

LOT-SIZING PROBLEM WITH SEVERAL PRODUCTION CENTERS

Franklina Maria Bragion de Toledo *

Departamento de Matemática Aplicada e Estatística
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP)
São Carlos – SP
fran@icmc.usp.br

André Luís Shiguemoto

Escola de Engenharia de Piracicaba
Fundação Municipal de Ensino de Piracicaba
Piracicaba – SP
andrels@densis.fee.unicamp.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 07/2004; aceito em 07/2005

Received July 2004; accepted July 2005

Abstract

In this paper, a case study is carried out concerning the lot-sizing problem involving a single item production planning in several production centers that do not present capacity constraints. Demand can be met with backlogging or not. This problem results from simplifying practical problems, such as the material requirement planning (MRP) system and also lot-sizing problems with multiple items and limited production capacity. First we propose an efficient implementation of a forward dynamic programming algorithm for problems with one single production center. Although this does not reduce its complexity, it has shown to be rather effective, according to computational tests. Next, we studied the problem with a production environment composed of several production centers. For this problem two algorithms are implemented, the first one is an extension of the dynamic programming algorithm for one production center and the second one is an efficient implementation of the first algorithm. Their efficiency are shown by computational testing of the algorithms and proposals for future research are presented.

Keywords: production planning; lot-sizing; dynamic programming; backlogging.

Resumo

Neste trabalho, estudamos um caso particular do problema de dimensionamento de lotes, envolvendo o planejamento da produção de um único item em vários centros produtivos que não apresentam restrições de capacidade. A demanda pode ser atendida com ou sem atraso. Este problema surge da decomposição de problemas práticos, como o sistema de requerimento de materiais (MRP), e também de problemas de dimensionamento de lotes com múltiplos itens e capacidade de produção limitada. Primeiramente, apresentamos uma implementação eficiente de um algoritmo de programação dinâmica progressiva proposto para a solução do problema com um único centro produtivo. Embora, essa implementação não reduza a complexidade do algoritmo, experimentos computacionais mostraram que ela é significativamente mais rápida. Em seguida, estudamos o problema com vários centros, para o qual são implementados dois algoritmos, o primeiro é uma extensão do algoritmo original e o segundo incorpora as idéias que tornam essa implementação mais eficiente. O desempenho computacional dos algoritmos é analisado e propostas para futuras pesquisas são apresentadas.

Palavras-chave: planejamento da produção; dimensionamento de lotes; programação dinâmica; demanda com atraso.

1. Introduction

The single-item lot-sizing problem without capacity constraints has been studied since 1958 and a wide array of new results have been presented in recent years. Good reviews have been presented by Wolsey (1995) and Brahimi *et al.* (2006). This problem often arises in practice as a sub-problem derived from the relaxation of capacity constraints of one of the most important problems of production planning: the multi-item lot-sizing problem with capacity constraints. Some of the methods proposed using this relaxation for such problems have been presented by Trigeiro *et al.* (1989), Lozano *et al.* (1991) and Diaby *et al.* (1992a, 1992b), among others.

Bahl *et al.* (1987) classified the various types of lot-sizing problems: single-stage (with one planning stage) or multi-stage (with several planning stages). A system has one single stage when the items to be produced are independent, that means, one does not depend on another to be produced. In multi-stage systems, the final item has component items, whose production or purchase should also be planned. According to the authors, another important criteria for classifying problems is the available production capacity, which can be constrained or not. Several research studies on this topic have been published in the literature and good reviews have been presented by Bahl *et al.* (1987), Drexl & Kimms (1997) and Kuik *et al.* (1994). Wolsey (2002) proposes a new problem classification and reformulation. Recently, Karimi *et al.* (2003) have presented a good review of models and algorithms. In this review, the authors consider single-level lot-sizing problems.

In this paper, a problem involving single-item, single-stage production planning, with no capacity constraint where backlogging is permitted in a production environment composed of several production centers is addressed. Today it is common to find companies where several plants, or production centers or assembly lines are available for the production of items. A production environment with several production centers can be generally found in food, beverage and steel industries, for example, see Sambasivan & Yahya (2005). Solving non-constraint problems enables us to solve lot-sizing problems with capacity constraints. When backlogging is possible, problems become more similar to actual situations since in practice deadlines can generally be negotiated, whether they add costs to companies or not.

