Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

METAHEURISTIC APPROACHES FOR THE MULTILEVEL RESOURCE-CONSTRAINED LOT-SIZING PROBLEM WITH SETUP AND LEAD TIMES

REGINA E. BERRETA
PAULO M. FRANÇA
VINÍCIUS A. ARMENTANO

Nº 89

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Mai./1999

METAHEURISTIC APPROACHES FOR THE MULTILEVEL RESOURCE-CONSTRAINED LOT-SIZING PROBLEM WITH SETUP AND LEAD TIMES

Regina E. Berretta[1] • **Paulo M. França**[2] • **Vinícius A. Armentano**[2]

[1] Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo

CEP 13560-970, São Carlos, SP, Brazil

[2]Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas

CEP13083-970, Campinas, SP, Brazil

META-HEURÍSTICAS PARA O PROBLEMA DE DIMENSIONAMENTO DE LOTES MULTIESTÁGIO COM RECURSOS LIMITADOS, LEAD TIMES E TEMPOS DE PREPARAÇÃO

RESUMO

Esse artigo propõem o uso de meta-heurísticas para o problema de dimensionamento de lotes em estrutura de produto do tipo geral, custos e tempos de preparação e lead times. Inicialmente é proposta uma heurística baseada em transferências de partes da produção entre períodos com objetivo de encontrar uma solução factível. Estratégias de memória de curto e longo prazo da meta-heurística Busca Tabu é incluída no sentido de guiar a busca a obter novas e melhores soluções factíveis. Componentes de Simulated Annealing são incorporados com o propósito de melhorar o desempenho da heurística. Para exemplos pequenos, as soluções obtidas pela heurística híbrida com Busca Tabu e Simulated Annealing são comparadas com as soluções ótimas. Para exemplos grandes, a qualidade da solução é avaliada através de um limitante inferior gerado pela aplicação de Relaxação Lagrangiana.

Palavras-chave: dimensionamento de lotes, heurísticas, meta-heurísticas, Busca Tabu, Simulated Annealing;

METAHEURISTIC APPROACHES FOR THE MULTILEVEL RESOURCE-CONSTRAINED LOT-SIZING PROBLEM WITH SETUP AND LEAD TIMES

ABSTRACT

We propose the use of metaheuristic for the resource-capacitated multilevel problem with general product structures, setup costs, setup times and lead times. Initially, we develop a heuristic which moves production in time in order to obtain feasible solutions with good quality. Strategies for the short term memory and long term memory of tabu search are then included to guide the search of the subordinate heuristic for new, feasible and better solutions. Simulated Annealing components are embedded into tabu search in order to improve its performance. For small problems, the solutions provided by tabu search and the hybrid metaheuristic are compared to optimal solutions and for larger problems, the quality of the solutions is evaluated against a lower bound generated by Lagrangean Relaxation.
Key-words: lot-sizing, heuristics, metaheuristics, Tabu Search, Simulated Annealing;

1. INTRODUCTION

This article considers the multilevel lot-sizing problem for general product structures and multiple capacity-constrained resources. Component safety lead times, setup costs and setup times are also taken into account. The determination of how much to produce of each component in each time period, in a synchronized way, arises in one of the planning steps in Materials Requirement Planning (MRP) systems. It is well known that most computerized MRP systems ignore capacity constraints and disregard setup, production and inventory costs when deciding lot sizes. As a result, the production plans generated by such systems are likely to be capacity infeasible and uneconomical. The aim of this research is to contribute to the development of more realistic MRP systems.

The inclusion of safety lead time makes the solution method more complex due to the need of synchronizing the planning periods of the components. The notion of safety lead time, as discussed in Billington et al. (1986), is the unavoidable time from the moment a component is produced until it is available for usage. For instance, this could be the time for paint to dry or hot metal to cool. It is important to emphasize that during this time there is no resource consumption.

When setup times are considered, the decision problem associated with the existence of a feasible solution for this problem is NP-complete (Maes et al. 1991). The problem addressed here is NP-hard since it is a generalization of the single-item, single-level, capacitated lot-sizing problem which is NP-hard (Florian et al. 1980).

There is a vast literature on lot-sizing which has been reviewed by Bahl et al. (1987) and Kuik et al. (1994). Harrison and Lewis (1996) provide a survey including optimal and heuristic methods for multilevel, multi-item lot-sizing problems. Katok et al. (1998) present a review of heuristic methods and classify them into four categories: decomposition approaches, Lagrangean relaxation schemes, local search methods and LP-based heuristics.

Restricting the review to the literature on multilevel systems with setup times and finite resources, to the best of our knowledge few papers address this subject. Two articles deal with optimization approaches. Pochet and Wolsey (1991) address several single-level and multilevel problems and solve them using strong cutting planes generated by a mathematical programming software package. More recently, Clark and Armentano (1995b) employed a solution method that identifies valid inequalities that increase the lower bound provided by the formulation's linear programming relaxation. The stronger lower bound is then used in a branch and bound algorithm.

The first heuristic developed for this class of lot-sizing problems was suggested by Billington et al. (1986). They present a heuristic method based on Lagrangean relaxation which is embedded in a branch and bound procedure. A smoothing routine attempts to reach capacity feasibility by moving production amounts between periods. A similar approach is proposed by Tempelmeier and Derstroff (1996). As opposed to Billington et al., who solved the Lagrangean problem heuristically, they show that the Lagrangean problem can be decomposed into a series of subproblems, one for each item, solvable by the Wagner-Whitin (1958) algorithm. Their smoothing procedures are more elaborated and the performance of the method is evaluated in larger problems.

Maes et al. (1991) suggest several heuristics based on rounding the setup variable values obtained by LP-relaxation of the original problem. Another approach using LP-relaxation restricted to serial systems is proposed by Harrison and Lewis (1996). The main idea is to modify the coefficients of certain variables of the capacity constraints to implicitly account for the resources consumed by setups. Katok et al. (1998) extend this approach to deal with setup costs and general product structure.

Clark and Armentano (1995a) present a heuristic method for general structures with lead times that starts from an infeasible solution and attempts to reach feasibility by employing a smoothing procedure. França et al. (1997) extend the previous method by adding new types of moves to achieve feasibility and an improvement procedure. Better results were obtained with respect to the percentage of feasible solutions found, as well as to their quality. In Armentano et al. (1998), problems with serial structures and non-zero lead times are studied. A new production shift procedure, which explores the dependency among components in the serial structure, has led to significant improvements with regard to finding feasible solutions.

Metaheuristic approaches using simulated annealing and tabu search are proposed by Kuik et al. (1993). Computational results have shown that such approaches compare favorably with respect to LP-relaxation.

This article deals with the multilevel capacitated lot-sizing problem - MLCLSP - with setup costs, setup times and lead times. We propose a heuristic method based on production shifts to obtain capacity feasibility and improved solutions. A hybrid metaheuristic, which combines tabu search and concepts of simulated annealing, is proposed.

2. PROBLEM FORMULATION

In this section, the MLCSLP is formulated as a mixed-integer programming model using the concept of echelon stock (Clark and Scarf 1960; Afentakis et al. 1984) in a rolling horizon basis. Consider the following mathematical notation:

Variables:

x_{it} lot-size of component i in period t.

y_{it} 1, if component i is produced in period t; 0, otherwise.

I_{it} inventory stock of component i at the end of period t

E_{it} level of echelon stock of component i at the end of period t.

Parameters:

N number of components.

T number of planning periods.

K number of resources.

S(i) the set of immediate successors of component i; S(i)=∅ if i is an end item.

