ON THE TRIPLE ALGORITHM
FOR OPTIMAL DYNAMIC LOTSIZING

ROBERT W. GRUBBSTRÖM

DEPARTMENT OF PRODUCTION ECONOMICS
LINKÖPING INSTITUTE OF TECHNOLOGY
S-581 83 LINKÖPING, SWEDEN
e-mail: rwg@ipe.liu.se

The *Wagner-Whitin* and the *Silver-Meal algorithms* for lotsizing are normally presented in terms of an average cost function in discrete time. An additional forward optimal dynamic lotsizing algorithm has been presented by the current author recently, here called *the triple algorithm*. This algorithm may be applied either time is discrete or not, and either the traditional average cost or the net present value is the objective function.

Obvious observations for dynamic lotsizing are that (i) when the setup cost is very high compared to the holding cost, there can only be one setup at the very beginning of the process, and (ii) in the converse case with a very low setup cost, there must be a setup at every time a non-zero requirement occurs. Therefore, the domain in which dynamic lotsizing exists as a non-trivial problem, is when this ratio is neither high, nor low.

Extensive simulations applying the triple algorithm have indicated that the speed of the algorithm appears to be of the order $n \log n$, where n is the number of points in time with a non-zero requirement. The paper suggest that this property should be examined further, i. e. the complexity of the algorithm, from a more theoretical point of departure. Underlying this question is how the complexity of the problem depends on the setup cost/holding cost ratio.

Paper submitted for presentation at the *Third Aegean International Conference on Design and Analysis of Manufacturing Systems*, Tinos Island, Greece, May 19-22, 2001.

Keywords: Dynamic lotsizing, inventory theory, net present value, annuity stream.

1. Introduction

The *dynamic lot sizing problem* is a celebrated problem, the two most well-known solution procedures being the Wagner-Whitin dynamic programming algorithm [8] leading to an optimum solution and the Silver-Meal heuristic [7] leading to an approximately optimum solution. Recently, Federgruen and Tzur [1] have developed an algorithm similar to ours for a discrete time process and they also list a couple of other available algorithms. The policy presented below, which is of the forward type, is more complex than the Silver-Meal heuristic, but it needs less computational efforts than the Wagner-Whitin policy. Our algorithm performs in continuous time, a condition which appears not to have been given attention to any greater extent in the literature. However, this policy also works perfectly well in discrete time. As the objective function is either used the *net present value* (NPV) or the *average cost* (AC). Our algorithm, here called *the triple algorithm*, applies to a deterministic one-level demand process allowing no backlogging. Based on a difference equation describing the relation between setup decisions and resulting production decisions, the problem is stated in terms of a set of zero/one decision variables.

2. Basic Problem

We consider a finite horizon $\hat{T}$ within which there are given requirements of a single product in the amounts $d_1,\ d_2,...,\ \ d_n$ at given times $t_1,\ t_2,...,\ \ t_n$. By convention we set $t_{n+1}=\hat{T}$. Let production at t_i be p_i. Cumulative requirements are written $D_i = \sum_{j=1}^{i} d_j$ and cumulative production $P_i = \sum_{j=1}^{i} p_j$, i = 1, 2, ... , n. The problem is to choose the amount to produce (or not to produce) at each time such that an objective function, either chosen as the NPV of the cash consequences involved or the traditional AC. As a special case we may also consider when all times t_i are equidistant turning the settings into a discrete-case problem. We limit the economic consequences to a setup cost K for each time a production takes place and a variable production cost c for each unit produced. The continuous interest rate applied is written ρ to be used when applying the NPV principle and the inventory holding cost per unit and time unit may then be interpreted as $h = \rho c$ for the average cost approach. Obviously, if K is sufficiently large compared to h, there will be only one setup, and if this ratio is sufficiently small, there will be a setup at every t_i. The problem therefore only exists for a medium range of the ratio K/h.

In previous articles it has been shown that for either of these two objective functions, the optimal production level must be chosen among the alternatives such that cumulative production always coincides with cumulative requirements for the current or a later interval [3,5,6]. This may be illustrated as a relationship between a right staircase function describing cumulative requirements and a left staircase describing cumulative production. Either the AC or the NPV is applied, the two staircases must fit into each other at the ”inner corners” when replenishments take place (indicated with circles in the left part of Figure 1). Thus the optimal number of batches produced cannot be higher than the number of events generating requirements. Also the final level of cumulative production and cumulative requirements must coincide, $P_n = D_n$.