In the literature, only few studies considering several production centers. Carreno (1990) proposed a heuristic to solve the economic lot-sizing problem with identical parallel machines. Sabbag (1993) and Toledo (1998) proposed algorithms for the solution of lot-sizing problems with distinct parallel machines with limit capacity. Sung (1986) proposed an optimal algorithm to solve a single product parallel facility production planning model where demands must be met and where there are no capacity constraints. Armentano & Toledo (1997) adapted an efficient implementation of Evans (1985) and a lower complexity algorithm proposed by Wagelmans *et al.* (1992). The effect of the lower complexity algorithm can be noticed just for a number of planning periods greater than 250.

In the other hand, the problem with a single production center has been studied for many years. The first optimal method to solve the problem where backlogging is not possible was proposed by Wagner and Whitin (1958), who presented a dynamic programming algorithm. It computes the minimum cost path in a directed graph and has polynomial complexity $O(T^2)$, where T is the number of periods over the planning horizon. Evans (1985) proposed an efficient implementation of the Wagner-Whitin algorithm, which, although not reducing its complexity, has shown to be rather effective, according to computational tests. In Bahl & Taj (1991), this code was adapted for the case where unit production costs do not decrease over time.

Almost ten years ago, several independent developments in the area – Aggarwal & Park (1993), Ferdergruen & Tzur (1991), Wagelmans *et al.* (1992) – led to exact algorithms which are solved in $O(T \log T)$ operations. Complexity can be reduced to $O(T)$ when unit production costs do not decrease during the planning horizon.

Zangwill (1969) considers the problem where backlogging is possible. This author was the first to highlight the importance of using networks to represent some production planning problems. The problem was represented as a minimum cost flow problem, in a network with concave costs on arcs and one single source. The proposed algorithm has $O(T^2)$ time complexity. Pochet & Wolsey (1988) proposed its reformulation and solution through the use of cutting planes. Aggarwal & Park (1993) developed an algorithm with complexity $O(T \log T)$. This algorithm solves a problem with T periods, recursively, by a procedure that solves two sub-problems with $T/2$ periods, successively. Finally, a problem involving the solution of the sub-problems is solved in linear time using array-searching techniques. According to Ferdergruen & Tzur (1993), the constant factor in the time bound is therefore quite large for a similar recursive procedure in Wilber (1988) (see Galil & Giancarlo, 1989). Ferdergruen & Tzur (1993) also proposed an algorithm with $O(T \log T)$ time complexity. Their algorithm consists of iterative updates of a list of candidate last order periods, as the planning horizon t is incremented to $t+1$. The $T \log T$ complexity term in the most general procedure is obtained from effort to insert or delete an element in the list which in the worst case may be of size T , but in practice is very small indeed. Golany *et al.* (1992) propose a decomposition technique which splits the problem of the production quantity decision over a larger number of (discrete) periods into a sequence of smaller problems, each dealing with a small set of consecutive periods.

This paper is organized as follows: in the next section the mathematical model for a single production center is presented and an efficient implementation of a forward dynamic programming algorithm as well as the computational results achieved by comparing these implementations are discussed. The extension of this algorithm to several production centers and its efficient implementation are proposed in Section 3. Computational results comparing both implementations presented in Section 3 are analyzed in Section 4. Finally, in Section 5 the results obtained and suggestions for future applications of this problem are discussed.

2. Lot-sizing Problems with Backlogging

The lot-sizing problem considered in this section consists of planning the production of one single item over a planning horizon subdivided in T periods. The demand of the item is known and backlogging may exist. The purpose is to minimize the sum of fixed setup costs of the production center, variable production costs, and inventory carrying cost, as well as the backorder cost of the item.

Zangwill (1969) extended the Wagner & Whitin model (1958) to allow the inventory to be negative. The following notations are utilized:

- s_t – setup costs in period t ;
- c_t – unit production costs in period t ;
- h_t – unit inventory carrying cost in period t ;
- hb_t – unit backorder cost of item in period t ;
- d_t – item demand in period t .