P(i) the set of immediate predecessors of component i.

c_{it} unit production cost of component i in period t.

s_{it} setup cost incurred if component i is produced in period t.

e_{it} unit cost of echelon stock of component i at the end of period t.

v_{ikt} unit amount of resource k used in the production of component i in period t.

f_{ikt} fixed amount of resource k used in the setup of component i in period t.

b_{kt} amount of resource k available in period t.

U_{it} upper bound for x_{it}.

D_{it} echelon demand of component i in period t.

r_{ij} number of components i needed by one unit of the successor component j.

L(i) lead time of component i, ensuring that the lot x_{it} is available for consumption only at the beginning of period t+L(i).

T(i)+1 the period at which, on a rolling horizon of T periods, production planning of component i must start in order to satisfy demand for its immediate successor at periods $t \geq T(i)+1+L(i)$ and eventually with that of the end-item. The lot x_{it} belongs to one of the following three categories: (1) if $t \leq T(i)$ then x_{it} is known from previous rolling horizon applications of the model and has a fixed value; (2) if $T(i)+1 \leq t \leq T(i)+T$ then x_{it} is a model decision variable; (3) if $t > T(i)+T$ then x_{it} is beyond the planning horizon and will enter the model only in a future rolling horizon application. Let $T_{max} = \max\{T(i) \mid i=1,\ldots,N\}$.

In order to illustrate the concept of T(i), consider the following example of a general product structure in Figure 1.

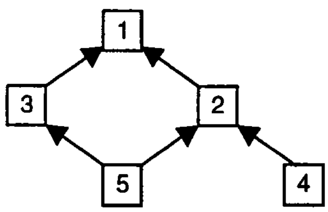


Figure 1. Example of a general product structure

Assume that the lead times of the components are L(1)=L(5)=0, L(2)=L(3)=L(4)=1. It follows that T(1)=2, T(2)=T(3)=T(5)=1, T(4)=0. Further details about T(i) and how it is determined can be found in Clark and Armentano (1993).

Echelon stock is defined as:

$$E_{it}=I_{i,L(i)+t}+\sum_{j\in S(i)} r_{ij} E_{j,L(i)+t}$$

In words, E_{it} is the total system stock of component i at the end of period t, both as a stand-alone component, $I_{i,L(i)+t}$, and as part of the successor components $r_{ij}E_{j,L(i)+t}$, $j\in S(i)$.

Similarly, echelon demand is defined as the total demand of a component, i.e. its own independent demand plus the echelon demand of the successor components:

$$D_{it}=d_{i,L(i)+t}+\sum_{j\in S(i)} r_{i,j} D_{j,L(i)+t}$$

The following mathematical model has been proposed by Clark and Armentano (1993).

$$\textbf{(ME):}\ \text{Minimize} \sum_{i=1}^{N} \sum_{t=T(i)+1}^{T(i)+T} [s_{it}\, y_{it} + c_{it}\, x_{it} + e_{it}\, E_{it}] + \sum_{i|P(i)\neq\varnothing}^{N} \sum_{t=1}^{T(i)} e_{it}\, E_{it} \qquad (1)$$

subject to

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad i = 1, ..., N\ ;\ t = 1+T(i), ..., T+T(i) \qquad (2)$$

$$\sum_{j\in S(i)} r_{ij} E_{j,t+L(i)} \leq E_{it} \qquad i = 1, ..., N\ ;\ t = 1+T(i), ..., T+T(i) \qquad (3)$$

$$\sum_{i|t\leq T(i)+T} [f_{ikt}\, y_{it} + v_{ikt}\, x_{it}] \leq b_{kt} \qquad k = 1, ..., K\ ;\ t = 1, ..., T + T_{max} \qquad (4)$$

$$x_{it} \leq U_{it} y_{it} \qquad i = 1, ..., N\ ;\ t = 1+T(i), ..., T+T(i) \qquad (5)$$

$$x_{it} \geq 0, E_{it} \geq 0, y_{it} \in \{0,1\} \qquad i = 1, ..., N\ ;\ t = 1+T(i), ..., T+T(i) \qquad (6)$$

The objective function consists of the minimization of setup, production, and inventory costs involving only the decision periods of each component. Due to the fact that the model is applied on a rolling horizon basis, the variables are those between T(i)+1 to T(i)+T. The values of x_{it}, y_{it}, and E_{it} for $t \leq T(i)$ are fixed, as they were decided in the previous application of the model. It is shown (Clark and Armentano 1993) that the constant term $\sum_{i|P(i)\neq\varnothing} \sum_{t=1}^{T(i)} e_{it}\, E_{it}$ results from the equivalence between models using conventional and echelon stocks. Constraints (2) represent the inventory balance equations. The inter-echelon constraints (3) follow from the definition of echelon stock and the non-negativity of the conventional stock I_{it}. The capacity constraints in (4) state that the fixed and variable amounts of resources used for production are limited to the resource available in period t. Constraints (5) provide an upper bound for x_{it}, if $y_{it} = 1$, and ensure that $x_{it}=0$, if $y_{it}=0$. It can be easily shown that $E_{it} \geq 0$ in (6) is redundant and follows from (3).

A lower bound on the optimal solution value for model ME can be easily obtained by Lagrangean relaxation of the inter-echelon stock constraints (3) and the capacity constraints (4). The resulting Lagrangean problem decomposes into a set of N independent, uncapacitated, single item lot-sizing problems solvable by the Wagner-Within algorithm. The subgradient method of Camerini et al. (1975) is then used to optimize the dual function.

3. THE HEURISTIC METHOD

The method consists of four procedures, which interact according to the diagram shown in Figure 2.

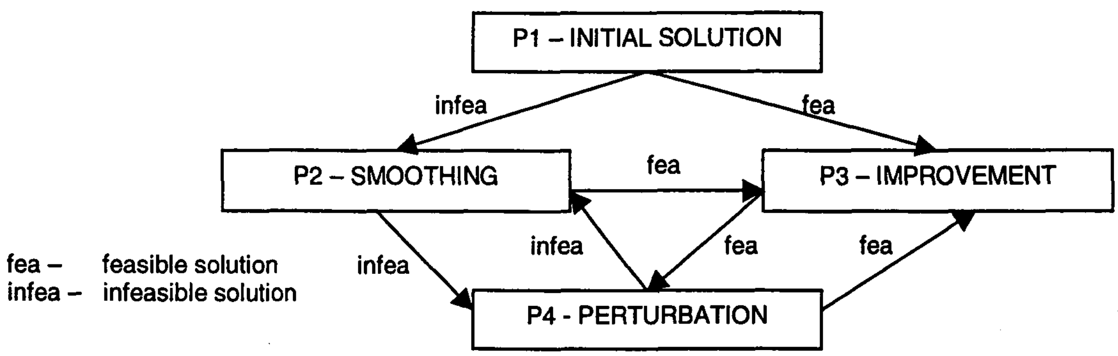


Figure 2. Diagram with the heuristic procedures

Initially, procedure P1 is applied to the conventional inventory stock model (Katok et al., 1998, Tempelmeier and Derstroff, 1996 and Clark and Armentano, 1995) without capacity constraints by applying the optimal Wagner-Whitin algorithm to each component, starting from the final-items, and then applying it to their predecessors. As the solution found in P1 is likely to be capacity infeasible, the smoothing procedure P2 tries to find a feasible solution by shifting production amounts between time periods. Starting from the solution found in P2 or a possible feasible solution yielded in P1, the improvement procedure P3 searches for a lower cost solution. The improved solution from P3 is then perturbed in order to direct the search to new regions of the solution space. In case procedure P2 fails to produce a feasible solution, procedure P4 is also activated. If the perturbed solution turns out to be feasible then an attempt is made to improve it. Otherwise, procedure P2 tries to render it feasible. In the following, the procedures P2, P3 and P4 are described in detail.

