{0, 1, \ldots, c_t\}$ such that

$$p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a$$

We will show that in this case I_{t-1} can be assumed to be $F_{t-1}(a)$. In the second situation we can not extend a plan corresponding to $I_{t-1} = F_{t-1}(a)$, i.e., for all $x_t \in \{0, 1, \ldots, c_t\}$ we have

$$p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) > b - a$$

For this case we can show that x_t may be assumed to be 0. Thus, in both situations, we can restrict the value of one of the variables x_t and I_{t-1}. This is proved in the following two propositions, which are valid even if the backlogging and production cost functions are not concave, but only monotone.

Proposition 6 *If there exists an $x_t \in \{0, 1, \ldots, c_t\}$ such that*

$$p_t(x_t) + h_t(x_t + F_{t-1}(a) - d_t) \leq b - a$$

then only production plans with $I_{t-1} = F_{t-1}(a)$ need to be considered when computing the maximum value of I_t, given budget a for the first $t-1$ periods.

Proof. Let $\hat{x}_t$ be the maximum feasible production level in period t given $I_{t-1} = F_{t-1}(a)$, i.e.,

$$\hat{x}_t = \max\{x_t \in \{0, 1, \ldots, c_t\} \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\} \tag{3}$$

Suppose that $y < F_{t-1}(a)$ and that there also exists a feasible production plan with $I_{t-1} = y$; let $\bar{x}_t$ be the corresponding maximum production level in period t, i.e.,

$$\bar{x}_t = \max\{x_t \in \{0, 1, \ldots, c_t\} \mid p_t(x_t) + h_t(y + x_t - d_t) \leq b - a\}$$

To prove the proposition, it suffices to show that $F_{t-1}(a) + \hat{x}_t - d_t \geq y + \bar{x}_t - d_t$, since this means that taking $I_{t-1} = F_{t-1}(a)$ is always at least as good as $I_{t-1} = y < F_{t-1}(a)$.

Define $\tilde{x}_t = \bar{x}_t - F_{t-1}(a) + y$. Thus, $\tilde{x}_t < \bar{x}_t \leq c_t$, since $y < F_{t-1}(a)$. We will prove that $\hat{x}_t \geq \tilde{x}_t$, which immediately implies $F_{t-1}(a)+\hat{x}_t-d_t \geq F_{t-1}(a)+\tilde{x}_t-d_t = y+\bar{x}_t-d_t$. The proof is by contradiction. Suppose that $\tilde{x}_t > \hat{x}_t$, then $0 \leq \hat{x}_t < \tilde{x}_t < \bar{x}_t \leq c_t$. Furthermore, $\tilde{x}_t$ is a feasible production level for $I_{t-1} = F_{t-1}(a)$. To see this, we note that $p_t(\tilde{x}_t) \leq p_t(\bar{x}_t)$ by the monotonicity of p_t, and $h_t(F_{t-1}(a)+\tilde{x}_t-d_t) = h_t(y+\bar{x}_t-d_t)$, since the arguments are equal. Hence,

$$p_t(\tilde{x}_t) + h_t(F_{t-1}(a) + \tilde{x}_t - d_t) \leq p_t(\bar{x}_t) + h_t(y + \bar{x}_t - d_t) \leq b - a$$

Thus, $x_t = \tilde{x}_t$, and $I_{t-1} = F_{t-1}(a)$ satisfy the budget constraint. We now have a contradiction with the definition of $\hat{x}_t$ in (3). Hence, $\hat{x}_t \geq \tilde{x}_t$, which completes the proof.

□

Proposition 7 *If for all $x_t \in \{0, 1, \ldots, c_t\}$*

$$p_t(x_t) + h_t(x_t + F_{t-1}(a) - d_t) > b - a \tag{4}$$

then only production plans with $I_{t-1} \in \{d_t, d_t + 1, \ldots, F_{t-1}(a) - 1\}$ and $x_t = 0$ need to be considered when computing the maximum value of I_t, given budget a for the first $t - 1$ periods.

Proof. Let $y < F_{t-1}(a)$ and $\bar{x}_t$ be such that taking $I_{t-1} = y$ and production in period t equal to $\bar{x}_t$ is feasible, and $y + \bar{x}_t$ is maximal.

If $y + \bar{x}_t \geq F_{t-1}(a)$, then — because of the monotonicity of p_t — it is also feasible to take $I_{t-1} = F_{t-1}(a)$ and production in period t equal to $\tilde{x}_t = \bar{x}_t - F_{t-1}(a) + y$. Note that $0 \leq \tilde{x}_t < \bar{x}_t \leq c_t$. However, (4) states that such a feasible plan does not exist. Hence, we have a contradiction, which implies $y + \bar{x}_t < F_{t-1}(a)$.

Now assume that $F_{t-1}(a) \leq d_t$. Then, $p_t(\bar{x}_t) + h_t(y + \bar{x}_t - d_t) \geq h_t(y + \bar{x}_t - d_t) \geq h_t(F_{t-1}(a) - d_t) > b - a$, where the second inequality follows from the fact that $h_t(I_t)$ is non–increasing on $(-\infty, 0]$, while the last inequality is (4) for the case $x_t = 0$. Again we have a contradiction. So, besides $y + \bar{x}_t < F_{t-1}(a)$, we may assume $F_{t-1}(a) > d_t$ in the sequel.

Any level of I_{t-1} in the interval $[d_t, F_{t-1}(a)]$ can be attained at total cost at most a in the first $t-1$ periods. To see this, take a production plan for the first $t-1$ periods with $I_{t-1} = F_{t-1}(a)$ and total cost at most a. Change this production plan by lowering the production level in the last production period until the desired value of I_{t-1} is reached or the production level becomes 0. In the latter case, repeat the procedure with the new production plan. Iterate until a production plan with the desired value of I_{t-1} is obtained. This production plan has cost at most a, because in the process of changing the production plan, both the production and holding costs do not increase.

Hence, in particular, we have that $I_{t-1} = d_t$ can be attained at cost at most a. In combination with zero production in period t we get $I_t = 0$. Clearly, this is feasible, because there are no additional costs in period t. Hence, the maximum value of I_t is non–negative, which implies $y + \bar{x}_t \geq d_t$. We now have derived that $d_t \leq y + \bar{x}_t < F_{t-1}(a)$. But this means that also $I_{t-1} = y + \bar{x}_t$ can be attained at cost at most a. In combination with zero production in period t, we get a production plan with total cost in period t equal to $h_t(y + \bar{x}_t - d_t) \leq p_t(\bar{x}_t) + h_t(y + \bar{x}_t - d_t) \leq b - a$. Since this means that the production plan is feasible, we now have shown that it suffices to consider only production plans with $d_t \leq I_{t-1} < F_{t-1}(a)$ and zero production in period t. □

The above two propositions lead to the following recursion formula for $b = 0, \ldots, B$ and $t = 2, \ldots, T$:

$$F_t(b) = \max_{0 \leq a \leq b} \max \left\{ \begin{array}{c} \max\limits_{0 \leq x_t \leq c_t} \{F_{t-1}(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\}, \\ \\ \max\limits_{0 \leq I_t < F_{t-1}(a) - d_t} \{I_t \mid h_t(I_t) \leq b - a\} \end{array} \right\}$$

or, equivalently,

$$F_t(b) = \max \left\{ \begin{array}{c} \max\limits_{0 \leq a \leq b} \max\limits_{0 \leq x_t \leq c_t} \{F_{t-1}(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\}, \\ \\ \max\{I_t \geq 0 \mid \exists\ a \in \{0, 1, \ldots, b\} :\ I_t < F_{t-1}(a) - d_t,\ h_t(I_t) \leq b - a\} \end{array} \right\} \quad (5)$$

Once more, we would like to mention that we have used the monotonicity, but not the concavity of the cost functions to derive the above recursion formula.

