CAPACITATED LOT-SIZING PROBLEM WITH SETUP TIMES, STOCK AND DEMAND SHORTAGES

Nabil Absi [*,1] **Safia Kedad-Sidhoum** [**]

[*] *Laboratoire d'Informatique d'Avignon, 339 chemin des Meinajariès, 84911 Avignon Cedex 09, France*
[**] *Laboratoire d'Informatique de Paris 6, 4 place Jussieu, 75 252 Paris Cedex 05, France*

Abstract: We address a multi-item capacitated lot-sizing problem with setup times, stock and demand shortages. Demand cannot be backlogged, but can be totally or partially lost. Safety stock is an objective to reach rather than an industrial constraint to respect. In this paper we consider an approach based on a lagrangean relaxation of the capacity constraints to generate a lower bound of the optimal value. This relaxation induces N single-item uncapacitated lot-sizing problems, where N denotes the number of items. A polynomial algorithm is developed to solve the sub-problems optimally. Some experimental results showing the effectiveness of the approach are reported.

Keywords: Production, Planning, Manufacturing, Optimization, Dynamic programming, Relaxation, Capacity, Constraints, Integer Programming.

1. INTRODUCTION

The Multi-item Capacitated Lot-sizing problem with Setup times, Shortage costs and Safety Stock deficit costs (MCLS4) is a production planning problem in which there is a time-varying demand for a set of N items over T periods. The production should satisfy a restricted capacity and must take into account a set of additional constraints. Indeed, launching the production of an item i at a given period t for a demand requirement d_{it} involves a variable resource consumption v_{it} and a setup time f_{it}. The total available capacity at period t is c_t. For each period t, an inventory cost γ_{it}^{+} is attached to each item i as well as a variable unit production cost α_{it} and a setup cost β_{it}.

The problem has the distinctive feature of allowing requirement shortages due to tight resource capacities. Indeed, when we are in lack of capacity to produce the total demand, we try to spread the available capacity among the items by minimizing the total amount of shortages. Thus, we introduce in the model a unit cost parameter φ_{it} for item i at period t for the requirement not met regarding the demand. These costs should be viewed as penalty costs and their values are very high in comparison with other cost components.

The use of safety stock is widely prevalent in industry to counter variability that may be present in a supply chain environment. In our study, a safety stock is an objective to be reached rather than an industrial constraint to respect. When we cannot reach this target, we talk about safety stock deficits for which we introduce a unitary cost parameter γ_{it}^{-} for each item i at each period t. These costs should also be viewed as penalty

[1] This work has been partially financed by DYNASYS S.A., under research contract no. 588/2002.

costs and their values are lower than shortage costs and very high in comparison with other cost components.

The MCLS4 problem consists in finding a production planning that minimizes the shortages, the safety stock deficits as well as the setup, the inventory and the production costs.

The new characteristics introduced in this paper are frequently encountered in process industries. They model real-world problems where the capacities are tight and the most important objectives are to try to meet the maximum amount of client's needs and to reach the safety stock levels.

(Chen and Thizy, 1990) have proved that the multi-item capacitated lot-sizing problem with setup times is strongly NP-hard.

There are few references dealing with lot-sizing problems with shortage costs or safety stocks. Recently, (Aksen *et al.*, 2003) addressed the uncapacitated single-item lot-sizing problem with shortage costs. The authors proposed an $O(T^2)$ forward dynamic programming algorithm to solve the problem. (Loparic *et al.*, 2001) proposed valid inequalities for the single-item uncapacitated lot-sizing problem with lower bounds on the stock variables and the objective of maximizing sales.

(Diaby *et al.*, 1992) deal with lack of capacity by considering overtimes. (Zangwill, 1966; Zangwill, 1969) proposed a model with backlogs where the demands must be satisfied with an extra cost if there is a delay.

The main contributions of this paper are twofold. First, we develop a polynomial algorithm to solve the uncapacitated single-item lot-sizing problem with shortage on demand and safety stock deficit costs (ULS4) optimally. Second, we use an approach based on a lagrangean relaxation of the capacity constraints to solve the MCLS4 problem.