3.1. Smoothing Procedure P2

Initially calculate

$$\Delta(t)=\sum_{k=1}^{K}\left(\sum_{i/t\leq T(i)+T}[f_{ikt}\,y_{it}+v_{ikt}\,x_{it}]-b_{kt}\right)\Big/b_{kt}\,. \qquad (7)$$

Then Excess(t) = max{0, Δ(t)} represents the capacity exceeded in period t. Procedure P2 consists of production shifts from periods with Excess(t) > 0 to other periods with the objective of finding a feasible solution. A shift can be represented by (q,i,t,τ), which means that a quantity q of component i is moved from period t to period τ.

Procedure P2 is divided into a backward step in time, where periods $t=T_{max}+T,\dots,2$ are analyzed in that order with τ < t, and a forward step where periods are analyzed in the order $t = 1,2,\dots,T_{max}+T-1$ with τ > t. In each of these steps and at each period where Excess(t) > 0, the procedure analyzes shifts (q,i,t,τ) for those components i which are produced in period t. The periods τ are those between t and t', where

$$t'=\begin{cases}\max\{1,\text{last period in which there is production of component } i\} & \text{if } \tau<t\\ \min\{T_{max}+T,\text{first period in which there is production of component } i\} & \text{if } \tau>t\end{cases}$$

The amount of the component to be shifted is determined in such a way as to satisfy constraints (2) and (3). For each component i that is produced in an infeasible period t, first determine $M_{i,t,\tau}$ which represents the maximum amount of component i that can be moved from period t to τ in such a way that the inter-echelon feasibility is preserved. Whether there is an amount Q_{itk} smaller than $M_{i,t,\tau}$ that reduces Excess(t) to zero for a given resource is also checked. The choice between these two quantities is based on a cost index $R(q,i,t,\tau)$ calculated as a function of the variation of the total cost and of the overuse of resources caused by the shift per unit of overload eliminated.

The calculation of $M_{i,t,\tau}$, Q_{itk} and $R(q,i,t,\tau)$ are detailed below:

- **$M_{i,t,\tau}$ for $\tau > t$**

From constraints (2), it follows that a shift (q,i,t,τ) with $\tau > t$ causes the echelon stock level of component i in periods t to τ-1 to be reduced by q. From (3), we can write

$$\sum_{j\in S(i)} r_{ij} E_{j,\delta+L(i)} \leq E_{i\delta} - q \qquad \delta = t,\ldots,\tau-1$$

Thus,

$$M_{i,t,\tau} = \min_{\delta=t,\ldots,\tau-1} \{E_{i\delta} - \sum_{j\in S(i)} r_{ij} E_{j,\delta+L(i)},\ x_{it}\} \tag{8}$$

- **$M_{i,t,\tau}$ for $\tau < t$**

Similarly, constraints (2) show that a shift (q,i,t,τ) with $\tau < t$ adds by q the echelon stock of component i in periods τ to t-1. In order to ensure that constraints (3) are satisfied after the shift, we must have

$$\sum_{m\in S(j)} r_{jm} E_{m,\delta} + r_{ji} q \leq E_{j,\delta-L(j)} \qquad j\in P(i),\ \delta = \tau,\ldots,t-1$$

Thus,

$$M_{i,t,\tau} = \min_{\substack{\delta=\tau,\ldots,t-1 \\ j\in P(i) \\ \delta-L(j)\leq T(j)+T}} \left\{ \frac{E_{j,\delta-L(j)} - \sum_{m\in S(j)} r_{jm} E_{m\delta}}{r_{ji}},\ x_{it} \right\} \tag{9}$$

- **$Q_{i,t,k}$**

The quantity that reduces the overload of resource k in a given period t can be expressed by

$$Q_{i,t,k} = \left(\sum_{j/t\leq T(j)+T} \left(f_{jkt} y_{jt} + v_{jkt} x_{jt} \right) - b_{kt} \right)^{+} / v_{ikt}, \quad \text{if } Q_{i,t,k} < M_{i,t,\tau}$$

where $a^{+} = \max\{0,a\}$

- **R(q,i,t,τ)**

As mentioned earlier, the choice of the amount to be shifted is based on the calculation of the cost index R(q,i,t,τ) for each one of the candidate shifts (q,i,t,τ).

$$R(q,i,t,\tau)=\frac{\Delta\text{cost}+\beta.\text{Penalty}}{\text{Excess\_decrease}} \tag{10}$$

Δcost is the ratio Additional_cost/Total_cost, where Additional_cost is the cost change due to the shift (q,i,t,τ) and Total_cost corresponds to the current production and inventory cost for all components. The expression for Additional_cost is as follows

$$\text{Additional\_cost}=su_1-su_2+q\left[(c_{i\tau}-c_{it})+\begin{cases}\sum\limits_{\delta=\tau,\dots,t-1}e_{i\delta} & \text{if } \tau<t\\ \sum\limits_{\delta=t,\dots,\tau-1}-e_{i\delta} & \text{if } \tau>t\end{cases}\right]$$

where,

$$su_1=\begin{cases}s_{i\tau} & \text{if } x_{i\tau}=0\\ 0 & \text{otherwise}\end{cases}\qquad su_2=\begin{cases}s_{it} & \text{if } q=x_{it}\\ 0 & \text{otherwise}\end{cases}$$

The Penalty term is a non-negative quantity which can be interpreted as a cost for overuse of resources in periods t and τ and β is a control parameter. Penalty is defined as

$$\text{Penalty}=\text{Excess\_after(t)}+[\text{Excess\_after}(\tau)-\text{Excess\_before}(\tau)]$$

where,

$\text{Excess\_before}(t)=\text{Excess}(t)$ before the move.

$\text{Excess\_after}(t)=\text{Excess}(t)$ after the move.

The cost index denominator Excess_decrease is given by

Excess_decrease = Excess_before(t) - Excess_after(t)

By cycle we mean a sequence of a backward and a forward step. During the first cycle β =1, and if a feasible solution is not found, β is increased by 1 in the subsequent cycles. In this way, the increase of parameter β in the smoothing procedure gives greater importance to the overuse of resources. If no feasible solution is found after a prespecified number of cycles, the procedure fails.

The shift (q,i,t,τ) that yields the smallest value of $R(q,i,t,\tau)$ is performed and new shifts are executed while Excess(t) is greater than zero.

For a given infeasible period t, it may occur that no shifts are allowed because $M_{i,t,\tau}=0$, for any i and τ. In this case, procedure P2 calls a new routine which tries to perform specific shifts in periods other than t. This routine, detailed in Armentano et al. (1998), tries to alter the echelon stock of other components in specific periods so that $M_{i,t,\tau}>0$ for some i and τ.

The backward and forward steps are performed consecutively until a feasible solution has been found or a stopping criterion has been reached.

3.2. Improvement Procedure P3

Starting from a feasible solution yielded by P1, P2 or P4, procedure P3 is applied in an attempt to find a lower cost solution. This procedure, also employing the backward and forward steps, is similar to procedure P2. However, we only allow feasible and improving moves, i.e. shifts to periods with capacity slack.