In order to formulate a lot-sizing problem with mixed integer programming, the following variables are considered:

- x_t – number of units produced in period t ;
- I_t – inventory level at the end of period t .

Production costs of x_t are given by:

$$g_t(x_t) = \begin{cases} s_t + c_t x_t & \text{if } x_t > 0; \\ 0 & \text{if } x_t = 0; \end{cases}$$

and inventory costs of I_t are given by:

$$H_t(I_t) = \begin{cases} h_t I_t & \text{if } I_t \geq 0; \\ hb_t I_t & \text{if } I_t \leq 0. \end{cases}$$

Lot-sizing problems can be written as follows (Zangwill, 1969):

$$\text{Min} \quad \sum_{t=1}^T (g_t(x_t) + H_t(I_t)) \quad (1)$$

$$\text{subject to} \quad x_t + I_{t-1} - I_t = d_t \quad t = 1, \dots, T; \quad (2)$$

$$I_0 = 0 \quad (3)$$

$$x_t \geq 0 \quad t = 1, \dots, T. \quad (4)$$

The objective function (1) describes the sum of production and inventory costs. In (2), inventory balance constraints are presented. Without losing generality, constraint (3) ensures that the initial inventory is zero. Non-negative production levels are ensured by constraints (4).

When backlogging is allowed, inventory in period t can be negative. To represent the problem as a network, Zangwill (1969) defined I_t^+ as the inventory at the end of period t and I_t^- as the accumulated demand that has not been met at the end of period t . Thus,

$$I_t = \begin{cases} I_t^+ & \text{if } I_t \geq 0; \\ I_t^- & \text{if } I_t \leq 0; \end{cases}$$

where $I_t^+ \geq 0$ and $I_t^- \geq 0$. Therefore the model (1-4) can be reformulated as a function of variables I_t^+ and I_t^- , as shown below, and can be represented by a network (Zangwill, 1969), see Figure 1.

$$\text{Min} \quad \sum_{t=1}^T (g_t(x_t) + h_t I_t^+ + hb_t I_t^-)$$

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

$$I_0^+ = 0, \quad I_0^- = 0$$

$$x_t \geq 0, \quad I_t^+ \geq 0, \quad I_t^- \geq 0 \quad t = 1, \dots, T.$$

The author proved that in a minimum cost network flow problem where the objective function is a concave function of the arc flows and the network has only one source, there is an optimal solution having the property that each node has at most one positive input. That is, no more than one variables I_{t-1}^+ , I_t^- , and x_t can be positive. This problem can be solved by backward dynamic programming using recursive equations proposed by Zangwill (1969) or by equivalent forward dynamic programming presented in Johnson & Montgomery (1974) describe below.

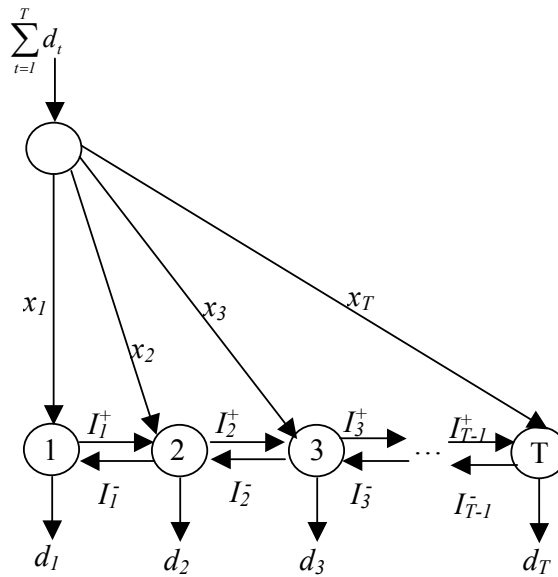


Figure 1 – The network for the single production center problem.