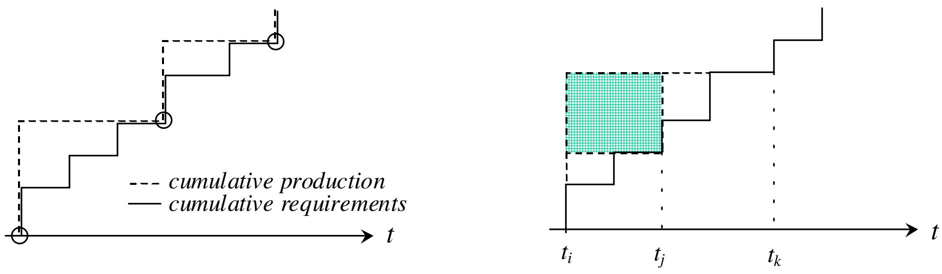

Figure 1. *Left*: The staircases of cumulative production and cumulative requirements. *Right*: The shaded area represents M_{ijk} (Section 4), its size determining whether or not a setup at t_j is justified.

Since there always is a setup at time t_1 there are altogether 2^{n-1} different alternatives candidating for the optimum production staircase, for n = 40 we have 2^{n-1}=549755823888. The problem is to find the best staircase. As decision variables we choose zero/one variables α_i for i = 2, 3, ... , n, letting $\alpha_i = 1$ mean there is a setup at t_i and $\alpha_i = 0$ that there is no setup at t_i. We always have $\alpha_1 = 1$. The basic difference equation relating cumulative production to requirements using the variables α_i may now be written $P_i = \alpha_{i+1} D_i + (1-\alpha_{i+1}) P_{i+1}$, i = 1, 2, ... , n – 1. If $\alpha_{i+1} = 1$ (a setup at time t_{i+1}), then the (i+1)st inner corners coincide, i. e. the ith cumulative production and ith cumulative requirement levels are equal, and if $\alpha_{i+1} = 0$, then instead the ith and (i+1)st cumulative production levels are the same. This difference equation has the solution $P_i = D_i + \sum_{j=i+1}^{n} d_j \prod_{k=i+1}^{j} (1-\alpha_k)$. Hence, production levels can conveniently be abandoned as decision variables in favour of zero/one decisions α_i, for i = 2, 3, ... , n.

3. Objective Functions

If we choose the net present value as our objective to be maximised, defining $P_0 = 0$, we have:

$$\text{NPV} = -\sum_{i=1}^{n} \alpha_i K e^{-\rho t_i} - cD_n - c\sum_{i=1}^{n-1} \left(e^{-\rho t_i} - e^{-\rho t_{i+1}}\right)\left(D_i + \sum_{j=i+1}^{n} d_j \prod_{k=i+1}^{j} (1-\alpha_k)\right). \tag{1}$$

If, instead applying the AC approach, this measure may be obtained from a first order expansion of the *annuity stream* corresponding to the NPV, i.e. the equivalent cash flow generating the same NPV, cf. e.g. [7], giving

$$\text{Annuity stream} = \frac{\rho \,\mathrm{NPV}(\rho)}{1-e^{\rho \hat{T}}} = \left(\frac{1}{\hat{T}} + \frac{\rho}{2} + \frac{\rho^2 \hat{T}}{12} + \ldots \right) \mathrm{NPV}(\rho)\,. \tag{2}$$

If the interest is set to zero, the annuity stream (the zeroth order term) will amount to the average costs over time excluding capital costs, which in this case simply will be a negative figure amounting to the sum of setup costs and production costs. However, the dynamic lot-size problem requires a holding cost and so ρ is positive in our treatment.