For each period t we consider moving a quantity q of the production x_{it} of each component i to a period τ. For each i and τ, we examine two quantities in period t: $q = F_{i,t,\tau}$ and q sampled from the uniform distribution $U[1, F_{i,t,\tau}]$, where

$$F_{i,t,\tau} = \min\left\{M_{i,t,\tau}, \min_{k=1,\dots,K}\left(G_{i,k,\tau}\right)\right\}$$

and $G_{i,k,\tau}$ = maximum quantity of item i that can be moved from period t to period τ so as not to violate the amount of resource k available in period τ.

Computational tests have shown that the inclusion of this random quantity leads to better solutions. Another reason for considering this random quantity is to avoid cycling which always occurs when procedure P3 is initiated from the same solution. Among all candidate shifts (q,i,τ) in a given period t, we choose the one which minimizes the ratio Δcost as defined in (10). As the procedure searches for improving solutions, only negative values of

Δcost are accepted. The backward and forward steps are performed consecutively until no improving solution is found.

3.3. Perturbation Procedure P4

In this procedure, for each component we transfer a quantity $q \in U[1, M_{i,t',\tau'}]$ from the most loaded period t' to the least one τ'. The aim of such a shift is twofold. If the solution from P2 is infeasible, then this perturbation is a crude attempt to obtain a feasible solution or at least to reduce the overuse of resources in period t'. If, however, the solution from P2 turns out to be feasible then this perturbation is likely to produce an infeasible solution without a large overuse of resources in period τ'.

For each period t we compute $-\Delta(t)$ which, if positive, represents the capacity slack in t. The periods are then arranged in decreasing order of $-\Delta(t)$ and for each component we search the most loaded period t' and the least loaded period τ' such that $M_{i,t',\tau'} > 0$. After the shift of one component the quantities $-\Delta(t')$ and $-\Delta(\tau')$ are recalculated and the periods are rearranged in decreasing order of $-\Delta(t)$.

4. INCLUSION OF METAHEURISTICS

Although the perturbation mechanism introduced by procedure P4 extends the ability of the method in producing good quality solutions, there are circumstances in which the method may lead to unsatisfactory production plans. When the output of procedure P2 is an infeasible solution, say IS, procedure P4 performs a small perturbation on this solution in an attempt to make it feasible. If the solution turns out to be infeasible, procedure P2 is called again and it may happen that it produces the same solution IS or an infeasible solution "close" to IS in such a way that procedure P2 is unable again to produce a feasible solution. An analogous situation occurs when procedure P2 yields a feasible solution, say FS. Assume that starting from FS, the improvement procedure P3 is able to produce a better solution. The perturbation on such a solution performed in P4 may lead to an infeasible solution so that procedure P2 yields the same solution FS or a feasible solution "close" to FS which in turn leads procedure P3 to produce the same or a "close" solution.

The performance of the heuristic, in terms of finding a feasible solution as well as high quality solutions, can be improved by guiding it through a master process called metaheuristic. Since the heuristic procedures consist of production shifts, we chose to use metaheuristics based on local search such as tabu search (Glover and Laguna, 1998) and simulated annealing (Kirkpatrick et al., 1983). Initially, tabu search with short and long term memory strategies is implemented. Next, concepts of simulated annealing are incorporated into tabu search in an attempt to obtain better solutions.

4.1.Tabu Search

Tabu search is a metaheuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. The local procedure is a search that uses an operation called *move* to define the neighborhood of any solution. Tabu search reduces the neighborhood by classifying certain moves as forbidden or tabu. This mechanism prevents cycling, i.e., the indefinite execution of the same sequence of moves, and directs the search to unexplored regions. An aspiration criterion removes the tabu condition of a move if it is considered attractive at that moment of the search.

One of the main features of tabu search is the use of adaptive memory to store key elements of the history of the search. The short term memory keeps track of solution attributes that have changed during the recent past. This is the kind of memory encountered in most applications of tabu search in the literature. However, in general, higher quality solutions are produced when using long term memory associated to diversification and intensification strategies in order to explore new regions of the solution space and exploit some attractive regions.

In the sequel, we identify the components for the short term memory and the long term memory of the tabu search method applied to our problem.

- **Short Term Memory**

Neighborhood structure. It is defined by moves (q,i,t,τ) such that $M_{i,t,\tau} > 0$, which represent the transference of the quantity q of component i from period t to period τ. The choice of a move depends on the heuristic procedure as explained above.

Solution attribute and tabu activation rule. The solution attribute is represented by (i,t) which means that component i is produced in period t. Several rules were tested in order to assign the tabu status to certain moves and the best one corresponds to forbidding the move of any amount of component i from period t for a certain number of production shifts.

Tabu tenure. The number of production shifts that a move, characterized by (i,t), remains tabu is selected from an interval [a, b] with a uniform distribution. A new value is chosen after each shift. This type of tabu tenure is found in many applications of tabu search and is motivated by the fact that a fixed tabu tenure, in general, does not allow a balance between exploitation of a region and exploration of new regions. Fixed ranges such as [10, 30] and ranges which depend on the problem size were tested but none turned out to be satisfactory for all problems. Good results were achieved from the number of shifts performed during the first execution of procedures P2, P3 and P4, denoted **sh**, and after extensive tests the range was defined as [sh, 4sh].

The data structure used for the short term memory is a matrix where its element (i,t) stores the last production shift in which production amount of component i is forbidden to be moved from period t. Suppose that n is the number of the current shift involving the move (q, i, t, τ). Then tabu tenure $[i,t] = n + r$, where $r \in [sh, 4sh]$.

Aspiration criterion. Computational tests revealed that the neighborhood of a solution is small because production shifts cannot violate the nonnegativity of the echelon stock. In addition, it was not uncommon to have a neighborhood composed only of tabu moves. Due to this kind of restriction, the standard aspiration criterion of removing the tabu status of a move if it results in a better solution than the incumbent, did not work well. Better results were achieved by applying an aspiration criterion only when all moves are tabu, in which case we choose the move which yields the least increase in the objective function.

- **Long Term Memory**

The diversification and intensification strategies are based on the residence frequency of two sets of solutions, one for each strategy. Diversification and intensification are achieved by a restart from procedure P1 where the setup costs are altered in such a way as to reduce or increase the likelihood of occurrence of component setups in certain periods. For this we construct two residence matrices of dimension NxT_{max} where the numerator of the element (i,t) represents the number of times that component i is produced in period t ($y_{it} = 1$) in a given set of solutions and the denominator is the largest element of the matrix. Figure 3 illustrates the setup matrix of four solutions and shows the corresponding residence matrix.

Solution 1

1	0	0
1	1	0

Solution 2

1	1	0
1	1	0

Solution 3

1	0	1
1	1	1

Solution 4

1	1	1
1	1	0

Residence Matrix

4/4	2/4	2/4
4/4	4/4	1/4

Figure 3. Example of four setup matrices and the corresponding residence matrix.

The elements of this matrix are normalized in the range [1,10] as follows. For diversification, the largest matrix element is assigned the value 10, the smallest element is assigned the value 1 and the remaining elements are linearly interpolated. For intensification, the largest and smallest elements are assigned the values 1 and 10, respectively. Figure 4 shows the normalized diversification matrix (M2) and the normalized intensification matrix (M3) associated to the residence matrix of Figure 3.

M2		
10	4	4
10	10	1

M3		
1	7	7
1	1	10

Figure 4. Normalized matrices.

The setup costs are multiplied by the elements of the normalized residence matrix. For diversification, for example, high setup costs are assigned to those binary variables that very frequently take on the value 1, thereby reducing its likelihood of assuming the value 1 in a Wagner-Whitin solution.