Let F_t denote the minimum production cost from period k up to period t , then F_t can be determined recursively as

$$F_t = \min_{0 \leq k \leq t-1} [F_k + M_{kt}], \quad t=1, 2, \dots, T \quad (5)$$

$$F_0 = 0$$

where M_{kt} is define as the cost of producing in period i ($k+1 \leq i \leq t$), in order to meet the demands from period $k + 1$ to period t ,

$$M_{kt} = \begin{cases} s_{k+1} + c_{k+1} d_{k+1} & \text{if } t = k+1 \\ \min_{k+1 \leq i \leq t} \left[s_i + c_i D_i + \sum_{l=k+1}^{i-1} hb_l I_l^- + \sum_{l=i}^{t-1} h_l I_l^+ \right] & \text{if } t > k+1. \end{cases} \quad (6)$$

where $D_i = \sum_{r=i+1}^t d_r$ is cumulative demand from period $k+1$ to t .

The M_{kt} values can be calculated in efficient way if we applied the computational modifications suggested by Evans (1985) for Wagner & Whitin case (1958). The idea consists in calculating first M_{k1} and then for $t > 1$, we can calculate M_{kt} by:

$$M_{k,t+1} = M_{kt} + d_{t+1} (c_k + sumh_k)$$

where $sumh_k$ represents the cumulative inventory costs and it is calculated by

$$sumh_k = \sum_{l=k+1}^{t-1} h_l .$$

By adapting the computational modifications suggested by Evans (1985) we obtain the efficient implementation shown in Figure 2. The complexity of this algorithm is $O(T^2)$. The adaptations of the original algorithm are in steps 2-6, 10-12, 14 and 18 of the algorithm of Figure 2. In steps 2-6 we initialize the value of M_k and we calculate the cumulative demand from period 1 to period t . The cumulative inventory cost and the value of M_k are actualized in efficient way in steps 10-12, we can use the old values to obtain new values. This idea is also applied in step 18.

```

[1]  Read data (s, c, h, hb, d)
[2*]  $D_0 = 0$ 
[3*] For  $t = 1$  to  $T$  do
[4*]    $M_t = s_t + c_t d_t$ 
[5*]    $D_t = D_{t-1} + d_t$ 
[6*]    $Sumh_t = 0$ ;
End For
[7*]  $F_0 = 0$    $F_1 = M_1$ 
[8]  For  $t = 2$  to  $T$  do
[9]    $F_t = 999999.0$ ;
[10*] For  $k = 1$  to  $t-1$  do
[11*]    $Sumh_k = Sumh_k + h_{t-1}$ 
[12*]    $M_k = M_k + d_t (c_k + Sumh_k)$ 
End For
[13] For  $k = t-1$  down to  $0$  do
[14*]   if  $(k+1 = t)$  then  $Aux = M_t$ 
[15]   Else Begin
[16]      $Aux = 999999.0$ ;
[17]     For  $i = k+1$  to  $t$  do
[18*]        $Temp = M_i + c_i (D_{i-1} - d_k)$ 
[19]       For  $l = i-1$  down to  $k+1$  do
[20]          $Temp = Temp + hb_l (D_l - d_k)$ 
[21]       if  $(Temp < Aux)$  then  $Aux = Temp$ ;
End For
End Else
[22]   If  $(Aux + F_k < F_t)$  then  $F_t = Aux + F_k$ 
End For
End For

```

Figure 2 – An efficient implementation of algorithm.

Computational tests have been run which are described below, in order to evaluate the performance of the original algorithm and the efficient implementation proposed in this section. The code was implemented in C++ language and tests were run on a 1.5GHz Pentium IV microcomputer (256MB RAM, Intel processor and LINUX operational system).

For a given number of T planning periods, we randomly generate 10 instances. We considered $T = 50, 100, 250$ and 500 and the parameters for each period were generated as in Armentano & Toledo (1987) from the following uniform distributions (U):

- Demand – $U[0, 100]$;
- Setup costs – $U[80, 120]$;
- Unit production costs – $U[1, 5]$;
- Unit inventory costs – $U[1, 4]$;
- Unit backlogging costs – $U[1, 4]$.

In Table 1, the average running time for both algorithms is presented. As shown in the table, the algorithm implemented in an efficient manner is generally 50% faster than the original one. However, it should be stressed that, although the algorithm is computationally more efficient, its complexity was not reduced.

Table 1 – Results of the computational tests.