2.3 Complexity

Using (5), $F_t(b)$ can be computed from the values $F_{t-1}(a)$, $a \in \{0, 1, \ldots, b\}$, as follows. For the evaluation of the first expression we propose a procedure similar to the procedure for $t = 1$, described at the beginning of the preceding subsection. Consider a fixed value of $a \in \{0, 1, \ldots, b\}$ and define

$$m_t^a = \begin{cases} 0 & \text{if } d_t - F_{t-1}(a) < 0 \\ d_t - F_{t-1}(a) & \text{if } 0 \leq d_t - F_{t-1}(a) \leq c_t \\ c_t & \text{if } d_t - F_{t-1}(a) > c_t \end{cases}$$

Now $p_t(x_t)+h_t(F_{t-1}(a)+x_t-d_t)$ is concave on $[0, m_t^a]$ and non–decreasing on $[m_t^a, c_t]$. Thus, because of Lemmas 3 and 4, the largest achievable value of $I_t = F_{t-1}(a)+x_t-d_t$ can be determined by binary search in $\mathcal{O}(\log c_t)$ time for each $a \in \{0,1,\ldots,b\}$. Hence, the first expression can be evaluated in $\mathcal{O}(b \log c_t)$ time.

For the second expression, we first note that there is no value of I_t satisfying the conditions in this expression if $F_{t-1}(b) \leq d_t$, since $F_{t-1}(a)$ is non–decreasing in a. Let $\bar{a}_b$ be the smallest value of a in $\{0,1,\ldots,b\}$ such that $F_{t-1}(\bar{a}_b) > d_t$ and $h_t(F_{t-1}(\bar{a}_b)-d_t) > b-\bar{a}_b$. The value of $\bar{a}_b$ is non–decreasing in b, since for any a the value $\min\{I_t \mid h_t(I_t) > b-a\}$ is non–decreasing in b. Therefore, the total computational effort for finding $\bar{a}_b$ is $\mathcal{O}(B)$ for all $b \in \{0,1,\ldots,B\}$ simultaneously, i.e., we can compute $\bar{a}_b$ in constant amortized time per $b \in \{0,1,\ldots,B\}$. To evaluate the second expression once $\bar{a}_b$ is known, we note that for each $a \in \{0,1,\ldots,b\}$ there are two functions that bound the inventory I_t, namely $I_t < F_{t-1}(a)-d_t$ and $h_t(I_t) \leq b-a$. Now $F_{t-1}(a)-d_t$ is non–decreasing in a, and $\max\{I_t \mid h_t(I_t) \leq b-a\}$ is non–increasing in a. Therefore, if the value of I_t that we are looking for exists, it belongs to the interval $[F_{t-1}(\bar{a}_b-1)-d_t, F_{t-1}(\bar{a}_b)-d_t)$. To be more precise, it is the largest non–negative value in the interval for which $h_t(I_t) \leq b-\bar{a}_b$. Hence, this value can be determined by binary search in $\mathcal{O}(\log(F_{t-1}(\bar{a}_b))) = O(\log(\sum_{\tau=1}^{t-1} c_\tau))$ time. To summarize, the evaluation of the second expression takes $\mathcal{O}(\log(\sum_{\tau=1}^{t-1} c_\tau))$ amortized time for all $b \in \{0,1,\ldots,B\}$.

We have now derived the main result of this section.

Theorem 8 *The complexity of the dynamic programming algorithm based on formulas (1) and (5) is* $\mathcal{O}(B^2\sum_{t=1}^T \log c_t + B\sum_{t=1}^T \log(\sum_{\tau=1}^{t-1} c_\tau))$.

3 Two approximation algorithms

In this section we discuss two approximation algorithms. The first one is based on the dynamic programming algorithm presented in the preceding section. It yields a feasible solution whose absolute deviation from optimality is bounded, but dependent on T. The second approximation algorithm is quite simple and yields a feasible solution whose relative deviation from optimality is less than $2T$. Both approximation algorithms are part of our approximation scheme to be presented in the next section. The first algorithm forms the basis of the approximation scheme, the second algorithm merely provides an appropriate upperbound B on the optimum value z^*.

3.1 Approximation based on DP algorithm

This approximation algorithm is based on scaling, an idea which is often used in approximation schemes. However, instead of scaling the cost functions, we are going to scale the budgets of the periods. Cost scaling is not a good idea, since it destroys concavity, i.e., functions such as $\lfloor p_t(x_t)/K \rfloor$, where K is a positive integer, are in general not concave.

As before, let B be any integer upperbound on z^*. Furthermore, let K be a positive integer such that $1 \leq K \leq B$. For $t \in \{1, 2, \ldots, T\}$ and $b \in \{0, K, 2K, \ldots, (\lfloor B/K \rfloor + T)K\}$ we define $G_t(b)$ as the maximal value of I_t which can be achieved by production in the first t periods under the restriction that the total budget for these periods is at most b and the budget allocated to each individual period is a multiple of K. From the preceding section it should be clear that we can compute $G_t(b)$ for all $t \in \{1, 2, \ldots, T\}$ and all $b \in \{0, K, 2K, \ldots, (\lfloor B/K \rfloor + T)K\}$ in a total computational effort which is $\mathcal{O}((B/K + T)^2 \sum_{\tau=1}^{T} \log c_\tau \; + \; (B/K + T) \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau))$. The idea is to take the smallest value of $b \in \{0, K, 2K, \ldots, (\lfloor B/K \rfloor + T)K\}$ for which $G_T(b) \geq 0$ as the value of the approximate solution of the lot sizing problem. We will show the existence of such a solution and give a bound on the absolute difference between the value of the approximate solution and the optimal value in the following proposition.

Proposition 9 *There exists a $b \in \{0, K, 2K, \ldots, (\lfloor B/K \rfloor + T)K\}$ with $G_T(b) \geq 0$. Moreover, the smallest such value is less than or equal to $z^* + TK$.*

Proof. Consider an optimal solution and let r_t denote the associated cost incurred in period t, $t \in \{1, 2, \ldots, T\}$. Clearly, the solution is feasible if we would allocate a budget of $(\lfloor r_t/K \rfloor + 1)K$ to each period $t \in \{1, 2, \ldots, T\}$. Because these budgets are multiples of K, this implies that $G_T(\sum_{t=1}^{T}(\lfloor r_t/K \rfloor K + K)) \geq 0$. The proposition now follows from $\sum_{t=1}^{T}(\lfloor r_t/K \rfloor K + K) \leq \lfloor \sum_{t=1}^{T} r_t/K \rfloor K + TK) \leq (\lfloor z^*/K \rfloor + T)K$ and the fact that the last expression is bounded from above by both $(\lfloor B/K \rfloor + T)K$ and $z^* + TK$.