To obtain an average cost approximation from the NPV, we take a zeroth order Maclaurin expansion of the setup cost term of the NPV and a first order expansion of the holding cost term. The zeroth order expansion is the usual approximation used for setup costs not taking discounting effects of distant setups into consideration. This results in the following expression:

$$\text{Annuity stream} \approx - \frac{\sum_{i=1}^{n} \alpha_i K + cD'_n + \rho\, c \sum_{i=1}^{n-1} \left(t_{i+1} - t_i\right) \left(D'_i + \sum_{j=i+1}^{n} d'_j \prod_{k=i+1}^{j} (1-\alpha_k) \right)}{\hat{T}}\,. \tag{3}$$

So, the average cost AC is given by

$$\mathrm{AC} = - \frac{\sum_{i=1}^{n} \alpha_i K + cD_n + h \sum_{i=1}^{n-1} \left(t_{i+1} - t_i\right) \left(D_i + \sum_{j=i+1}^{n} d_j \prod_{k=i+1}^{j} \left(1-\alpha_k\right) \right)}{\hat{T}}\,, \tag{4}$$

with h interpreted as ρc.

Computing the derivatives of NPV and AC, we obtain expressions of the type

$$\frac{\partial \ \mathrm{NPV}}{\partial \alpha_k} \cdot \frac{e^{\rho t_k}}{K} = \tag{5}$$

$$= -1 + \frac{c\, e^{\rho t_k}}{K} \left(\sum_{i=1}^{k-1} \left(e^{-\rho t_i} - e^{-\rho t_{i+1}} \right) \prod_{l=i+1}^{k-1} \left(1-\alpha_l\right) \right) \cdot \left(\sum_{j=k}^{n} d_j \cdot \prod_{l=k+1}^{j} \left(1-\alpha_l\right) \right),$$

$$\frac{\partial \ \mathrm{AC}}{\partial \alpha_k} \cdot \frac{1}{K} = -1 + \frac{\rho\, c}{K} \left(\sum_{i=1}^{k-1} \left(t_{i+1} - t_i \right) \prod_{l=i+1}^{k-1} \left(1-\alpha_l\right) \right) \cdot \left(\sum_{j=k}^{n} d_j \cdot \prod_{l=k+1}^{j} \left(1-\alpha_l\right) \right), \tag{6}$$

where the convention $\prod_{l=i+1}^{i} \left(1-\alpha_l\right) = 1$ is adopted. Hence, we reach the important separation result that the sign of derivatives may be separated into two factors with mutually exclusive sets of past and future decision variables and where the former factor only depends on the past given times and the other only on future required amounts. If one of these derivatives is positive then there must be a setup at that point, i.e. if $c e^{\rho t_k}\left(e^{-\rho t_i}-e^{-\rho t_{i+1}}\right)d_k / K > 1$ or $\rho c\left(t_{i+1}-t_i\right)d_k / K > 1$, respectively, for some k, then $\alpha_k = 1$. This provides a local sufficient condition for production necessarily to occur. For each value of k we can therefore immediately examine whether or not this condition is fulfilled, and if so, for each such qualifying period, we must have a setup. Sequences of intervals surrounded by known setups may be treated independently.

4. Algorithm

Based on the signs of the derivatives (5)-(6), the algorithm works as follows. Consider three times at which demand events occur t_i, t_j, and t_k, where $i < j < k$, (cf right part of Figure 1). If there is a setup at t_j, it has to be justified by the inventory reduction shown by the shaded rectangle. Define a measure M_{ijk} = $\rho c\left(t_j - t_i\right)\left(D_{k-1} - D_{j-1}\right)/K$ for the AC case, this justification meaning $M_{ijk} > 1$. A corresponding similar measure and condition applies for the NPV. For a given i, a setup requires therefore that such a condition is valid for the next two setups j, k. Letting k increase from i+1 and upwards, the latest point at which the *next* setup must occur is the highest value of k such that for all intermediate j we have $M_{ijk} \leq 1$. If this were not so, an intermediate setup at some j would be necessary. Hence, for any given i, we define a *latest* point $L(i)$ by $L(i) = \underset{j, M_{ijk} \leq 1}{\text{Max}}\ k$. Therefore, if $k > L(i)$, an intermediate setup somewhere in the interval is required. On the other hand, if there are optimal setups at i and $k > L(i)$, then the intermediate setup must be chosen so as to maximise its effect, i. e. to maximise M_{ijk}. Furthermore, as k increases, the maximising j cannot decrease, i.e. the maximising $j(k)$ is a nondecreasing function of k. Hence, given a setup at i, an *earliest* time at which the next optimal setup is possible is the j maximising $M_{i\,j(L(i)+1)}$. This earliest next setup is thus defined by $E(i) = \text{Argmax}\, M_{i\,j(L(i)+1)}$. Obviously $E(i) \leq L(i)$. In the equality case, a setup at i must be followed by a setup at j = $E(i)$ = $L(i)$. Hence, each i defines an interval [$E(i)$, $L(i)$], within which the next optimal setup must occur.