An outline of the remainder of the paper follows. Sections 2 and 3 describe MIP formulations of the MCLS4 problem and its capacity constraints relaxation. In section 4 we state some properties of the ULS4 optimal solutions and we describe a polynomial algorithm to solve it. Finally, computational results are given in section 5 to show the effectiveness of the approach based on a lagrangean relaxation of capacity constraints.

2. FORMULATION OF THE MCLS4 PROBLEM

In this section we present a MIP formulation of the MCLS4 problem. In the remainder of the section, we consider that $i = 1, \ldots, N$ and $t = 1, \ldots, T$. We set x_{it} as the quantity of item i produced at period t. We need also to define y_{it} as a binary variable equal to 1 if item i is produced at period t (i.e. if $x_{it} > 0$) to deal with the fixed setup times and costs. The variable s_{it} for item i is the inventory value at the end of period t. The shortage for item i at period t is modelled by a non negative variable r_{it} with a high unit penalty cost in the objective function, because the main goal is to satisfy the customer and thus to have the minimum amount of the requirements not met.

Let I_{it}^+ and I_{it}^- represent respectively overstock and safety stock deficit variables of item i at period t. I_{it}^- has a high unit penalty cost in the objective function. However, this penalty is lower than shortage cost. We set l_{it} as a parameter which represents safety stock value of item i at period t. Thus, the stock of item i at period t is given by the expression $I_{it}^+ + l_{it} - I_{it}^-$. We set $\delta_{it} = l_{it} - l_{i(t-1)}$ which is the safety stock variation between two consecutive periods, it can be positive or negative. The MCLS4 problem is described by the following MIP:

$$\min \sum_{i,t} (\alpha_{it} x_{it} + \beta_{it} y_{it} + \varphi_{it} r_{it} + \gamma_{it}^+ I_{it}^+ + \gamma_{it}^- I_{it}^-) \quad (1)$$

$$I_{i(t-1)}^+ - I_{i(t-1)}^- + r_{it} + x_{it} = d_{it} + \delta_{it} + I_{it}^+ - I_{it}^-, \quad (2)$$

$$\sum_i v_{it} x_{it} + f_{it} y_{it} \leq c_t, \quad \forall t \quad (3)$$

$$x_{it} \leq M y_{it}, \quad \forall i, t \quad (4)$$

$$r_{it} \leq d_{it}, \quad \forall i, t \quad (5)$$

$$I_{it}^- \leq l_{it}, \quad \forall i, t \quad (6)$$

$$x_{it}, r_{it}, I_{it}^+, I_{it}^- \geq 0, \quad \forall i, t \quad (7)$$

$$y_{it} \in \{0, 1\}, \quad \forall i, t \quad (8)$$

The objective function (1) minimizes the total cost induced by the production plan (unit production costs, inventory costs, shortage costs, safety stock deficit costs and setup costs). Constraints (2) are the inventory flow conservation equations through the planning horizon. Constraints (3) are the capacity constraints; the overall consumption must remain lower than the available capacity. If we produce an item, then the production must not exceed a maximum production level M (constraints (4)). M could be set to the minimum between the total requirement on section $[t, T]$ of the horizon and the highest quantity of item i that can be produced, $M = \min\left(\sum_{t'=t}^{T} d_{it'}; (c_t - f_{it})/v_{it}\right)$. Constraints (5) and (6) define upper bounds on respectively the shortage and the safety stock deficit for item i on period t. Constraints (7) and (8) characterize the variable's domains: x_{it}, r_{it}, I_{it}^+ and I_{it}^- are non-negative and y_{it} is a binary variable for $i = 1, \ldots, N$ and $t = 1, \ldots, T$.

3. RELAXING THE RESOURCE CAPACITY CONSTAINTS

The general idea behind the lagrangean relaxation approach is to decompose the large multi-item capacitated lot-sizing problem into smaller single-item uncapacitated lot-sizing sub-problems. This is possible by relaxing the resource capacity constraints while using a set of lagrange multipliers π_t in the objective function of the MCLS4 model. The capacity constraints (3) are considered as coupling constraints in the sense that they prevent the separation into single-item sub-problems.