The diversification and intensification matrices are denoted by R_b and R_e, respectively. Matrix R_b uses the solutions produced by procedure P1. In this case, the aim is to obtain different initial solutions. In order to build R_e, we use the 20 best solutions produced by procedure P3. The objective here is to obtain an initial solution with a setup configuration which is "close" to the best solutions.

In order to explain the alternation of diversification and intensification in the heuristic, we define an *iteration* as a sequence of procedures P2, P3 and P4, in case P2 succeeds or P2 and P4, when P2 fails. A new restart occurs after 5 *iterations* (q_{it}) without updating the incumbent solution. The diversification and intensification strategies alternate every 5 restarts without updating the incumbent solution.

4.2. Simulated Annealing-based Strategy

Simulated annealing is a random metaheuristic in which an improving solution is always accepted and a non-improving one is accepted according to an acceptance probability. Martin and Otto (1994) propose a combination of local search methods with simulated annealing for minimizing a given function f. Let S_o be a local optimum and suppose that it is perturbed so that the local search is reinitiated from another point until it reaches a new local optimum S_o'. The simulated annealing acceptance function is then applied to $\Delta f = f(S_o')-f(S_o)$. If S_o' is accepted, the local search proceeds from S_o', otherwise, it continues from S_o.

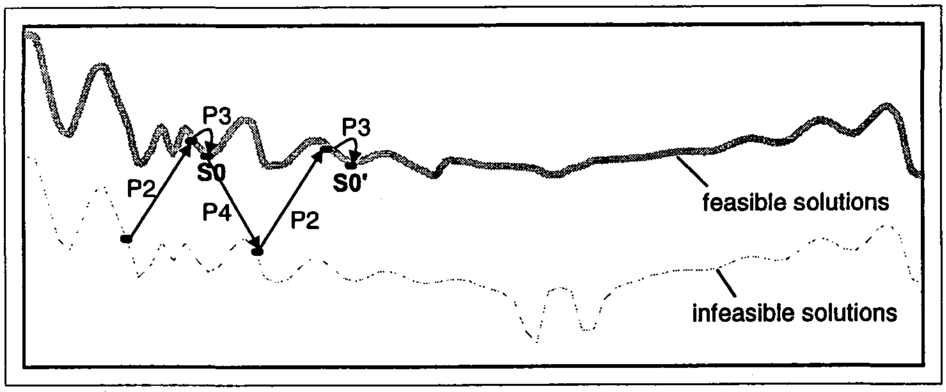


Figure 5. Simulated Annealing applied to local optima.

This idea is used in our method in the following way, see Figure 5. A random perturbation, as defined in procedure P4, is applied to a local optimum obtained in procedure P3, leading most of the times to an infeasible solution. Procedure P2 is then applied to this solution and if it succeeds in finding a feasible solution, a new local optimum is produced by P3 which most likely is different from the previous one due to a random amount of production shifts performed in procedures P3. The acceptance function is then applied to the new local optimum.

In order to specify a cooling schedule, we need to define an initial temperature T_1 and the temperature T_i associated with the acceptance of the $i^{\underline{th}}$ new local optimum S_i. Good results were obtained by setting T_i as function of the variation of the local optimal costs, namely

$$T_1 = f(S_1) - f(S_0),$$

where S_0 is the first local optimum and

$$T_i = \left[T_{i-1} + \left|f(S_i) - f(S_{i-1})\right|\right]/2$$

Regarding the acceptance function, computational tests have shown that the deterministic acceptance function proposed by Moscato and Fontanari (1990) performs better than the classical probabilistic acceptance function $e^{-\Delta f/T}$. The deterministic function F is given by F = 0 if $\Delta f > T$; 1, otherwise. The diagram in Figure 6 illustrates the functioning of the complete heuristic.

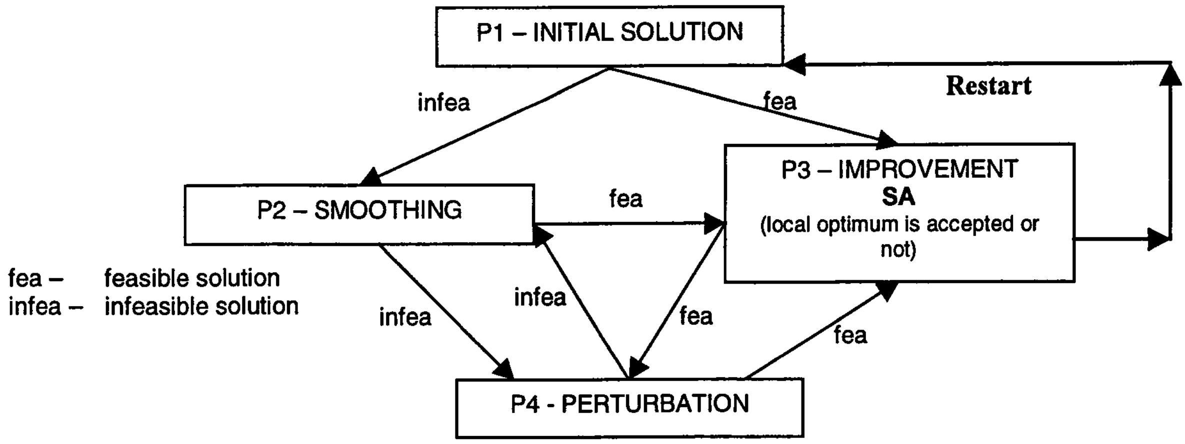


Figure 6. Diagram of the complete heuristic.

5. COMPUTATIONAL RESULTS

The performance of the heuristic methods proposed in this article is analyzed using two different groups of randomly generated test problems. The first group (G1) contains problems used to make comparisons of the heuristics with a lower bound provided by Lagrangean relaxation and subgradient optimization applied to model ME. The other group (G2) is composed of small-scale problems used to evaluate the performance of the heuristics relative to optimal solutions obtained by CPLEX. We also present a comparative study showing the behavior of the heuristic methods with and without the addition of metaheuristcs in order to emphasize the role they play in the achievement of the quality of the solution. The tests end with the application of the CPLEX LP-solver to the binary solution yielded by the heuristics. The goal of this test is to estimate the suboptimality degree attained by the heuristic solutions with respect to the binary variables and continuous variables separately.

5.1. Test Problems Generation

Different problem characteristics were simulated by means of three control factors: problem dimension, product structure and setup cost pattern. Table 1 summarizes the features of groups G1 and G2.

Table 1. Characteristics of the test problems

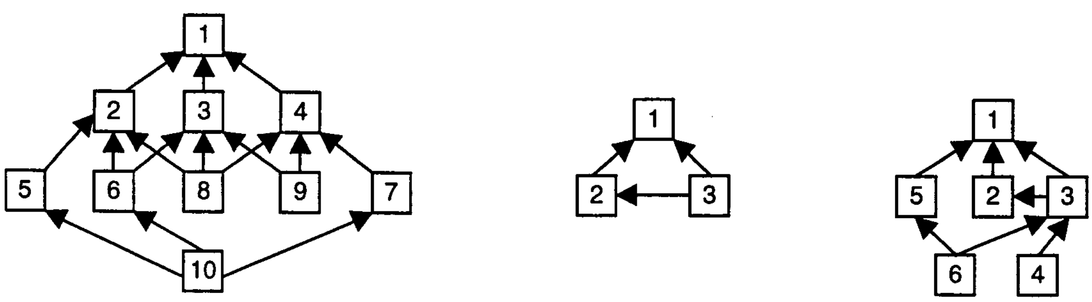
	Group G1	Group G2
product structures	flat, serial and general	flat, serial and general
N x T	10x12, 10x18, 20x12, 20x18, 40x12 and 40x18	3x6, 6x6, 10x6 and 3x12
setup cost	low and high	low and high
number of replications	10	5
Total	360	120

The data used to generate the problems come from uniform distributions as shown in Table 2. The ranges are the same used in Clark and Armentano (1995a).