Period	Original algorithm	Efficient implementation
50	0.03s	0.01s
100	0.45s	0.26s
250	16.29s	9.53s
500	4m16s	2m29s

3. Lot-sizing Problem for Several Production Centers

In this step, the lot-sizing problem described in the previous section is considered. However, in each of the T planning periods, the production environment is composed of m production centers (manufacturing cells, production lines or machines). In order to formulate the problem for multiple centers, the variable of production x_{jt} represents the number of units produced in production center j in period t and the lot-sizing problem can be written as follows:

$$\begin{aligned}
 \text{Min} \quad & \sum_{t=1}^T \left[\sum_{j=1}^m g_{jt}(x_{jt}) + h_t I_t^+ + hb_t I_t^- \right] \\
 \text{subject to} \quad & \sum_{j=1}^m x_{jt} + I_{t-1}^+ - I_{t-1}^- - I_t^+ + I_t^- = d_t \quad t = 1, \dots, T; \\
 & I_0^+ = 0, \quad I_0^- = 0 \quad t = 1, \dots, T; \\
 & x_{jt} \geq 0, \quad I_t^+ \geq 0, \quad I_t^- \geq 0 \quad t = 1, \dots, T, \quad j = 1, \dots, m.
 \end{aligned}$$

Similarly to the single production center problem, this problem can be represented by a minimum cost network flow problem with only one source, see Figure 3. Then, there is an optimal solution having the property that each node has at most one positive input, that is, no more than one of the variables I_{t-1}^+ , I_t^- and x_{jt} (for $j = 1, \dots, m$) can be nonzero. Therefore, this problem can be solved by an adaptation of the algorithm presented in Section 2 or by an extension of the lower complexity algorithm proposed by Ferdergruen & Tzur (1993). In Armentano & Toledo (1997) the similar problem without backlogging is solved for two approaches and the authors concluded that the effect of the lower complexity algorithm can be noticed just for a number of planning periods greater than 250. In this work, we just extended the algorithm proposed in Section 2.

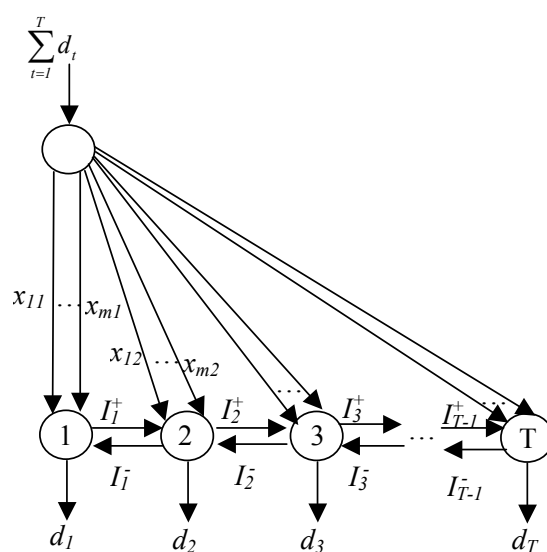


Figure 3 – The network for several production centers.

Let F_t denote the minimum production cost from period k up to period t , then F_t can be determined recursively as

$$F_t = \min_{\substack{0 \leq k \leq t-1 \\ 1 \leq j \leq m}} [F_k + M_{ktj}], \quad t=1, 2, \dots, T \tag{7}$$

$$F_0 = 0$$

where M_{ktj} is define as the cost of producing in period i ($k+1 \leq i \leq t$) in the production center j ($1 \leq j \leq m$), in order to meet the demands from period $k+ 1$ to period t ,

$$M_{kt} = \begin{cases} s_{k+1,j} + c_{k+1,j} d_{k+1} & \text{if } t = k+1 \\ \min_{k+1 \leq i \leq t} \left[s_{ij} + c_{ij} D_i + \sum_{l=k+1}^{i-1} hb_l I_l^- + \sum_{l=i}^{t-1} h_l I_l^+ \right] & \text{if } t > k+1. \end{cases} \tag{8}$$

where $D_i = \sum_{r=i+1}^t d_r$ is cumulative demand from period $k+1$ to t .