□

3.2 A simple polynomial approximation algorithm

We will now show how to compute an upperbound on z^* which is at most $2Tz^*$. This approximation algorithm is quite simple and it can also be used if the cost functions are not concave, but only monotone. It is based on the fact that there are $2T$ different cost functions. The idea of the algorithm is to find the smallest value L for which there exists a feasible solution if all cost functions are restricted to contribute at most L to the total cost. Hence, such a feasible solution has cost at most $2TL$. Clearly, in any optimal solution of the original problem, each cost function contributes not more than z^*. Therefore, it holds that $L \leq z^*$. This implies that $\bar{B} \equiv 2TL$ is an upperbound on z^* such that $\bar{B} \leq 2Tz^*$.

To show that L can be found in polynomial time, we first show that it is possible to determine in polynomial time whether or not there exists a feasible solution if the contribution of each cost function is at most some given value l. For each period t we define an upperbound on the production level by $\bar{c}_t = \max\{x \leq c_t \mid p_t(x) \leq l\}$, and a lower and upperbound on the inventory level by $u_t = \min\{I \leq 0 \mid h_t(I) \leq l\}$ and $v_t = \max\{I \geq 0 \mid h_t(I) \leq l\}$, respectively. These bounds can be determined using binary search.

A feasible solution in which each cost function contributes at most l exists if and only if there exists a feasible solution which satisfies the above upper and lowerbounds on the production and inventory levels. We can use dynamic programming to check this. Let M_t denote the largest value of I_t, achievable by production in the first t periods by a production plan satisfying all upper and lowerbounds. In particular, we have $M_1 = \min\{\bar{c}_1 - d_1, v_1\}$. If $M_1 < u_1$, then there does not exist a feasible solution. Otherwise, we proceed using the recursion formula

$$M_t = \min\{M_{t-1} + \bar{c}_t - d_t, v_t\} \quad \text{for } t = 2, \ldots, T$$

and we stop as soon as we find a t for which $M_t < u_t$. There exists a feasible solution if and only if we reach T and $M_T \geq 0$.

Clearly, L is non–negative and a trivial upperbound on L is given by

$$U \equiv \max_{1 \leq t \leq T} \{(p_t(c_t), h_t(-\sum_{i=1}^{t-1} d_i), h_t(\sum_{i=t+1}^{T} d_i)\}$$

(or $\max_{1 \leq t \leq T}\{(p_t(c_t), h_t(\sum_{i=t+1}^{T} d_i)\}$ if backlogging is not allowed). Now it should be clear how L can be determined using binary search. Note, however, that the value of any feasible solution is also an upperbound on L. Suppose such a value, say $\tilde{B}$, is known (for instance, $\tilde{B}$ could be the value of any heuristic solution), then we can do the following. We first check whether there exists a feasible solution in which each cost function contributes at most $\lceil \tilde{B}/2T \rceil$. If this is not the case, then $\lceil \tilde{B}/2T \rceil < L \leq z^*$. Hence, $\tilde{B} < 2Tz^*$, and we are done. Otherwise, we carry out the binary search for L on $[0, \tilde{B}]$.

The running time of the above heuristic is easily seen to be $\mathcal{O}(T \log^2 U)$. Note that this heuristic can also be used to check in polynomial time whether there exists an optimal solution with zero cost.

4 The fully polynomial approximation scheme

We will first describe a straightforward version of our approximation scheme, and then discuss possible ways to improve its complexity.

4.1 Description and correctness

Our fully polynomial approximation scheme consists of two steps and combines the approximation algorithms discussed in the preceding section. Let $\epsilon > 0$ be given.

1. Use the simple approximation algorithm to calculate in polynomial time an upperbound B which satisfies $B \leq 2Tz^*$.
2. Apply the DP based approximation algorithm; use the calculated B as the upperbound and $K = \max\{\lfloor \epsilon B/2T^2 \rfloor, 1\}$.

We now state the main result of this paper.

Theorem 10 *The above procedure has a complexity which is polynomial in both the size of the problem instance and $1/\epsilon$, and determines a feasible solution with a value not larger than $(1+\epsilon)z^*$.*

Proof. For the first part of the proposition, we only have to analyze the complexity of Step 2. As already mentioned in Subsection 3.1, its running time is $\mathcal{O}((B/K + T)^2 \sum_{\tau=1}^{T} \log c_\tau + (B/K+T) \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau))$. Clearly, this is a polynomial bound if $B/K \leq T$. Therefore, let us assume $B/K > T$. If $\epsilon B/2T^2 > 1$, then $K > \epsilon B/4T^2$; otherwise, $K \geq \epsilon B/2T^2$. In both cases, it is easily verified that the running time is $\mathcal{O}(T^4 \sum_{\tau=1}^{T} \log c_\tau/\epsilon^2 + T^2 \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)/\epsilon)$, which is polynomial in the size of the problem instance and $1/\epsilon$.

If $K = 1$, then a solution with value z^* is found in Step 2. If $K = \lfloor \epsilon B/2T^2 \rfloor$, we can use the fact that this step yields a solution whose value exceeds z^* by at most KT, which is less than or equal to $\epsilon B/2T \leq \epsilon z^*$. This completes the proof. □

4.2 Complexity

In the proof of Theorem 10, we mentioned the complexity bound $\mathcal{O}(T^4 \sum_{\tau=1}^{T} \log c_\tau/\epsilon^2 + T^2 \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)/\epsilon)$ for Step 2 of the approximation scheme. There are several ways to improve this bound. An obvious approach is to apply the DP based approximation algorithm not once, but twice. First it is applied with $K = \max\{\lfloor \hat{\epsilon} B/2T^2 \rfloor, 1\}$, where $\hat{\epsilon}$ is a relatively large error. This yields an upperbound, say $\hat{B}$. Subsequently, the approximation algorithm is applied with $K = \max\{\lfloor \hat{\epsilon}\hat{B}/T(1+\hat{\epsilon})\rfloor, 1\}$, yielding a solution with the required quality guarantee. A good choice for $\hat{\epsilon}$ is one for which the complexity of the first and second execution of the approximation algorithm is about the same. For instance, if we take $\hat{\epsilon} = \sqrt{T\epsilon}$, the overall complexity, including Step 1, is $\mathcal{O}(T\log^2 U \ + \ (T + T\sqrt{T\epsilon})^2 \sum_{\tau=1}^{T} \log c_\tau/\epsilon^2 \ + \ (T + T\sqrt{T\epsilon}) \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)/\epsilon)$.