The lagrangean relaxation of the MCLS4 capacity constraints decomposes the problem into N single-item uncapacitated lot-sizing problems with shortage and safety stock deficit costs denoted ULS4. Since we consider each item separately, it may be more convenient to remove the item index to facilitate the reading of the remaining mathematical formulations. The ULS4 problem is described in what follows:

$$\min \sum_t \alpha_t x_t + \beta_t y_t + \gamma_t^+ I_t^+ + \gamma_t^- I_t^- + \varphi_t r_t + \sum_t \pi_t (v_t x_t + f_t y_t) \quad (9)$$

$$I_{t-1}^+ - I_{t-1}^- + x_t + r_t = d_t + \delta_t + I_t^+ - I_t^-, \quad \forall t \quad (10)$$

$$x_t \leq M y_t, \quad \forall t \quad (11)$$

$$r_t \leq d_t, \quad \forall t \quad (12)$$

$$I_t^- \leq l_t, \quad \forall t \quad (13)$$

$$x_t, r_t, I_t^+, I_t^- \geq 0, \quad \forall t \quad (14)$$

$$y_t \in \{0,1\}, \quad \forall t \quad (15)$$

To the best of our knowledge, the ULS4 problem was never addressed in the literature. It is an extension of the classical ULS problem (Wagner and Whitin, 1958). If we remove r_t and I_t^- from the model we obtain the ULS model which is solvable in $O(T \log T)$ (Wagelmans *et al.*, 1992). If we remove only I_t^- variables, we obtain the ULS model with shortage costs. This problem was solved in $O(T^2)$ by (Aksen *et al.*, 2003). If we remove r_t, δ_t and the constraint (13), we obtain the ULS model with backlogs. This problem was solved in $O(T^2)$ by (Zangwill, 1969).

4. DYNAMIC PROGRAMMING ALGORITHM FOR SOLVING ULS4

In this section, we propose a fixed charge network model for the ULS4 problem. We present also some properties of the ULS4 optimal solutions. These properties enable us to propose a recursive dynamic programming algorithm for solving the problem in polynomial time in the worst case.

4.1 Fixed charge network formulation

Figure 1 corresponds to a fixed charge network presentation of ULS4 problem. Vertices 1 to T represent the periods of the problem. Vertex 0 represents the principal source, its availability is equal to $D = \sum_{t=1}^{T} (d_t + \delta_t) = \sum_{t=1}^{T} d_t + l_T - l_0$, with $l_0 = 0$. X and R are two transhipment nodes which transfer respectively the total demand and the sum of the shortages. The remainder of the transhipment transits from node T, that is the maximum deficit on safety stock at the last period. It's obvious that the sum of inputs at nodes X, R and T is equal to quantity which transit from node 0. Each node $t \in \{1, 2, \ldots, T\}$ has a need equal to δ_t, $\delta_t = l_t - l_{t-1}$. δ_t can be positive or negative. Nodes $\{1', 2', \ldots, T'\}$ represent the demands d_t at periods $t \in \{1, \ldots, T\}$. At each node $t \in \{1', 2', \ldots, T'\}$, we have a demand d_t. The arcs of this graph are defined as follows:

- (X, t) are the production arcs, $t \in \{1, 2, \ldots, T\}$. They represent the quantity produced at period t,
- $(t, t+1)$ are the inventory arcs, $t \in \{1, 2, \ldots, T-1\}$. Each arc represents the ending stock of period t and the entering stock at period $t+1$,
- $(t, t-1)$ are the safety stock deficit arcs, $t \in \{2, \ldots, T\}$. They represent the safety stock deficit of period $t-1$. The capacity of this arc is limited to l_t,
- (t, t') arcs represent the satisfied demand at period t, $t \in \{1, 2, \ldots, T\}$ and $t' \in \{1', 2', \ldots, T'\}$,
- (R, t') arcs represent the shortage at period t, $t' \in \{1', 2', \ldots, T'\}$.