Table 2. Parameters used to generate test problems

Parameter		Interval
unit production cost	c_{it}	U[1.5, 2]
low setup cost	s_{it}	U[5, 95]
high setup cost		U[50, 950]
unit cost of echelon stock	e_{it}	U[0.2, 0.4]
unit amount of resource k	v_{ikt}	U[200, 300]
fixed amount of resource k	f_{ikt}	U[2, 3]
demand for final items	d_{it}	U[0, 180]
demand for non-final items		U[0, 18]

The number of resources K used in all problems and the parameter r_{ij} were chosen equal to 1, and the lead times L(i) were randomly selected as 0 or 1. Three types of structures are used: flat, serial and general. A flat product structure is a two-level assembly structure. The general product structures used in the test problems are shown in Figure 7. The 20-item general structure was replaced the 17-item general structure that was taken from Maes et al. (1991). The 40-item structure was taken from Clark and Armentano (1995a).



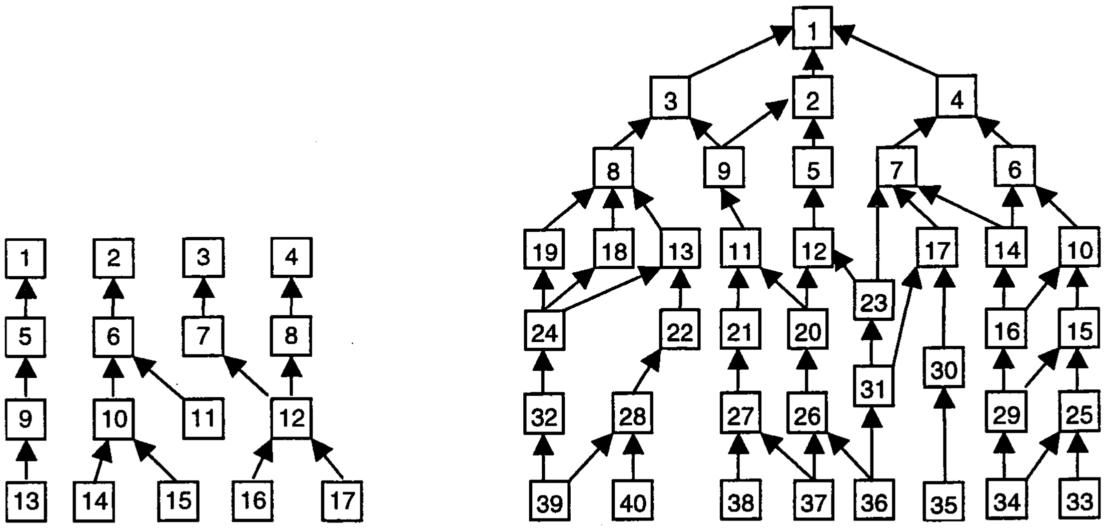


Figure 7. General product structures used in the test problems

The capacities were generated in the following way. After obtaining the lot-for-lot solution - the production in each period t is equal to its corresponding demand - the capacities are calculated in such a way that the availability of a resource k is the same in all periods. However, due to the existence of lead times, the capacity in a given period t should be proportional to the number of components such that t is a planning period. Formally, we have:

$$b_{kt} = \frac{\sum_{\tau=1}^{T+T_{max}} \Delta_{k\tau}}{T+T_{max}} \cdot \frac{Q_t}{N} \qquad \begin{array}{l} t = 1,\dots,T+T_{max} \\ k = 1,\dots,K \end{array}$$

where $\Delta_{k\tau} = \sum_{i/\tau \le T(i)+T} \begin{cases} f_{ik\tau} + v_{ik\tau} D_{i\tau} & \text{if } D_{i\tau} \neq 0 \\ 0 & \text{c.c.} \end{cases}$

Q_t = quantity of items i in period t / $t \le T(i)+T$.

The stopping criterion for all heuristic versions which execute the 4 procedures is N*T seconds. Computational experiments have shown that using more than 6 cycles does not significantly improve the ability of P2 to find feasible solutions and therefore the maximum number of cycles allowed was 6.

5.2. Computational Results without Metaheuristics

First, we evaluate some heuristics without the addition of metaheuristics. The aim here is to construct a reference which allows the analysis of the importance of the inclusion of metaheuristics. We also compare the proposed heuristics to the method of Clark and Armentano (1995a). In this part of the computational experiments, the test problem group used was G1. Table 3 shows the four heuristic versions used in comparisons.

Table 3. Description of the heuristics used in comparisons

HP1	Heuristic composed by P1 (only the initial solution obtained by our heuristic)
HCA	Heuristic by Clark and Armentano (1995a)
HP2	Heuristic composed by P1 and P2
HP4	Heuristic composed by P1, P2, P3 and P4

The heuristics' performance are evaluated with respect to three indexes calculated for each heuristic solution:

EXC: Average percentage of overuse of the resources, given by

$$100\left(\sum_{t=1}^{T+T_{max}} \text{Excess}(t)/(T+T_{max})\right)$$

Such a percentage can be viewed as a measure of the capacity infeasibility degree of a solution.

FEA: Average percentage of feasible solutions that the heuristic was able to find with respect to the total number of problems.

GAP: Average relative deviation between a heuristic solution and the lower bound provided by the Lagrangean relaxation.

We first compare the effectiveness of HP2 in finding feasible solutions relative to the heuristic of Clark and Armentano (1995a) which only uses a single backward step. Table 4 shows the average performance attained by the heuristics with respect to the indexes EXC and FEA when applied to all problems of G1 and it shows the cost effectiveness of the solutions considering the two setup cost patterns. In Table 4, the heuristic HCA found feasible solutions for problems which are a subset of the problems for which the heuristic HP2 found feasible solutions. The average gap is computed only for the problems for which the heuristic HCA found a feasible solution.

Table 4. Group G1: HCA and HP2 capacity feasiblility and solution cost comparisons

Product structure	EXC			FEA		GAP			
						low setup cost		high setup cost	
	HP1	HCA	HP2	HCA	HP2	HCA	HP2	HCA	HP2
Flat	5.03	0.88	0.01	72.5	98.3	1.43	2.16	2.88	2.66
Serial	4.20	2.55	0.86	7.5	68.3	9.09	10.53	0.50	0.04
General	4.41	1.35	0.50	38.3	75.8	4.41	6.03	6.45	6.88

While HP2 improves substantially over HCA relative to the feasibility indexes EXC and FEA, the same is not true with respect to cost effectiveness. HCA employs a smoothing procedure quite similar to our procedure P2. The difference is that HCA conducts the search for feasible solutions exclusively guided by cost improvements while HP2 is oriented by the ratio (10) which combines cost and feasibility aspects. So it expected that HCA outperforms HP2 in solution cost. However, the cost degradation observed in HP2 is more than compensated by its gain in terms of capacity feasibility, which proves the superiority of combined ratio (10) over the cost-oriented ratio.

The next comparison evaluates the cost performance of HP2 against HP4 taking into account different setup costs. The results shown in Table 5 clearly demonstrate the effectiveness of procedures P3 and P4 in finding lower cost solutions. Note that embedded into GAP we may have a duality gap. We will turn back to this later when another experiment will compare the solutions found by metaheuristics to optimal solutions given by CPLEX, which allows us to evaluate the extent of the duality gaps.