By applying the recursive formula (7), the algorithm of Figure 4 can be written. The complexity of this algorithm is $O(mT^2)$.

```

[1]  Read data ( $s, c, h, hb, d$ )
[2]   $D_0 = 0$ 
[3]  For  $t = 1$  to  $T$  do  $D_t = D_{t-1} + d_t$ 
[4]   $F_0 = 0; F_1 = 999999.0;$ 
[5]  For  $j = 1$  to  $m$  do
[6]     $M_{ij} = s_{1j} + c_{1j} d_1$ 
[7]    if ( $M_{ij} < F_1$ ) then  $F_1 = M_{ij}$ 
      EndFor
[8]  For  $t = 2$  to  $T$  do
[9]     $F_t = 999999.0$ 
[10]   For  $j = 1$  to  $m$  do
[11]     For  $k = 0$  to  $t-1$  do
[12]       if ( $k+1=t$ ) then  $aux = M_{ij}$ 
[13]       Else Begin
[14]          $aux = 999999.0;$ 
[15]         For  $i = k+1$  to  $t$  do
[16]            $EB = 0$ 
[17]           For  $l = k+1$  to  $i-1$  do  $EB = EB + hb_l (D_l - D_k)$ 
[18]            $EF = 0$ 
[19]           For  $l = i$  to  $t$  do  $EF = EF + h_l (D_t - D_l)$ 
[20]            $Temp = s_{tj} + c_{tj} (D_t - D_k) + EF + EB;$ 
[21]           if ( $Temp < aux$ ) then  $aux = Temp;$ 
          EndFor
        EndElse
[22]       if ( $aux + F_k < F_t$ ) then  $F_t = aux + F_k$ 
      EndFor
    EndFor
  EndFor

```

Figure 4 – Parallel production centers algorithm (PPCA).

As we can see in the algorithm presented above, several operations are repeated and therefore, similarly to the case of one production center, the algorithm for parallel production centers can also be implemented in an efficient manner. The adaptations of the original algorithm are in steps 5-6, 8,10, 13-15, 19 and 23 of the algorithm of Figure 5. In steps 5-6 we initialized the value of M_k and the cumulative inventory cost. In steps 8 and 10 we evaluated M_{11} . The cumulative inventory cost and the value of M_{kj} are actualized in efficient way in steps 13-15, we can use the old values to obtain new values. This idea is also apply in steps 19 and 23.

```

[1]  Read data ( $s, c, h, hb, d$ )
[2]   $D_0 = 0$ 
[3]  For  $t = 1$  to  $T$  do
[4]     $D_t = D_{t-1} + d_t$ 
[5*]   For  $j = 1$  to  $m$  do  $M_{ij} = s_{ij} + c_{ij} d_t$ 
[6*]    $Sumh_t = 0$ ;
      EndFor
[7]    $F_0 = 0$ 
[8*]    $F_1 = M_{11}$ 
[9]   For  $j = 1$  to  $m$  do
[10*]  If ( $M_{1j} < F_1$ ) then  $F_1 = M_{1j}$ 
      EndFor
[11]  For  $t = 2$  to  $T$  do
[12]     $F_t = 999999.0$ 
[13*]   For  $k = 1$  to  $t-1$  do
[14*]     $Sumh_k = Sumh_k + h_{k-1}$ 
[15*]    For  $j = 1$  to  $m$  do  $M_{kj} = M_{kj} + d_t(c_{jk} + Sumh_k)$ 
      EndFor
[17]   For  $j = 1$  to  $m$  do
[18]     For  $k = t-1$  down to  $0$  do
[19*]      If ( $k+1 = t$ ) then  $Aux = M_{ij}$ 
[20]      Else
[21]         $Aux = 999999.0$ 
[22]        For  $i = k+1$  to  $t$  do
[23*]          $Temp = M_{ij} + c_{ij} (D_{i-1} - d_k)$ 
[24]         For  $l = i-1$  down to  $k+1$  do
[25]            $Temp = Temp + hb_{l-1} (D_l - d_k)$ 
[26]           If ( $Temp < Aux$ ) then  $Aux = Temp$ 
        EndFor
      EndElse
[27]   If ( $Aux + F_k < F_t$ ) then  $F_t = Aux + F_k$ 
      EndFor
    EndFor
  EndFor

```

Figure 5 – An efficient implementation of the PPCA (EIPPCA).
* – adaptations of the original algorithm.