Now, given i, we examine which next k (second next setup) that is possible. This k is at least $L(i)$ + 1 and not later than such that the intermediate maximising j falls within [$E(i)$, $L(i)$]. A given i thus generates a set of possible triples (i, j, k) which are compatible with this condition. For each such triple, starting with its j component, a new set of triples is generated, and so on. At a particular *stage* defined as the index pertaining to a setup (first setup is stage 1, etc.), there exists a set of possible triples with one or more elements. It might seem that the number of possible paths generated in this manner increases continuously and rapidly. However, this is not so since acceptable paths must be compatible forwards and backwards. If a triple generated at a certain stage matches no triple at the former stage, it has to be discarded. Also, any previously generated triple, not matching one of the next stage generated triples, must be discarded. This may have consequences backwards for several stages, reducing the number of elements in their sets, sometimes involving substantial reductions.

At each stage, the procedure outlined here generates a limited set of feasible paths, one of which must be optimal. In Section 5 the number of such paths is denoted Φ_{final}. When the end stage is reached (which may be different for different feasible paths), each remaining sequence may be evaluated by complete enumeration and the optimum determined. The optimum belongs to one of these remaining paths.

5. Simulation Study

A limited simulation study has been carried out to indicate some characteristics of the algorithm. Sequences of demand events were generated separated by stochastic intervals rectangularly generated on the interval [0,10]. The levels of demand were similarly generated with a rectangular distribution on the interval [0,100]. Throughout the parameters were chosen as $\rho = 0.001$, $c = 1000$, making $h = 1$ for the AC case. The number of demand events n was changed between 20 and 100, and the setup cost K was changed between 100 and 100000, samples of which Table 1 shows. For each combination of n and K, 20 runs were made. The following indicators averaged over the 20 runs are reported: Φ_{final} = average number of final sequences to be evaluated for each run, $\delta_{\max}$ = average maximum number of parallel alternative triples to have to consider at any particular stage during a run, and $\bar{N}^*$ = average optimal number of setups.

K	n=20			n=40			n=60			n=80			n=100		
	Φ_{final}	$\delta_{\max}$	$\bar{N}^*$	Φ_{final}	$\delta_{\max}$	$\bar{N}^*$	Φ_{final}	$\delta_{\max}$	$\bar{N}^*$	Φ_{final}	$\delta_{\max}$	$\bar{N}^*$	Φ_{final}	$\delta_{\max}$	$\bar{N}^*$
100	1.00	5.10	15.15	1.00	6.85	29.40	1.00	6.50	44.00	1.10	7.40	60.30	1.05	8.05	74.30
500	1.35	8.05	8.55	2.30	12.45	16.50	2.05	12.75	25.90	3.65	16.85	33.35	6.05	16.50	43.30
1000	1.55	10.85	6.35	2.70	17.75	12.40	9.70	24.10	19.65	7.00	24.85	24.55	11.30	30.70	31.75
2500	1.35	13.15	4.40	4.50	27.30	8.10	10.65	47.20	11.95	18.60	49.15	16.40	61.95	66.35	20.60
5000	1.45	14.00	3.15	3.00	39.65	5.80	7.10	54.85	9.20	20.75	73.00	12.55	22.40	86.90	14.25
7500	1.35	9.10	2.50	3.40	39.05	5.45	4.40	49.85	7.30	11.70	83.85	9.65	23.05	90.60	12.30
10000	1.40	8.50	2.15	2.15	41.50	4.30	4.85	64.70	6.45	7.30	79.90	8.45	16.60	117.35	10.35
15000	1.00	5.45	1.90	2.45	40.40	3.65	3.60	73.00	5.35	6.50	96.65	7.25	7.85	117.90	8.60
20000	1.00	3.30	1.95	1.85	28.90	3.10	3.15	66.00	4.60	4.95	102.15	5.95	8.15	137.65	7.55
30000	1.00	1.85	1.45	1.40	18.85	2.80	2.25	69.75	3.85	5.35	113.25	5.20	7.70	149.35	6.20
40000	1.00	1.00	1.05	1.30	15.10	2.20	2.30	58.35	3.20	2.85	106.85	4.55	10.00	146.70	5.60
50000	1.00	1.00	1.00	1.00	12.45	2.00	1.55	47.30	2.95	2.60	97.95	4.05	5.00	158.25	5.00
75000	1.00	1.00	1.00	1.00	7.05	1.95	1.85	26.45	2.75	2.20	77.85	3.15	3.15	145.35	4.10
100000	1.00	1.00	1.00	1.45	2.00	1.45	1.15	20.60	2.10	1.65	60.10	3.00	2.60	152.55	3.70