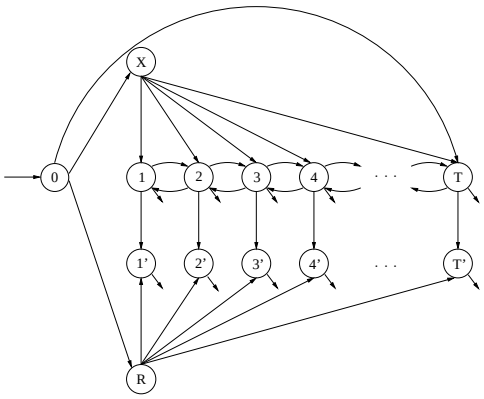

Fig. 1. Fixed charge network model for ULS4

All the cost functions on this network are concave. In fact, the costs on arcs (t, t'), $(0, X)$ and $(0, R)$ are equal to zero. The arcs (R, t') have a unitary cost equal to φ_t. The arcs $(t, t+1)$ and $(t, t-1)$ have unitary costs equal to γ_t^+ and γ_t^- respectively. The cost on the production arcs (X, t) is equal to zero if the flow x_t is null and $(\beta_t + \pi_t f_t + (\alpha_t + \pi_t v_t) x_t)$ if the flow x_t is positive.

Finding an optimal solution to the ULS4 problem is equivalent to finding an extreme flow with minimum cost on the network of figure 1.

4.2 Characterization of the ULS4 optimal solution

In order to propose a dynamic programming algorithm to solve the ULS4 problem, we present some properties of the optimal solutions. We first begin with some definitions.

Definition 1. A period t is a **production point** if $x_t > 0$.

Definition 2. A period t is an **inventory point** if ($I_t^+ = 0$ and $I_t^- = 0$) or ($I_t^+ = 0$ and $I_t^- = l_t$). More precisely, a period t is an:

- **Inventory point of *type 1*** if $I_t^+ = 0$ and $I_t^- = 0$;
- **Inventory point of *type 2*** if $I_t^+ = 0$ and $I_t^- = l_t$.

Definition 3. A feasible flow $(\widehat{x}, \widehat{r}, \widehat{I^+}, \widehat{I^-})$ is an extreme flow if there does not exist feasible flows $(\widehat{x}_1, \widehat{r}_1, \widehat{I_1^+}, \widehat{I_1^-})$ and $(\widehat{x}_2, \widehat{r}_2, \widehat{I_2^+}, \widehat{I_2^-})$ such that: $(\widehat{x}, \widehat{r}, \widehat{I^+}, \widehat{I^-}) = (\widehat{x}_1, \widehat{r}_1, \widehat{I_1^+}, \widehat{I_1^-}) + (\widehat{x}_2, \widehat{r}_2, \widehat{I_2^+}, \widehat{I_2^-})$

It is well known (Zangwill, 1968) that if the objective function has a finite global minimum on the feasible region, then there must be an extreme flow which is an optimal flow. Since the feasible region has a finite number of extreme flows, a search of all extreme flows will give the optimal flow. However, such a procedure might be extremely hard. In what follows, we present some characteristics of the extreme flows of ULS4 which will allow us to facilitate this search.

Lemma 1.
$I_t^+ I_t^- = 0$ for all $t \in 1, \ldots, T$.

Proof. This result derives from the fact that an extreme flow uses a set of arcs forming a tree. Hence, we can not have $I_t^+ > 0$ and $I_t^- > 0$. □

Proposition 1.
The following propositions are equivalent:

- $(\widehat{x}, \widehat{r}, \widehat{I^+}, \widehat{I^-})$ is an extreme flow of ULS4;
- $(\widehat{x}, \widehat{r}, \widehat{I^+}, \widehat{I^-})$ is feasible and between every pair of production points (i, k) / $i < k$, there exists an inventory point j / $(i \leq j < k)$.