Another way to improve the complexity is due to Kovalyov (1995), to whom we refer for details. Given the lowerbound L, the upperbound B and the fact that $B/L \leq 2T$, it can be shown that a lowerbound $\hat{L}$ and an upperbound $\hat{B}$ with $\hat{B}/\hat{L} \leq 3$ can be found in $\mathcal{O}(\log T(T^2 \sum_{\tau=1}^{T} \log c_\tau \ + \ T \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)))$ time. The idea is to iteratively apply the DP based approximation algorithm with $K = \max\{\lfloor L'/T \rfloor, 1\}$, starting with $L' = L$. If the approximation algorithm does not find a feasible solution, the value of L' is doubled and the algorithm is repeated. When a feasible solution is found, the procedure terminates. $\hat{B}$ is equal to the value of the feasible solution and $\hat{L}$ is equal to the current value of L'. Since $\hat{B} \leq 3z^*$, we can subsequently apply the DP based approximation algorithm with $K = \max\{\lfloor \epsilon\hat{B}/3T \rfloor, 1\}$ to obtain a solution with the desired accuracy. The overall complexity of this approach is $\mathcal{O}(T\log^2 U \ + \ \log T(T^2 \sum_{\tau=1}^{T} \log c_\tau \ + \ T \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)) \ + \ T^2 \sum_{\tau=1}^{T} \log c_\tau/\epsilon^2 \ + \ T \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau)/\epsilon)$.

Further improvements of the complexity may be achieved for certain special cases of the cost functions, as discussed in the next section.

5 Special cases and extensions

The model for which we have developed the approximation scheme in the preceding sections, is quite general. On one hand, stronger results can be obtained for interesting special cases. On the other hand, our results can be extended to even more general capacitated lot sizing problems encountered in the literature.

5.1 No backlogging

In our exposition, we have only used the concavity of the production cost functions to evaluate (2) in Subsection 2.2 and the first expression in (5) in Subsection 2.3 efficiently. To be more precise, the assumption is used to deal efficiently with the possibility of backlogging. Hence, in case backlogging is not allowed, it is not necessary to assume that the production cost functions are concave. Therefore, we have the following result.

Theorem 11 *If backlogging is not allowed, the approximation scheme is still correct if the production cost functions are only non–decreasing and not concave.*

5.2 Pseudo–linear cost functions

An important special case is the one in which all cost functions are pseudo–linear, as is often assumed in the literature (see, for instance, Baker et al. 1978; Lambrecht and Vander Eecken 1978; Bitran and Yanasse 1982; Chung and Lin 1988; Chung, Flynn and Lin 1994; Chen, Hearn and Lee 1994a). In the appendix we show that in this case the dynamic programming algorithm can be adapted to run in $\mathcal{O}(TB)$ time. Also, the simple polynomial heuristic of Section 3.2 runs in $\mathcal{O}(T \log U)$ time, because each of the bounds $\bar{c}_t$, u_t and v_t can now each be calculated analytically in constant time. Hence, a straightforward version of the approximation scheme runs in $\mathcal{O}(T \log U + T^3/\epsilon)$ time. Using Kovalyov's complexity improvement idea, we obtain the following result.

Theorem 12 *If all cost functions are pseudo–linear, then the fully polynomial approximation scheme runs in $\mathcal{O}(T \log U + T^2 \log T + T^2/\epsilon)$ time.*

5.3 Piecewise concave or convex cost functions

Love (1973) and Swoveland (1975) consider the problem in which the cost functions are piecewise concave (see also Chen, Hearn and Lee 1994b). Let us first discuss how our approximation scheme should be adapted if the backlogging cost functions are piecewise concave (and non–increasing) instead of simply concave. Our DP algorithm is only affected with respect to the evaluation of (2) and the first expression in (5), since these are the only steps in the algorithm where concavity is used. If the backlogging cost function of period 1 consists of n_1 concave pieces, it is easily seen that evaluating (2) can be done by performing at most n_1 binary searches, instead of just one. The evaluation of the first expression in (5) can be adapted in a similar way. Hence, if each backlogging function consists of at most n concave pieces, then the complexity of the dynamic programming algorithm, as given in Theorem 8, is increased by at most a factor n. The following result is now obvious.

Theorem 13 *If the backlogging cost functions are piecewise concave and the number of pieces is polynomially bounded in the size of the problem instance, then there exists a fully polynomial approximation scheme.*

Also note that lower and upperbounds on the inventory levels can easily be incorporated in our approximation scheme, since these bounds can be modeled by defining the holding–backlogging costs to be infinite outside the feasible range.

Now suppose that the production cost functions are piecewise concave and monotone. Again we only have to discuss how this affects the evaluation of (2) and the

first expression in (5). Let us consider the latter. If $p_t(x_t)$ is concave on some interval $[x^l, x^u] \subset [0, c_t]$, then for any a, $0 \leq a \leq b$, we have
$\max_{x^l \leq x_t \leq x^u}\{F_{t-1}(a) + x_t - fd_t \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\} =$
$\max_{0 \leq \Delta x_t \leq x^u - x^l}\{F_{t-1}(a)+x^l+\Delta x_t - d_t \mid p_t(x^l+\Delta x_t)+h_t(F_{t-1}(a)+x^l+\Delta x_t - d_t) \leq b-a\}$.

It is obvious that the value of Δx_t which maximizes this expression can again be found by binary search. Hence, to evaluate the first expression of (5), it suffices to perform a number of binary searches which is at most the number of concave pieces of $p_t(x_t)$. A similar remark holds for the evaluation of (2). This implies the following result.

Theorem 14 *If the production cost functions are piecewise concave and the number of pieces is polynomially bounded in the size of the problem instance, then there exists a fully polynomial approximation scheme.*

Finally, it is worth mentioning that Veinott (1964) and Erenguc and Aksoy (1990) consider models in which the cost functions are (piecewise) convex instead of concave. We just note that if both the backlogging and production cost functions are piecewise convex (and monotone), our fully polynomial approximation scheme can be applied, since we can still use binary search to evaluate (2) and the first expression in (5) efficiently.

5.4 Start–up and reservations costs

Karmarkar, Kekre and Kekre (1987) have introduced the dynamic lot–sizing problem with start–up and reservation costs. In this model a start–up cost S_t is incurred if the production facility is switched on in period t, and a separate reservation cost R_t is charged for keeping the facility on whether or not it is used for production. These costs are incurred in addition to the the production cost $p_t(x_t)$. To handle this cost structure, the DP algorithm should be modified. For $t = 1, 2, \ldots, T$ and $b = 0, 1, \ldots, B$, we define $F_t(b)$ as before. Furthermore, $F_t^0(b)$ is defined as the maximum value of I_t which can be achieved by production in the first t periods if the total cost is at most b and the production facility is off in period t. Finally, we define $F_t^1(b)$ as the maximum value of I_t achievable in the first t periods if the total cost is at most b and the production facility is on in period t. Hence, $F_t(b) = \max\{F_t^0(b), F_t^1(b)\}$.