Table 5. Group G1: HP2 and HP4 solution cost comparisons

Product structure	low setup cost		high setup cost	
	HP2	HP4	HP2	HP4
Flat	2.19	1.23	7.12	4.69
Serial	17.55	13.02	27.98	15.04
General	6.3	4.42	15.29	11.38

5.3. Computational Results with Addition of Metaheuristics

The evaluation of how the inclusion of tabu search and simulated annealing affect the heuristic's performance is exclusively done with respect to solution cost since they have not increased the number of feasible solutions found (FEA) and the degree of infeasibility (EXC). The computational experiments used group G1 and was first conducted in such way that the effect of tabu search could be evaluated separately and then in conjunction with simulated annealing. Table 6 shows the values of GAP as a function of product structure, problem size and setup cost pattern. Consider the additional notation:

HT: Heuristic incorporating of tabu search.

HTS: Heuristic with the inclusion of tabu search and simulated annealing.

The analysis of the results shown in Table 6 indicates that the addition of tabu search improved the average cost performance for almost all product structures and problem sizes. The combination with simulated annealing caused a positive effect as well, specially in problems with flat and general structures. For serial problems, the average gap reduction was larger and for problems with 40 items, it is significant that the heuristic was able to find a feasible solution only in 5% of them. In an overall evaluation, it is possible to say that HT and HTS obtained the smallest gaps for problems with flat structure and with low setup cost. However, we emphasize that this result depends on duality gaps. This point is clarified in the next section where the heuristic HTS is compared to optimal solutions. Moreover, the behavior of the heuristics are affected by the problem dimension.

Table 6. Group G1: HP4, HT and HTS cost comparisons

Structure	N	Low setup cost			High setup cost		
		HP4	HT	HTS	HP4	HT	HTS
	10	0.96	0.82	0.78	3.88	3.53	3.38
Flat	20	0.62	0.62	0.50	2.39	2.05	1.97
	40	2.17	2.15	1.93	7.98	7.95	7.08
	Mean	**1.23**	**1.18**	**1.05**	**4.69**	**4.45**	**4.09**
	10	7.16	5.79	5.84	10.19	6.93	7.21
Serial	20	18.26	15.91	15.70	18.02	13.66	13.25
	40	25.43	23.78	25.75	52.73	54.33	51.96
	Mean	**13.02**	**11.16**	**11.13**	**15.04**	**11.37**	**11.25**
	10	3.63	3.61	3.18	10.07	9.57	9.14
General	17	1.69	1.54	1.44	5.29	4.54	4.46
	40	9.85	9.63	9.59	24.96	25.20	24.75
	Mean	**4.42**	**4.34**	**4.11**	**11.38**	**10.94**	**10.63**

The graphs showing the dynamic behavior of the incumbent solution when a heuristic is executed are also helpful in the comparative analysis. We selected 10 replications of a problem of size 10x12 to perform this test. Figure 8 shows the evolution of the mean gap attained by heuristics HP4, HT and HTS for a flat structure. Figures 9 and 10 show the same results for serial and general structures, respectively.

Two points deserve comments. As expected, the metaheuristic methods yield better solutions during all execution time. Moreover, HTS is able to attain good solutions much faster than HP4, particularly for flat and serial structures.

5.4. Comparisons with Optimal Solutions

In this section the performance of HTS is compared to optimal solutions. The commercial package CPLEX was applied to optimally solve the set of small-sized problems of group G2. The number of binary variables present in model ME ranges from 18 to 60. Additional experiments have shown that larger problems cannot be solved in a reasonable time by the CPLEX/MIP-solver. In fact we succeeded in finding optimal solution only for 95 of the 120 problems. Table 7 shows the average deviation from optimum achieved by heuristic HTS separated by problem sizes, product structures and setup cost patterns. The number in parentheses represents the number of problems that CPLEX was able to solve to optimality using default parameters. We chose to report two indexes, in order to emphasize the role played by the duality gap in the previous results using group G1. **GAP*opt*** stands for the average cost deviation of HTS from the optimum while **GAP*lb*** stands for the average cost deviation from the Lagrangean lower bound. The averages were only calculated over the problems actually solved by CPLEX. The last column shows the averages attained by each product structure and setup cost profile.

Table 7. Group G2: HTS comparisons to the optimum and the Lagrangean lower bound

Structure	Setup cost	3x6		6x6		3x12		10x6		Mean	
		GAP*lb*	GAP*opt*	GAP*lb*	GAP*opt*	GAP*lb*	GAP*opt*	GAP*lb*	GAP*opt*	GAP*lb*	GAP*opt*
Flat	Low	1.3	0.5 (5)	0.8	0.2 (5)	3.8	1.0 (1)	-	-	1.3	0.4
	High	6.4	0.2 (5)	2.5	0.1 (5)	6.4	1.3 (4)	3.3	1.0 (1)	4.9	0.5
Serial	Low	2.6	1.2 (5)	3.0	1.4 (5)	2.0	0.8 (4)	10.0	8.5 (5)	4.5	3.1
	High	11.8	0.7 (5)	4.4	0.8 (5)	2.7	0.4 (4)	10.6	5.9 (5)	7.6	2.0
General	Low	6.1	3.4 (5)	5.1	1.5 (5)	4.2	1.6 (5)	7.9	4.8 (1)	5.3	2.3
	High	10.1	0.9 (5)	8.0	1.4 (5)	9.1	1.2 (5)	-	-	9.1	1.2
Mean		**6.4**	**1.2**	**4.0**	**0.9**	**5.0**	**0.9**	**9.0**	**6.5**	**5.7**	**1.7**

With regard to the product structure and dimension, the results in Table 7 confirm the analysis made when comparisons were performed using group G1, namely, HTS performs better for flat structures.

Relative to the setup cost profiles, Table 7 shows that if we use GAP*opt* as the comparative index, HTS attains better results in problems with high setup costs. On the other hand, if we use GAP*lb* the conclusion is the opposite. A possible explanation for that resides in the Lagrangean/subgradient method, which generates greater duality gaps when solving problems with high setup costs. The most important observation, however, is concerned with the magnitude of the duality gaps embedded in the lower bounds. Note that the average deviation from the Lagrangean bound is 5.7% while the real average deviation of HTS is 1.7%. In short, this limited computational test shows that about 75% of the deviation from the lower bound is due to the duality gap.

5.5. LP-based Refinements

Another experiment takes the binary solutions yielded by the application of the HTS heuristic, fixes them in model ME and solves it using the CPLEX LP-solver. This procedure can be viewed as a final refinement of heuristic HTS. As stated before, this test also aims to estimate the suboptimality degree attained by the heuristic solutions with respect to the binary variables and continuous variables taken separately. We first compare the solutions obtained by CPLEX to the optimal solutions using group G2. Moreover, the same test is applied to problems from group G1. In Table 8, the values in column **GAP*lb*-LP** represent the deviations of CPLEX solutions relative to the Lagrangean bound and column **GAP*opt*-LP** shows the CPLEX values with respect to the optimum.