Proof. Suppose that we have two production points i and k with $i < k$ and there is no inventory point between these two points in an extreme flow. This implies that for all the vertices t between i and k we have either ($I_t^+ > 0$ and $I_t^- = 0$) or ($I_t^+ = 0$ and $0 < I_t^- < l_t$) (Lemma 1, Definition 2). This extreme flow contain a cycle where there is no edge fixed to its bound. If such a cycle exists in an extreme flow, then this flow can easily be expressed according to two other feasible flows. This proves that such a flow is not extreme and contradicts the assumption. □

The following proposition is equivalent to those of proposition 1:

- $(\widehat{x}, \widehat{r}, \widehat{I^+}, \widehat{I^-})$ is feasible and between every pair of adjacent inventory points there exists at most one production point.

4.3 Dynamic programming algorithm

Based on the properties of the optimal solutions, we develop a dynamic programming algorithm to solve the ULS4 problem. We have proved in the last section that a feasible solution is an extreme flow if and only if between every couple of adjacent inventory points i, k with $i < k$, there exists at most a production point j $(i < j \leq k)$. To find the optimal solution, it suffices to search all the extreme flows that satisfy these properties and identify the minimum cost extreme flow. The general idea of the algorithm is to consider all the triplets (i, j, k) (with $i < j \leq k$, i and k are inventory points, and j is a production point) and to find the sequence of triplets (i, j, k) with the minimum cost value. For each triplet (i, j, k) we define a sub-problem which we solve optimally.

We define a restricted problem ULS4 denoted ULS4_{ijk}^{uv}. The indices u and v characterize the inventory points (Type 1 or 2). Indice u (resp. v) is fixed to 0 if the period $i-1$ (resp. k) is an inventory point of type 1. Indice u (resp. v) is fixed to 1 if the period $i-1$ (resp. k) is an inventory point of type 2. If we assume that $\{i, \ldots, k\}$ represent the set of integers from i to k, ULS4_{ijk}^{uv} is defined on the interval of periods $\{i, \ldots, k\}$. There is a production only at period j, the production at periods $t \in \{i, \ldots, k\} \setminus j$ is null ($x_t = 0$). We denote Z_{ijk}^{uv} the optimal value of the objective function of the problem ULS4_{ijk}^{uv}. ULS4_{ijk}^{uv} can be solved using a minimum cost flow algorithm.

To find the value of Z_{ik}^{uv} on the interval $i, \ldots, k$, we search the minimum between all the optimal policies on the interval $\{i, \ldots, k\}$ with j the unique production period $(i < j \leq k)$ and the optimal policy on the interval $\{i, \ldots, k\}$ with no production period (denoted Z_{i0k}^{uv}).

$$Z_{ik}^{uv} = \min \left[Z_{i0k}^{uv}, \min_{i<j\leq k} Z_{ijk}^{uv} \right] \qquad (16)$$

An optimal basis for the ULS4 solution corresponds to the minimum value of a sequence of adjacent inventory points intervals. The function F_k^u defines the minimum value of the adjacent inventory point sequence from period 1 to period k, this function can be calculated using the following recursive formula:

$$F_k^v = \min_{i \leq k; u=0,1} \left\{ F_{i-1}^u + Z_{ik}^{uv} \right\} \tag{17}$$

The optimal value of ULS4 problem is given by $F_T = \min_{v=0,1} F_T^v$.

The algorithm is described as follows:

Algorithm 1 Algorithm to solve ULS4

$u \leftarrow 0$, $v \leftarrow 0$
for all $t \in \{1, \ldots, T\}$ and $u \in \{0, 1\}$ **do**
  $F_t^u \leftarrow +\infty$, $F_0^u \leftarrow 0$
end for
for $i = 1 \;\; T-1$ **do**
  for $k = i+1 \;\; T$ **do**
    repeat
      Calculate Z_{i0k}^{uv}, $Z_{ik}^{uv} \leftarrow Z_{i0k}^{uv}$
      for $j = i+1 \;\; k$ **do**
        Calculate Z_{ijk}^{uv}, $Z_{ik}^{uv} \leftarrow \min \left[Z_{ik}^{uv}, Z_{ijk}^{uv} \right]$
      end for
      $F_k^v \leftarrow \min \left\{ F_k^v, F_{i-1}^u + Z_{ik}^{uv} \right\}$
      $v \leftarrow 1 - v$
      if $v = 0$ **then**
        $u \leftarrow 1 - u$
      end if
    until $u = 0$ and $v = 0$
  end for
end for
$F_T = \min_{u=0,1} F_T^u$