Let us assume that there is no production in period 0. Then, we have, for $b = 0, 1, \ldots, B$,

$$F_1^0(b) = \begin{cases} -d_1 & \text{if } \ h_1(-d_1) \leq b \\ -\infty & \text{otherwise} \end{cases}$$

$$F_1^1(b) = \max_{0 \le x_1 \le c_1} \{x_1 - d_1 \mid S_1 + R_1 + p_1(x_1) + h_1(x_1 - d_1) \le b\}$$

The latter formula can be evaluated analogously to (2). Let us now consider the recursion formulas for $t \ge 2$. The formula for $F_t^0(b)$, i.e., $x_t = 0$, is trivial:

$$F_t^0(b) = \max\{I_t \ge 0 \mid \exists\ a \in \{0, 1, \ldots, b\} : \ I_t \le F_{t-1}(a) - d_t,\ h_t(I_t) \le b - a\}$$

This recursion formula can be evaluated in a similar way as the second expression in (5). Furthermore, we have

$$F_t^1(b) = \max_{0 \le a \le b} \max$$

$$\left\{ \begin{array}{l} \max\limits_{0 \le x_t \le c_t} \{F_{t-1}^0(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}^0(a) + x_t - d_t) \le b - a - S_t - R_t\}, \\ \\ \max\{I_t \ge 0 \mid \exists\ a \in \{0, 1, \ldots, b\} : \ I_t < F_{t-1}^0(a) - d_t,\ h_t(I_t) \le b - a - S_t - R_t\}, \\ \\ \max\limits_{0 \le x_t \le c_t} \{F_{t-1}^1(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}^1(a) + x_t - d_t) \le b - a - R_t\}, \\ \\ \max\{I_t \ge 0 \mid \exists\ a \in \{0, 1, \ldots, b\} : \ I_t < F_{t-1}^1(a) - d_t,\ h_t(I_t) \le b - a - R_t\} \end{array} \right\} (6)$$

Of course, this recursion formula resembles (5). Its correctness is based on properties similar to those stated in Propositions 6 and 7, which can be proven analogously. The only difference is that we have to distinguish between the two possible states of the production facility in period $t - 1$. Efficient evaluation of (6) can be done analogously to the evaluation of (5). It follows that the model with start–up and reservation costs can be solved by a dynamic programming algorithm based on the above formulas with complexity $\mathcal{O}(B^2 \sum_{t=1}^T \log c_t + B \sum_{t=1}^T \log(\sum_{\tau=1}^{t-1} c_\tau))$. Because the simple polynomial approximation algorithm described in Subsection 3.2 can trivially be adapted to incorporate start–up and reservation costs (distinguish again between the two possible states in every period and define corresponding variables and parameters), we have the following result.

Theorem 15 *If there are start–up and reservation costs in addition to the usual production costs, then there exists a fully polynomial approximation scheme.*

6 Concluding remarks

We have developed the first fully polynomial approximation schemes for single–item capacitated lot–sizing problems, where the error is measured in the usual way, i.e., as the relative deviation form optimality. To the best of our knowledge, even polynomial approximation methods which produce solutions with a relative error bounded by a constant were previously unknown. We have shown that our approach is applicable to many single–item capacitated lot–sizing models encountered in the literature.

The most important idea in our the approximation schemes is the non–trivial "dual" DP formulation in combination with budget scaling. A similar approach may result in approximations schemes for problems which are closely related to single–item capacitated lot–sizing problems, such as certain NP–hard location and network design problems on trees (see, for instance, Flippo et al. 1996) and NP–hard variants of the discrete lot–sizing and scheduling problem (Salomon et al. 1991). It is unlikely, however, that our results can be extended to fairly general multi–item capacitated economic lot–sizing problems, since these are known to be strongly NP–hard (Chen and Thizy 1990).

Appendix: Pseudo-linear cost functions

In this appendix, the cost functions are assumed to be of the following form for $t = 1, \dots, T$:

$$p_t(x_t) = \begin{cases} 0 & \text{if } \; x_t = 0 \\ f_t + r_t x_t & \text{if } \; 0 < x_t \leq c_t \end{cases}$$

$$h_t(I_t) = \begin{cases} 0 & \text{if } \; I_t = 0 \\ e_t + s_t I_t & \text{if } \; I_t > 0 \\ g_t - q_t I_t & \text{if } \; I_t < 0 \end{cases}$$

where f_t, r_t, e_t, s_t, g_t and q_t are non–negative integers. We will show that in this case the complexity of the dynamic programming algorithm can be reduced. Consider the following expression, which is part of recursion (5).

$$\max_{0 \leq a \leq b} \max_{0 \leq x_t \leq c_t} \{F_{t-1}(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\} \qquad (7)$$

As before, we would like to evaluate this expression for every $b \in \{0, 1, \dots, B\}$. To do this efficiently, we will no longer consider these expressions for each value of b separately, but we will exploit the fact that for consecutive values of b the expressions are closely related. Our main result will be an $\mathcal{O}(TB)$ bound on the total computational effort to evaluate (7) for all $b \in \{0, 1, \dots, B\}$ and all $t \in \{1, 2, \dots, T\}$, instead of the

$\mathcal{O}(B^2 \sum_{t=1}^{T} \log c_t + B \sum_{t=1}^{T} \log(\sum_{\tau=1}^{t-1} c_\tau))$ bound, which was proved for the general case in Section 2.

To start the exposition, we rewrite (7) in terms of I_t, which results in the following maximization problem.

$$\begin{aligned} \max\nolimits_{0 \le a \le b} \{ \; I_t \mid F_{t-1}(a) - d_t \le I_t \le F_{t-1}(a) + c_t - d_t \; , \\ a + p_t(I_t - F_{t-1}(a) + d_t) + h_t(I_t) \le b \} \end{aligned}$$

We split the maximization problem above into four (possibly overlapping) subproblems corresponding to the following cases: (I) $I_t = 0$, (II) no production, i.e., $I_t = F_{t-1}(a) - d_t$, (III) $I_t > 0$ and positive production, and (IV) $I_t < 0$ and positive production. We will solve these subproblems independently of each other. However, each subproblem is considered for all $b \in \{0, 1, \ldots, B\}$ simultaneously. We will show that the total computational effort to solve a subproblem for all $b \in \{0, 1, \ldots, B\}$ together is $\mathcal{O}(B)$.

Subproblem (I)

Since the value of I_t is fixed, this is essentially a feasibility problem. If the feasible region is non–empty for a certain value of b, then it is also feasible for larger values of b. Hence, the problem boils down to finding the smallest value of b for which the feasible region is non–empty. This is done by considering b in order of increasing value and keeping track of

$$\min_{0 \le a \le b} \{ \; a + p_t(-F_{t-1}(a) + d_t) \mid F_{t-1}(a) - d_t \le 0 \le F_{t-1}(a) + c_t - d_t \} \tag{8}$$

As soon as (8) is smaller than b, we have found the smallest value for which the feasible region is non–empty. Otherwise, we proceed with the next value of b. Since (8) can be updated in constant time when the value of b is increased by 1, it follows that it takes in total $\mathcal{O}(B)$ time to solve subproblem (I) for all $b \in \{0, 1, \ldots, B\}$.

Subproblem (II)

The problems are of the following form.

$$\max_{0 \le a \le b} \{ \; F_{t-1}(a) - d_t \mid a + h_t(F_{t-1}(a) - d_t) \le b \} \tag{9}$$

To solve these subproblems efficiently, we consider them in order of increasing value of b. We first determine a_B, which is defined as the largest a for which $a + h_t(F_{t-1}(a) - d_t) \le B$. Since $F_{t-1}(a)$ is non–decreasing in a, the optimal value of (9) for $b = B$ is $F_{t-1}(a_B) - d_t$. Next we determine the largest a for which $a + h_t(F_{t-1}(a) - d_t) \le B - 1$. Clearly, we can do this by considering a in decreasing order, starting from a_B until we reach the desired value. This gives us the optimal value of (9) for $b = B - 1$, and so on. The total computational effort of this procedure is easily seen to be $\mathcal{O}(B)$.