4. Computational Results

Computational tests were run to evaluate the performance of implementations PPCA and EIPPCA comparatively. The code was implemented in C++ language and run on the microcomputer described in Section 2.

For a given number of planning periods n and a given number of production centers m , 10 instances were randomly generated, considering $T = 50, 100, 250$ and 500 and $m = 2, 4, 6$. Parameters were generated as described in Section 2.

Table 2 – Results of the computational tests.

Periods	Production Centers	PPCA	EIPPCA
50	2	0.06s	0.03s
	4	0.13s	0.07s
	6	0.18s	0.11s
100	2	0.89s	0.52s
	4	1.79s	1.05s
	6	2.69s	1.57s
250	2	32.78s	19.06s
	4	1m06s	38.11s
	6	1m39s	57.20s
500	2	8m35s	5m05s
	4	17m08s	10m10s
	6	25m39s	15m15s

Table 2 shows the average running time for the implementations of the proposed algorithm. Results show that IEAMP presents better computational times. The efficient implementation of the algorithm, although not reducing its complexity, reduces the computational time approximately by 40%.

As shown above in the results, the longer the planning horizon is, the harder the problem solution becomes. For example, take problems with 1,000 binary variables, or in other words, problems with 250 periods and 4 production centers and problems with 500 periods and 2 production centers. On average, the first set of problems is solved in 38 seconds, while the second set of problems takes, on average, 5 minutes. The same is true for other sets of problems; thus, we can conclude that the planning horizon has a stronger influence on the problem solution.

5. Conclusions and Future Research

This paper studies the lot-sizing problem with one single item where backlogging is allowed. An efficient implementation of forward dynamic programming algorithm was proposed to solve the problem with only one production center, while a dynamic programming algorithm was developed for several production centers.

Results have shown that the efficient implementation of the dynamic programming algorithm has reduced computational time by 50%, although it has not reduced its complexity. The algorithm developed for problems with several production centers produced results in less than one minute for problems with 1500 binary variables. It could also be observed that the efficient implementation reduces computational time by 40% in this case, and that an increase in the number of periods over the planning horizon is more significant than an increase in the number of production centers.

Above we have only considered the adaptation of the algorithm presented in Section 2. Although, in Armentano & Toledo (1997) the authors concluded that the effect of the lower complexity algorithm can be noticed just for a number of planning periods greater than 250, we suggest an extension of Ferdergruen & Tzur (1993) and the results will be compared with the results obtained in this article.

The solution to the problem studied will allow the development of exact and heuristic algorithms for lot-sizing problems with several items, several production centers and capacity constraints, where backlogging may exist. The possibility of backlogging makes problems more similar to actual production conditions, where generally deadlines can be negotiated, whether they add costs to companies or not.

Acknowledgements

This work was supported by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo).

References

- (1) Armentano, V.A. & Toledo, F.M.B. (1997). Dynamic Programming Algorithms for the Parallel Machine Lot Sizing Problem. *Pesquisa Operacional*, **17**, 137-149.
- (2) Aggarwal, A. & Park, J.K. (1993). Improved algorithms for economic lot size problems. *Operations Research*, **41**, 549-571.
- (3) Bahl, H.C.; Ritzman, L.P. & Gupta, J.N.D. (1987). Determining lot sizes and resource requirements: a review. *Operations Research*, **35**, 329-345.
- (4) Bahl, H.C. & Taj, S. (1991). A data-dependent efficient implementation of the Wagner-Whitin algorithm for lot-sizing. *Computers and Industrial Engineering*, **20**, 289-291.
- (5) Brahim, N.; Dauzere-Peres, S.; Najid, N.M. & Nordli, A. (2006). Single item lot sizing problems. *European Journal of Operational Research*, **168**, 1-16.
- (6) Carreno, J.J. (1990). Economic lot scheduling for multiple products on parallel identical processors. *Management Science*, **36**, 348-358.
- (7) Diaby, M.; Bahl, H.C.; Karwan, M.H. & Zionts, S. (1992a). A lagrangean relaxation approach for very-large-scale capacitated lot-sizing. *Management Science*, **38**, 1329-1340.
- (8) Diaby, M.; Bahl, H.C.; Karwan, M.H. & Zionts, S. (1992b). Capacitated lot-sizing and scheduling by lagrangean relaxation. *European Journal of Operational Research*, **59**, 444-458.
- (9) Drexel, A. & Kimms, A. (1997). Lot sizing and scheduling – survey and extensions. *European Journal of Operational Research*, **99**, 221-235.
- (10) Evans, J.R. (1985). An efficient implementation of the Wagner-Whitin algorithm for dynamic lot-sizing problem. *Journal of Operational Management*, **5**, 229-235.