Table 1. Simulation results concerning complexity indices Φ_{final} and $\delta_{\max}$ and the average number of setups.

Our simulation results confirm that complexity increases towards a medium range of the ratio K/h. Also is clearly demonstrated the astonishingly few remaining paths Φ_{final} that need to be evaluated by complete enumeration. Computational aspects on the Wagner-Whitin and other algorithms are discussed in [1]. Our trials in the simulation suggest that for a number of demand events less than 15, our algorithm completes its job in a fraction of a second. For horizons in the order of 100 it takes a few seconds to reach the optimum. A complete enumeration procedure treating the same problem takes about a minute for 15 demand events and an estimated time of 10^{20} years for a 100 demand-event case! Numerical comparisons concerning computational efficiency with the Wagner-Whitin and other algorithms have yet to be made.

6. Conclusions

This paper has presented a recently developed policy, *the triple algorithm*, leading to an optimal solution of the dynamic lot-sizing problem. The policy works equally well applying the net present value principle or using an average cost approach, in continuous or discrete time. The computations needed are simple compared to the Wagner-Whitin algorithm. Comparisons with other algorithms need to be worked out. There seems to be no obstacle in applying the same procedure when modifying the assumptions in such a way that setup costs and variable unit production costs (also including holding costs in the average cost case) are allowed to vary between different periods.

Current research on this topic is directed towards examining how the complexity of the algorithm depends on the important setup cost/holding cost ratio. This question will be examined further from a more theoretical point of departure.

7. References

1. Federgruen, A., Tzur, M., A Simple Forward Algorithm to solve General Dynamic Lot Sizing Models with n Periods in $0(n \log n)$ or $0(n)$ Time, *Management Science*, Vol. 37, 1991, pp 909-925
2. Grubbström, R. W., A principle for determining the correct capital costs of work-in-progress and inventory, *International Journal of Production Research,* Vol. 18, No 2, 1980, pp. 259-271
3. Grubbström, R. W., A net present value approach to safety stocks in a multi-level MRP system, *International Journal of Production Economics*, Vol. 59, 1999, pp. 361-365
4. Grubbström, R. W., "An Optimal Forward Dynamic Lot-Sizing Policy for the NPV and Average Cost Cases in Continuous Time", in Hillery, M. T., Lewis, H. J., (Eds), *The 15th International Conference on Production Research. ICPR 15 Manufacturing for a Global Market,* Limerick, Ireland, August 9-13, 1999, pp. 381-384
5. Grubbström, R. W., Molinder , A., Safety production plans in MRP-systems using transform methodology*, International Journal of Production Economics*, Vol. 46-47, 1996, pp. 297-309

6. Tang, O., *Optimal Safety Stocks in an MRP System Applying a Binary Matrix for the Decision Variables*, Third ISIR Summer School on Inventory Modelling, Aug. 25-29, 1997, Ioannina, Greece
7. Wagner, H. M., Whitin, T. M., A Dynamic Version of the Economic Lot Size Model, *Management Science*, Vol. 5, 1958, pp. 89-96
8. Silver, E. A., Meal, H. C., A Heuristic for Selecting Lot Size Quantities for the Case of a Deterministic Time Varying Demand Rate and Discrete Opportunities for Replenishment, *Production and Inventory Management*, Vol. 14, 1973, pp. 64-74

8. Acknowledgement

The author gratefully acknowledges the many valuable suggestions and comments made by Dr Ou Tang, Department of Production Economics, and by Professor Lars-Erik Andersson, Department of Mathematics, Linköping Institute of Technology.