Subproblem (III)

We now consider the case in which both x_t and I_t are positive. Substituting the specific cost functions, the corresponding problems can be written as

$$\begin{aligned}\max_{0\le a\le b}\max\{\ I_t \mid \max\{1, F_{t-1}(a)+1-d_t\} \le I_t \le F_{t-1}(a)+c_t-d_t\ ,\\ f_t+e_t+r_td_t+(r_t+s_t)I_t - r_tF_{t-1}(a)+a\le b\}\end{aligned} \tag{10}$$

Let a^l be the smallest value of a with $F_{t-1}(a)+c_t-d_t\ge 1$. Clearly, values of $a<a^l$ can be ignored. If $r_t+s_t=0$, then it is optimal to take $I_t=F_{t-1}(a)+c_t-d_t$ for all $a\ge a^l$. In this case we can use a similar approach as for Subproblem (II). Therefore, we assume $r_t+s_t>0$ from now on.

Consider for any $a\in\{a^l, a^l+1,\ldots,B\}$ the maximization problem

$$\begin{aligned}\max\{\ I_t \mid \max\{1, F_{t-1}(a)+1-d_t\} \le I_t \le F_{t-1}(a)+c_t-d_t\ ,\\ f_t+e_t+r_td_t+(r_t+s_t)I_t - r_tF_{t-1}(a)+a\le b\}\end{aligned} \tag{11}$$

Of course, the optimal value of this problem depends on the value of b. In particular, the feasible region of the maximization problem is empty if b is less than $b_l(a)\equiv f_t+e_t+r_td_t+(r_t+s_t)\max\{1,F_{t-1}(a)+1-d_t\}-r_tF_{t-1}(a)+a$. On the other hand, if b is larger than $b_u(a)\equiv f_t+e_t+(r_t+s_t)c_t-s_td_t+s_tF_{t-1}(a)+a$, then the constraint involving b is redundant and it is optimal to take I_t equal to its simple upperbound. For values of b from $b_l(a)$ to $b_u(a)$, the constraint involving b is binding. Hence, for each value of a, we have the following optimal solution of (11):

$$I_t=\begin{cases}-\infty & \text{if}\ \ b<b_l(a)\\[2ex] H^1(a,b)\equiv\lfloor\frac{1}{r_t+s_t}(b-f_t-e_t-r_td_t+r_tF_{t-1}(a)-a)\rfloor & \text{if}\ \ b_l(a)\le b\le b_u(a)\\[2ex] H^2(a)\equiv F_{t-1}(a)+c_t-d_t & \text{if}\ \ b\ge b_u(a)+1\end{cases}$$

For any value of $b\in\{a^l,a^l+1,\ldots,B\}$, we can now rewrite (10) as

$$\max\left\{\begin{array}{l}\max\{\ H^1(a,b)\mid a^l\le a\le b,\ b_l(a)\le b\le b_u(a)\},\\ \max\{\ H^2(a)\mid a^l\le a\le b,\ b_u(a)+1\le b\}\end{array}\right\}$$

Our approach will be to determine the values $\max\{\ H^1(a,b)\mid a^l\le a\le b,\ b_l(a)\le b\le b_u(a)\}$ for all $b\in\{a^l,a^l+1,\ldots,B\}$, and — independently — the values $\max\{\ H^2(a)\mid a^l\le a\le b,\ b_u(a)+1\le b\}$, $b\in\{a^l,a^l+1,\ldots,B\}$. In order to do this efficiently, we will use the facts stated in the following three propositions.

Proposition 16 *The value $b_u(a)$ is strictly increasing in a.*

Proof. The statement follows immediately from the definition of $b_u(a)$ and the fact that $F_{t-1}(a)$ is non–decreasing in a.

□

Proposition 17 *Suppose that for some $\bar{a} \in \{a^l, a^l+1, \ldots, B-1\}$ it holds that $b_l(\bar{a}) \geq b_l(\bar{a}+1)$, then $\max\{\ H^1(a,b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\} = \max\{\ H^1(a,b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a),\ a \neq \bar{a}\}$.*

Proof. Because of Proposition 16, we have that $[b_l(\bar{a}), b_u(\bar{a})] \subset [b_l(\bar{a}+1), b_u(\bar{a}+1)]$. Therefore, it suffices to show that $H^1(\bar{a}+1, b) \geq H^1(\bar{a}, b)$ for all $b \in [b_l(\bar{a}), b_u(\bar{a})]$.

The inequality $b_l(\bar{a}) \geq b_l(\bar{a}+1)$ immediately implies

$$\begin{aligned}&\max\{1, F_{t-1}(\bar{a}) + 1 - d_t\} - r_t F_{t-1}(\bar{a}) + \bar{a} \geq \\ &\max\{1, F_{t-1}(\bar{a}+1) + 1 - d_t\} - r_t F_{t-1}(\bar{a}+1) + \bar{a} + 1\end{aligned}$$

or, equivalently,

$$\begin{aligned}&r_t F_{t-1}(\bar{a}+1) - r_t F_{t-1}(\bar{a}) - 1 \geq \\ &\max\{1, F_{t-1}(\bar{a}+1) + 1 - d_t\} - \max\{1, F_{t-1}(\bar{a}) + 1 - d_t\} \geq 0\end{aligned}$$

Since $H^1(\bar{a}+1, b) - H^1(\bar{a}, b) = \frac{1}{r_t+s_t}(r_t F_{t-1}(\bar{a}+1) - r_t F_{t-1}(\bar{a}) - 1)$ for any $b \in [b_l(\bar{a}), b_u(\bar{a})]$, the desired result now follows.

□

Proposition 18 *For all $a \in \{a^l, a^l+1, \ldots, B\}$ it holds that $b_l(a) \geq a$.*

Proof.

$$\begin{aligned}b_l(a) &= f_t + e_t + r_t d_t + (r_t + s_t)\max\{1, F_{t-1}(a) + 1 - d_t\} - r_t F_{t-1}(a) + a \\ &\geq f_t + e_t + r_t d_t + r_t(F_{t-1}(a) + 1 - d_t) + s_t - r_t F_{t-1}(a) + a \\ &= f_t + e_t + r_t + s_t + a \\ &\geq a\end{aligned}$$

□

Theorem 19 *The values $\max\{\ H^1(a,b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\}$ and $\max\{\ H^2(a) \mid a^l \leq a \leq b,\ b_u(a) + 1 \leq b\}$ can be computed for $b = a^l, a^l+1, \ldots, B$ in a total computational effort which is $\mathcal{O}(B)$.*

Proof. Let us first focus on the computation of the values $\max\{\ H^2(a) \mid a^l \leq a \leq b,\ b_u(a) + 1 \leq b\}$, $b = a^l, a^l + 1, \ldots, B$. Let a^u be the largest value of a for which $b_u(a) + 1 \leq B$. From $b_u(a^u) \geq b_l(a^u)$ and Proposition 18 it follows that $b > a^u$ if $b \geq b_u(a) + 1$. Because $H^2(a) = F_{t-1}(a) + c_t - d_t$ is non–decreasing in a, we can now conclude $\max\{\ H^2(a) \mid a^l \leq a \leq b,\ b_u(a) + 1 \leq b\} = H^2(a^u)$ for $b_u(a^u) + 1 \leq b \leq B$. Analogously, we can prove $\max\{\ H^2(a) \mid a^l \leq a \leq b,\ b_u(a) + 1 \leq b\} = H^2(a^u - 1)$ for $b_u(a^u - 1) + 1 \leq b \leq b_u(a^u)$, and so on. Hence, the procedure boils down to determining for all $b \in \{a^l, a^l + 1, \ldots, B\}$ the largest value $b_u(a) + 1$ which is less than or equal to b. This can easily be done in $\mathcal{O}(B)$ time.