4.4 Algorithm complexity

The ULS4_{ijk}^{uv} problems can be solved using a minimum cost flow algorithm. The best strongly polynomial minimum cost flow algorithm is proposed by (Orlin, 1993). It solves the problem in $O(m \log n(m + n \log n))$, where m and n represent respectively the number of arcs and the number of nodes of the network.

In the worst case, the number of vertices of the network which represents the problem Z_{ijk}^{uv} is $2T+3$ and the number of arcs is $5T+1$. Then, $n = 2T+3$ and $m = 5T+1$. The calculation of Z_{ijk}^{uv} is done in $O(T^2 \log^2 T)$. The global complexity of the algorithm that solves ULS4 is $O(T^5 \log^2 T)$.

5. COMPUTATIONAL EXPERIMENTS

In this section, we present experimental results resulting from the application of the lagrangean relaxation of the capacity constraints of the problem MCLS4. Our algorithms are implemented in the C++ programming language and are integrated in an APS software. We use the callable CPLEX 9.0 library to solve the MIP problems.

We have performed computational tests on a series of extended instances from the lot-sizing library LOTSIZELIB, initially described in (Trigeiro *et al.*, 1989). These instances are denoted tr_{N-T}, where N is the number of items and T is the number of periods. These are characterized by a variable resource consumption equal to one, and enough capacity to satisfy all the requirement over the planning horizon. They are also characterized, by an important setup cost, a small setup time and no safety stock.

Since these instances have enough capacity to satisfy all the requirements over the planning horizon, we have performed our experiments on the class B of instances described in (Absi and Kedad-Sidhoum, 2005). These new instances are obtained by increasing the resource requirements and adding safety stocks to (Trigeiro *et al.*, 1989) instances.

The lagrangean sub-problems obtained at each step of the lagrangean relaxation are solved optimally using CPLEX 9.0 solver for the model ULS4 described in section 2. However, it would be interesting to use a polynomial minimum cost flow algorithm. We use a subgradient method to calculate these multipliers. The upper bounds used to calculate the lagrangean multipliers are obtained using a MIP-based heuristic described in (Absi and Kedad-Sidhoum, 2005). The CPU-time needed to obtain these upper bounds varies between 1 and 40 seconds for our test instances. The lagrangean relaxation parameters are fixed according to the schema proposed by (Held *et al.*, 1974).

We have measured the quality of the lower bounds obtained using the lagrangean relaxation (denoted LR) by comparing them with the lower bounds obtained with a monolithic resolution of the initial MCLS4 problem (denoted BC). The MCLS4 problem as well as the generated sub-problems at each step of the lagrangean relaxation are solved using the standard branch-and-cut of CPLEX 9.0 solver.

For the two algorithms, LB and UB represent respectively the lower bound and the upper bound values at the termination of the algorithm. The algorithm comparisons are also based on the following criteria. The first one called GAP is equal to $(UB - LB)/UB$, and the second one is a CPU-time denoted Tm. Md represents the method that is used. The BC method stops if a time-limit of 900 seconds is reached. The LR method stops if a maximum number of 100 iterations is reached. Table 1 summarizes the computational behaviour of the approach.

Table 1. Experimental results

N	T	Md	GAP	UB	LB	Tm
6	15	BC	1,7%	5825939	5728964	900
6	15	LR	3,3%	5858168	5665290	26
6	30	BC	4,0%	7814548	7501442	900
6	30	LR	3,5%	7832065	7555426	89
12	15	BC	3,5%	13308806	12838374	900
12	15	LR	4,1%	13292194	12749866	69
12	30	BC	14,9%	17044633	14510544	900
12	30	LR	9,8%	16988946	15330246	233
24	15	BC	6,4%	28685166	26854630	900
24	15	LR	3,7%	28655048	27587051	134
24	30	BC	12,9%	46005650	40053010	900
24	30	LR	8,9%	45512020	41459351	351

From table 1 we can notice that the LR method gives generally better lower bounds than BC method, CPU times are also clearly lower. We also can notice that the GAPs obtained by the LR method are better than those obtained by BC for the instances of bigger size.