- (11) Federgruen, A. & Tzur, M. (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.
- (12) Federgruen, A. & Tzur, M. (1993). The dynamic lot-sizing model with backlogging: a simple $O(n \log n)$ algorithm and minimal forecast horizon procedure. *Naval Research Logistics*, **40**, 459-478.
- (13) Galil, Z. & Giancarlo, R. (1989) Speeding up dynamic programming with applications to molecular biology. *Theoretical Computer Science*, **64**, 107-118, *apud* in Federgruen & Tzur (1993).
- (14) Golany, B.; Maman, R. & Yadin, M. (1992). An efficient algorithm for the dynamic economic lot size problem. *Computers and Operations Research*, **19**, 495-504.
- (15) Johnson, L.A. & Montgomery, D.C. (1974). *Operations Research in Production Planning, Scheduling and Inventory Control*. New York, John Wiley & Sons.
- (16) Karimi, B.; Fatemi Ghomi, S.M.T. & Wilson, J.M. (2003). The capacitated lot sizing problem: a review of models and algorithms. *OMEGA*, **31**, 365-378.
- (17) Kuik, R.; Salomon, M. & van Wassenhove, L.N. (1994). Batching decisions: structure and models. *European Journal of Operational Research*, **75**, 243-263.
- (18) Lozano, S.; Larraneta, J. & Onieva, L. (1991). Primal-dual approach to the single level capacitated lot-sizing problem. *European Journal of Operational Research*, **51**, 354-366.
- (19) Pochet, Y. & Wolsey, L.A. (1988). Lot-size models with backlogging – strong reformulations and cutting planes. *Mathematical Programming*, **40**, 317-335.
- (20) Sabbag Neto, Z. (1993). Planejamento da produção em máquinas paralelas sob restrições de capacidade. Dissertação de Mestrado, FEEC – UNICAMP.
- (21) Sambasivan, M. & Yahya, S. (2005). A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research*, **32**, 537-555.
- (22) Sung, C.S. (1986). A single-product parallel-facilities production-planning model. *International Journal of Systems Science*, **17**, 983-989.
- (23) Toledo, F.M.B. (1998). Dimensionamento de lotes em Máquinas Paralelas. Tese de Doutorado, FEEC – UNICAMP.
- (24) Trigeiro, W.W.; Thomas, L.J. & McClain, J.O. (1989). Capacitated lot sizing with setup times. *Management Science*, **35**, 353-366.
- (25) Wagelmans, A.; van Hoesel, S. & Kolen, A. (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.
- (26) Wagner, H.M. & Whitin, T.M. (1958). Dynamic version of the economic lot size model. *Management Science*, **5**, 89-96.
- (27) Wilber, R.E. (1988). The concave least weight subsequence problem revisited. *Journal of Algorithms*, **9**, 418-425, *apud* in Federgruen & Tzur (1993).

- (28) Wolsey, L.A. (1995). Progress with single-item lot-sizing. *European Journal of Operational Research*, **86**, 395-401.
- (29) Wolsey, L.A. (2002). Solving Multi-Item Lot-Sizing Problems with an MIP Solver Using Classification and Reformulation. *Management Science*, **48**, 1587-1602.
- (30) Zangwill, W.I. (1969). A backloging model and a multi-echelon model of a dynamic economic lot size production system – a network approach. *Management Science*, **15**, 506-527.