Let us now consider the computation of $\max\{\ H^1(a, b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\}$, $b = a^l, a^l + 1, \ldots, B$. Because of Proposition 17 any value $\bar{a}$ for which $b_l(\bar{a}) \geq b_l(\bar{a} + 1)$ may be ignored while determining these maxima. This implies that it suffices to consider the subsequence A of $a^l, a^l + 1, \ldots, B$ defined by the property that $a \in A$ if and only if there does not exist any $a' \in \{a + 1, a + 2, \ldots, B\}$ with $b_l(a') \leq b_l(a)$. Note that this means that both $b^l(a)$ and $b^u(a)$ are strictly increasing for increasing $a \in A$. Also note that A can be constructed in $\mathcal{O}(B)$ time.

Now define for every $b \in \{a^l, a^l + 1, \ldots, B\}$ the — possibly empty — subset $S(b)$ of elements of A as follows. If $S(b) = \{a_1, a_2, \ldots, a_m\}$ then

1. a_1 is the smallest $a \in A$ for which $b_l(a) \leq b \leq b_u(a)$,
2. a_i, $i = 2, 3, \ldots, m$, is the smallest $a \in A$ for which $a_i > a_{i-1}$, $b_l(a_i) \leq b \leq b_u(a_i)$ and $H^1(a_i, b) > H^1(a_{i-1}, b)$.

If $S(b)$ is empty, then clearly $\max\{\ H^1(a, b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\} = -\infty$. If $S(b)$ is non–empty, then we have the properties $a_1 < a_2 < \ldots < a_m$ and $H^1(a_1, b) < H^1(a_2, b) < \ldots < H^1(a_m, b)$. Because of Proposition 18, we know that $a_i \leq b$ for all $i = 1, 2, \ldots, m$. It is now easily verified that $H^1(a_m, b) = \max\{\ H^1(a, b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\}$.

Besides the fact that we immediately obtain the value $\max\{\ H^1(a, b) \mid a^l \leq a \leq b,\ b_l(a) \leq b \leq b_u(a)\}$, there is another reason for keeping track of $S(b)$. If for a certain value of b a value $a_0 \in A$ with $b_l(a_0) \leq b \leq b_u(a_0)$ is not in $S(b)$, then it is not in $S(b')$ for any $b' < b$ with $b_l(a_0) \leq b' \leq b_u(a_0)$. This follows from the fact that there exists an $a_i \in A$ with $a_i < a_0$, $b_l(a_i) \leq b \leq b_u(a_i)$ and $H^1(a_i, b) \geq H^1(a_0, b)$. Because $b_l(a_i) < b_l(a_0) \leq b' < b \leq b_u(a_i)$, it holds that $b_l(a_i) \leq b' \leq b_u(a_i)$. Moreover, $H^1(a_i, b) \geq H^1(a_0, b)$ implies $H^1(a_i, b') \geq H^1(a_0, b')$. Hence, a_0 is not in $S(b')$.

We will consider b in order of decreasing value. The elements of subset $S(B)$ can trivially be found in $\mathcal{O}(B)$ time. To achieve this complexity bound for all $b \in \{a^l, a^l + 1, \ldots, B\}$ together, we represent the subsets by a list in which the elements are stored in increasing order. This list has the property that at the bottom elements can only be deleted, while at the top elements may be deleted and added. It is well–known that this data structure can be implemented such that each deletion and each addition requires constant time (see, for instance, Aho, Hopcroft and Ullman 1983).

Now suppose that $S(b)$ has been determined for a certain value of $b \in \{a^l+1, a^l+2, \ldots B\}$. Let a_0 be the largest element of A with $b_u(a_0) < b$. In order to determine $S(b-1)$, we do the following. If $S(b)$ is empty, then we check whether $b-1 = b_u(a_0)$. If this is the case, then $S(b-1) = \{a_0\}$, otherwise $S(b-1) = \emptyset$. If $S(b)$ is non–empty, say $S(b) = \{a_1, a_2, \ldots, a_m\}$, then the following steps are carried out.

(i) If $b-1 < b_l(a_m)$, then delete a_m from the list.

(ii) If $b-1 = b_u(a_0)$, then delete from the top of the list all a_i for which $H^1(a_i, b-1) \leq H^1(a_0, b-1)$ and add a_0 to the top of the list.

The total amount of work involved in carrying out these steps for $b = a^l+1, a^l+2, \ldots, B$ can be bounded by a constant times the total number of additions to and deletions from the list. Since every $b \in \{a^l, a^l+1, \ldots, B-1\}$ is added to the list exactly once and deleted at most once, the $\mathcal{O}(B)$ bound now follows. This completes the proof.

□

Subproblem (IV)

The subproblems are now of the following form.

$$\max_{0\leq a\leq b} \max\{\ I_t \mid F_{t-1}(a)+1-d_t \leq I_t \leq \min\{-1, F_{t-1}(a)+c_t-d_t\}\ ,$$
$$f_t+g_t+r_td_t+(r_t-q_t)I_t - r_tF_{t-1}(a)+a \leq b\}$$

Clearly, values of a for which $F_{t-1}(a)+1-d_t \geq 0$ can be ignored. If $r_t - q_t \leq 0$, then it is optimal to take $I_t = \min\{-1, F_{t-1}(a)+c_t-d_t\}$ for all remaining values of a. Since this value is non–decreasing when a increases, we can use a similar approach as for Subproblem (II).

If $r_t - q_t > 0$ we propose essentially the same approach as the one for Subproblem (III) in the case $r_t + s_t > 0$. The following observations — of which the proof is left to the reader — are useful.

1. Define $b^l(a) \equiv f_t+g_t+r_t+(r_t-q_t)(F_{t-1}(a)+1-d_t)-r_tF_{t-1}(a)+a$. It holds that $b^l(a) \geq a$ for all a with $F_{t-1}(a)+1-d_t \leq -1$ (i.e., for all relevant values of a).