Table 8. Group G2: Average gaps using LP solver on HTS binary solutions

Structure	Setup cost	GAP*lb*	GAP*lb*-LP	GAP*opt*	GAP*opt*-LP
Flat	Low	1.3	1.0	0.4	0.1
	High	4.9	4.7	0.5	0.3
Serial	Low	4.5	1.4	3.1	0.1
	High	7.6	5.5	2.0	0.0
General	Low	5.3	3.4	2.3	0.5
	High	9.1	8.6	1.2	0.8
Mean		**5.7**	**4.2**	**1.7**	**0.3**

For problems with flat structure, the results show that the gap reduction with the application of the LP solver is small. For serial and general problems the refinement proved to be of great importance. For these classes, it can be stated that the heuristic HTS is able to find binary solutions very near to the optimal ones. Restricting the analysis to the 8 replications of the 10x6 problem which were solved to optimality, we observed that the deviations from the optimal solutions are always below 2%.

Taking the problems from group G1, we also applied the CPLEX LP-solver to the binary solutions obtained by HP4 and HTS. The aim is to check, in terms of quality of the solutions, if it is worth including metaheuristics instead of just using the simpler heuristic

HP4, since both run in N*T seconds. In this test only part of the problems from group G1, namely the problems with 10 and 20 items, and 12 periods were solved, totalling 120 problems. The results are shown in Table 9.

The overall means shows that the use of metaheuristics is important to finding good binary solutions. On the average, it can be observed that the final gap of the LP solutions provided by HTS is reduced by about 40% when compared to the gap found using HP4. The greatest improvement occurs in problems with serial structure.

Table 9. Group G1: Average gaps using LP-solver on HP4 and HTS binary solutions

Structure	Setup cost	HP4		HTS	
		GAP*lb*	GAP*lb*-LP	GAP*lb*	GAP*lb*-LP
Flat	Low	0.9	0.9	0.7	0.6
	High	3.2	3.1	2.5	2.5
Serial	Low	12.0	4.6	10.1	2.8
	High	17.2	12.2	9.6	4.9
General	Low	3.6	2.8	2.7	1.9
	High	11.6	11.0	9.5	9.0
Mean		**8.1**	**5.8**	**5.9**	**3.6**

6. CONCLUSIONS

In this paper several heuristics were proposed for solving the multilevel constrained lot-sizing problem with setup and lead times. Initially, a basic heuristic method HP4 built upon a mathematical formulation of the problem in terms of echelon stock was developed. This method is composed of 4 procedures and the computational results on a series of randomly generated problems have proved that it is able to find a greater number of capacity feasible solutions when compared to a previous heuristic proposed by Clark and Armentano (1995a). The inclusion of metaheuristics such as tabu search and simulated annealing was tested on a set of medium to large-sized problems. Comparisons to a Lagrangean lower bound showed that the metaheuristic HTS reduced the gap with respect to heuristic HP4. For small-sized problems, the metaheuristic produced solutions very close to the optimal ones. We have also shown that the application of the CPLEX LP-solver to the metaheuristic binary solution reduced substantially the gap in comparison with the application of the LP-solver to the binary solution yielded by the basic heuristic HP4.

We believe that the heuristic HTS is a practical and efficient way to improve the performance of lot-sizing tasks embedded in MRP systems as it takes into consideration some crucial practical issues such as finite capacity, non-zero lead times and cost optimization.

ACKNOWLEDGEMENTS

This research was partially supported by Fundação de Amparo à Pesquisa do Estado de S. Paulo (FAPESP) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERENCES

Afentakis P., B. Gavish and U. Karmarkar, "Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multistage Assembly Systems", *Management Science*, 30, 2 (1984), 222-239.

Armentano, V. A., R. E. Berretta, and P. M. França, "Lot-Sizing in Capacitated Multi-Stage Serial Systems", submitted to *Production and Operations Management.*

Bahl H. C., L. P. Ritzman and J. N. D. Gupta, "Determining Lot-Sizes and Resource Requirements: A Review", *Operations Research*, 35, 3 (1987), 237-249.

Billington P. J., J. O. McClain and L. J. Thomas, "Heuristics for Multilevel Lot-Sizing with a Bottleneck", *Management Science*, 32, 8 (1986), 989-1006.

Camerini, L. Fratta and F. Maffioli, "On Improving Relaxation Methods by Modified Gradient Techniques", *Mathematical Programming Study*, 3, (1975), 26-54.

Clark A. J. and H. Scarf, "Optimal Policies for a Multi-Echelon Inventory Problem", *Management Science*, 6, (1960), 475-490.

Clark, A. R. and V. A. Armentano, "Echelon Stock Formulation for Multi-Stage Lot-Sizing with Component Lead Times", *International Journal of Systems Sciences*, 24, 9 (1993), 1759-1775.

— and —, "A Heuristic for a Resource-Capacitated Multi-Stage Lot-Sizing Problem with Lead Times", *Journal of the Operational Research Society*, 46, (1995a), 1208-1222.

— and —, "The Application of Valid Inequalities to the Multi-Stage Lot-Sizing Problem", *Computers and Operations Research*, 22, 7 (1995b), 669-680.

Florian M., J. K. Lenstra and A. H. G. Rinnoy Kan, "Deterministic Production Planning Algorithms and Complexity", *Management Science*, 26, 7 (1980), 669-679.

França, P. M., V. A. Armentano, R. E. Berretta e A. R. Clark, "A Heuristic for Lot-Sizing in Multi-Stage Systems", *Computers and Operations Research*, 24, 9 (1997), 861-874.

Glover, F. and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1998.

Harrison, T. P. and H. S. Lewis, "Lot Sizing in Serial Assembly Systems with Multiple Constrained Resources", *Management Science*, 42, 1 (1996), 19-36.

Katok, E., T. P. Harrison and H. S. Lewis, "Lot sizing in General Assembly Systems with Setup Costs, Setup Times and Multiple Constrained Resources", *Management Science*, 44, 6 (1998), 859-877.

Kirkpatrick S., C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, 220, 4598 (1983), 671-680.

Kuik, R., M. Salomon, L. N. Van Wassenhove, "Batching Decisions: Structure and Models", *European Journal of Operation Research*, 75, (1994), 243-263.

—, —, — and J. Maes, "Linear Programming, Simulated Annealing and Tabu Search Heuristics for Lotsizing in Bottleneck Assembly Systems", *IIE Transactions*, 25, 1 (1993), 62-72.

Maes J. and L. N. Van Wassenhove, "Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A Computational Comparison (Part I: Static Case)", *IIE Transactions*, 18, 2 (1986), 114-123.

—, J. O. McClain and L. N. Van Wassenhove, "Multilevel Capacitated LotSizing Complexity and LP based Heuristics", *European Journal of Operational Research*, 53, (1991), 131-148.

Martin O. C. and S. W. Otto, "Combining Simulated Annealing with Local Search Heuristics", http://www.cse.ogi.edu/~otto/co.html, to appear in Annals of Operations Research.

Moscato P. and J. F. Fontanari, "Stochastic vx. Deterministic Update in Simulated Annealing", *Physics Letters*, 146, 4 (1990), 204-208.

Pochet, Y. and L. A. Wolsey, 1991, " Solving Multi-Item Lot-sizing Problems Using Strong Cutting Planes", *Management Science*, 37, 1 (1991), 53-67.

Roll, Y. and R. Karni, "Multi-item, Multi-level Lot Sizing with an Aggregate Capacity Constraint", *European Journal of Operation Research*, 51, (1991), 73-87.

Tempelmeier H. and M. Derstroff, "A Lagrangean-based Heuristic for Dynamic Multilevel Multiitem Constrained Lotsizing with Setup Times", *Management Science*, 42, 5 (1996), 738-757.

Wagner, H. M. and T. M. Whitin, "Dynamic Version of the Economic Lot Size Model", *Management Science*, 5, 1 (1958), 89-96.