Figure 2 represents the GAP variation of the LR method according to the number of lagrangean relaxation iterations which varies from 0 to 200.

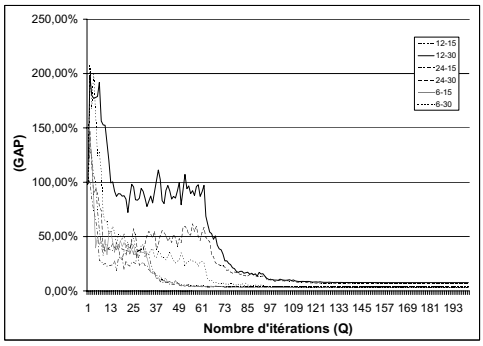

Fig. 2. Lagrangean relaxation

From Figure 2 we can notice that instances with 30 periods need more iterations than instances with 15 periods to converge and the instances with more items need also more iterations. In fact, instances with 15 periods converge to a good solution after 100 iterations, when instances with 30 periods converge after 140 iterations.

6. CONCLUSION

In this paper, we propose a mathematical formulation of a new multi-item capacitated lot-sizing problem with setup times. This formulation takes into account several industrial constraints, shortage costs and safety stock deficit costs. To derive a good lower bound, we have used an approach based on the lagrangean relaxation of the resource capacity constraints. We have also proposed an algorithm that solves the induced single-item sub-problems optimally by solving T^3 minimum cost flow problems where T denotes the number of periods. It would be interesting to implement a specific minimum cost flow algorithm to get a better time complexity in practice. Several issues could be pursued. The complexity can be improved using a geometric approach (Wagelmans *et al.*, 1992). Moreover, the lagrangean solutions could be used to construct feasible solutions (Trigeiro *et al.*, 1989) or in a branch-and-bound framework.

REFERENCES

Absi, N. and S. Kedad-Sidhoum (2005). Mip--based heuristics for multi-item capacitated lot-sizing problem with setup times and shor--tage costs. Available at www.optimization-online.org.

Aksen, D., K. Altinkemer and S. Chand (2003). The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research* **147**, 558–566.

Chen, W.H. and J.M. Thizy (1990). Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research* **26**, 29–72.

Diaby, M., H.C. Bahl, M.H. Karwan and S. Zionts (1992). A lagrangean relaxation approach for very large scale capacitated lot-sizing. *Management Science* **38 (9)**, 1329–1340.

Held, M., P. Wolfe and H.D. Crowder (1974). Validation of subgradient optimization. *Mathematical Programming* **6**, 62–88.

Loparic, M., Y. Pochet and L.A. Wolsey (2001). The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming* **89**, 487–504.

Orlin, J.B. (1993). A faster strongly polynomial minimum cost flow algorithm. *Operations Research* **41(2)**, 338–350.

Trigeiro, W., L.J. Thomas and J.O. McLain (1989). Capacitated lot-sizing with setup times. *Management Science* **35**, 353–366.

Wagelmans, A., C.P.M. Van Hoesel and A. Kolen (1992). Economic lot sizing: an $o(n \log n)$ that runs in linear time in the wagner-whitin case. *Operations Research* **40**, S145–S156.

Wagner, H.M. and T.M. Whitin (1958). A dynamic version of the economic lot size model. *Management Science* **5**, 89–96.

Zangwill, W.I. (1966). A deterministic multi-period production scheduling model with backlogging. *Management Science* **13**, 105–119.

Zangwill, W.I. (1968). Minimum concave cost flows in certain networks. *Management Science* **14**, 429–450.

Zangwill, W.I. (1969). A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science* **15 (9)**, 506–527.