2. Define $b^u(a) \equiv f_t+g_t+r_t+(r_t-q_t)(\min\{-1, F_{t-1}+c_t-d_t\})-r_tF_{t-1}(a)+a$. For every $\bar{a} \in \{1, 2, \ldots, B-1\}$ it holds that $b^l(\bar{a}) \geq b^l(\bar{a}+1)$ if and only if

$b^u(\bar{a}) \geq b^u(\bar{a}+1)$. Moreover, if $b^l(\bar{a}) \geq b^l(\bar{a}+1)$, then for $b^l(\bar{a}) \leq b \leq b^u(\bar{a}+1)$:

$$\lfloor \tfrac{1}{r_t - q_t}(b - f_t - e_t - r_t d_t + r_t F_{t-1}(\bar{a}) - \bar{a}) \rfloor \leq$$

$$\lfloor \tfrac{1}{r_t - q_t}(b - f_t - e_t - r_t d_t + r_t F_{t-1}(\bar{a}+1) - \bar{a}+1) \rfloor$$

and for $b^u(\bar{a}+1)+1 \leq b \leq b^u(\bar{a})$:

$$\lfloor \frac{1}{r_t - q_t}(b - f_t - e_t - r_t d_t + r_t F_{t-1}(\bar{a}) - \bar{a}) \rfloor \leq \min\{-1, F_{t-1}(\bar{a}+1) + c_t - d_t\}$$

This implies that $\bar{a}$ may be ignored while computing the maxima. Therefore, it suffices to consider a particular subsequence of $a = 0, 1, \ldots, B$ which has the property that both $b^l(a)$ and $b^u(a)$ are strictly increasing in a.

It is now left to the reader to verify that the same approach as discussed for Subproblem (III) can be applied. Hence, for fixed value of t, Subproblem (IV) is also solvable in $\mathcal{O}(B)$ time. We have now derived the following result.

Theorem 20 *If all cost functions are pseudo–linear, then it takes $\mathcal{O}(TB)$ time to compute* $\max_{0 \leq a \leq b} \max_{0 \leq x_t \leq c_t} \{F_{t-1}(a) + x_t - d_t \mid p_t(x_t) + h_t(F_{t-1}(a) + x_t - d_t) \leq b - a\}$ *for all $b \in \{0, 1, \ldots, B\}$ and all $t \in \{1, 2, \ldots, T\}$.*

The maximization in the theorem above is the first part of recursion (5). The other part consists of evaluating $\max\{I_t \geq 0 \mid \exists\ a \in \{0, 1, \ldots, b\} :\ I_t < F_{t-1}(a) - d_t,\ a + h_t(I_t) \leq b\}$. It is left to the reader to verify that in case the holding cost functions are pseudo–linear, this expression can be computed for all $b \in \{0, 1, \ldots, B\}$ and all $t \in \{1, 2, \ldots, T\}$ in a total computational effort which is $\mathcal{O}(TB)$. The crucial observation to achieve the reduction in complexity is that the binary searches which were needed in the general case can now be replaced by $\mathcal{O}(1)$ computations.

This appendix can now be summarized as follows.

Theorem 21 *If all cost functions are pseudo–linear, then the complexity of the dynamic programming algorithm based on formulas (1) and (5) is $\mathcal{O}(TB)$.*

References

Aho, A.V., J.E. Hopcroft and J.D. Ullman. 1983. *Data Structures and Algorithms.* Addison–Wesley Publishing Company, Reading, Massachusetts, 53–60.

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.

Bitran, G.R. and H. Matsuo. 1986. Approximation Formulations for the Single–Product Capacitated Lot Size Problem. *Operations Research* **34**, 63–74.

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

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

Chen, H.-D., D. Hearn and C.-Y. Lee. 1994b. A Dynamic Programming Algorithm for Dynamic Lot Size Models with Piecewise Linear Costs. *Journal of Global Optimization* **4**, 397-413.

Chen, W.-H., and J.-M. Thizy. 1990. Analysis of Relaxations for the Multi–Item Capacitated Lot–Sizing Problem. *Annals of Operations Research* **26**, 29–72.

Chung, C.-S., and C.-H. M. Lin. 1988. An $\mathcal{O}(T^2)$ 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. 1994. An Effective Algorithm for the Capacitated Single Item Lot Size Problem. *European Journal of Operational Research* **75**, 427-440.

Constantino, M. 1995. Lower Bounds in Lot–Sizing Models: A Polyhedral Study. Working Paper 12/95 FCUL, Universidade de Lisboa, Lisbon, Portugal (to appear in *Mathematics of Operations Research*).

Erenguc, S.S., and Y. Aksoy. 1990. A Branch and Bound Algorithm for a Single Item Nonconvex Dynamic Lot Sizing Problem with Capacity Constraints. *Computers and Operations Research* **17**, 199-210.

Flippo, O.E., A.W.J. Kolen, A.M.C.A. Koster and R.L.M.J. Van de Leensel. 1996. A Dynamic Programmimg Approach for the Local Access Network Expansion Problem.

METEOR Research Memorandum RM/96/027, Faculty of Economics and Business Administration, Maastricht University, Maastricht, The Netherlands.

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. Rinnooy Kan. 1980. Deterministic Production Planning: Algorithms and Complexity. *Management Science* **26**, 669-679.

Gavish, B., and R.E. Johnson. 1990. A Fully Polynomial Approximation Scheme for Single–Product Scheduling in a Finite Capacity Facility. *Operations Research* **38**, 70–83.

Lambrecht, M., and J. Vander Eecken. 1978. A Capacity Constrained 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.

Karmarkar, U.S., S. Kekre and S. Kekre. 1987. The Dynamic Lot–Sizing Problem with Startup and Reservation Costs. *Operations Research* **35**, 389–398.

Kirca, Ö. 1988. An Efficient Algorithm for the Capacitated Single Item Dynamic Lot Size Problem. *European Journal of Operational Research* **45**, 15-24.

Kovalyov, M.Y. 1995. Improving the Complexities of Approximation Algorithms for Optimization Problems. *Operations Research Letters* **17**, 85–87.

Lofti, V., and Y.-S. Yoon. 1994. An Algorithm for the Single Item Capacitated Lot-sizing Problem with Concave Production and Holding Costs. *Journal of the Operational Research Society* **45**, 934-941.

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

Pochet, Y. 1988. Valid Inequalities and Separation for Capacitated Economic Lot Sizing. *Operations Research Letters* **7**, 109-116.

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

Pochet, Y., and L.A. Wolsey. 1995. Algorithms and Reformulations for Lot–Sizing Problems. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **20**, 245–293.

Rosling, K. 1993. A Capacitated Single–Item Lot–Size Model. *International Journal of Production Economics* **30–31**, 213–219.

Salomon, M., L.G. Kroon, R. Kuik and L.N. Van Wassenhove. 1991. Some Extensions of the Discrete Lotsizing and Scheduling Problem. *Management Science* **37**, 801–812.

Shaw, D.X., and A.P.M. Wagelmans. 1995, An Algorithm for Single-item Capacitated Economic Lot-Sizing with Piecewise Linear Production Costs and General Holding Costs. Discussion Paper TI 95-167, Tinbergen Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands (to appear in *Management Science*).

Swoveland, C. 1975. A Deterministic Multi–Period Production Planning Model with Piecewise Concave Production and Holding–Backorder Costs. *Management Science* **21**, 1007–1013.

Veinott, A.F., Jr. 1964. Production Planning with Convex Costs: A Parametric Study. *Management Science* **10**, 441-460.

Van Hoesel, C.P.M., and A.P.M. Wagelmans. 1996. An $\mathcal{O}(T^3)$ Algorithm for the Economic Lot-Sizing Problem with Constant Capacities. *Management Science* **42**, 142–150.