Production Planning Problem with Sequence Dependent Setups as a Bilevel Programming Problem

Zrinka Lukač, Kristina Šorić, Višnja Vojvodić Rosenzweig
Faculty of Economics Zagreb
Kennedyev trg 6
10000 Zagreb, Croatia
e-mail: zlukac@efzg.hr, ksoric@efzg.hr, vvojvodic@efzg.hr

November 30, 2004

Abstract

Each of n products is to be processed on two machines in order to satisfy known demands in each of T periods. Only one product can be processed on each machine at any given time. Each switch from one item to another requires sequence dependent setup time. The objective is to minimize the total setup time and the sum of the costs of production, storage and setup. We consider the problem as a bilevel mixed 0-1 integer programming problem. The objective of the leader is to assign the products to the machines in order to minimize the total sequence dependent setup time, while the objective of the follower is to minimize the production, storage and setup cost of the machine. We develop a heuristics based on tabu search for solving the problem. At the end, some computational results are presented.

Key words: production, two machines, sequence dependent setup times, bilevel mixed 0-1 integer programming problem, heuristics based on tabu search.

1 Introduction

Our work is motivated by the problem confronted by a pharmaceutical company. The considered company has to produce certain products in order to satisfy the known demands for each period of the planning horizon T and with minimum production, setup and storage costs. In the considered company, there are two assembly lines, each of them with its machine producing the final form of the product. These two machines represent the bottleneck of the production and are different in the sense of different capacities, technologies, production costs, setup costs and setup times for the same product, but both of them are appropriate for the production of all the products. Due to the different types of pharmaceutical products, each switch of these machines from one product to another requires the sequence dependent setup time, that is, the need to clean up after each product results in significant setup times. Each machine can process only one product at a time.

We can say that the described problem can be determined as a kind of a classical deterministic capacitated lot sizing problem with sequence independent setup costs and sequence dependent setup times, on two machines (in this case, representing the bottleneck of the production process), without the possibility to produce more than one product in a certain period.

But, due to the market changes and the present competition, a senior manager in the considered firm wants to satisfy the demand for different products and to be ready, if possible, for other orders. Also, he tried to avoid the underutilization of human and capital resources. In this sense, he wants to minimize the total setup time regardless of the costs of production, storage and setup. In this way he wanted to avoid the bottleneck mentioned above. Because of his request, we introduce the hierarchy into the problem and model it as a bilevel mixed 0-1 integer programming problem where the senior manager acts as the leader and the middle manager controlling the machines acts as the follower. The leader of the bilevel problem minimizes the sum of the total setup time assigning the products to the machines. Since the machine capacities are expressed in time units, shorter setup time leaves more time for production. More time for production further implies better performance in terms of satisfying the demand, preparedness for other orders and avoiding the bottleneck. After the leader's decision about how to reduce the setup time, the follower can operate to minimize his production, storage and setup cost of the machines.

A good review of scheduling problems involving setup consideration and its importance could be found in Allahverdi, Gupta, Aldowaisan (1999) which mentions other works dealing with the minimization of the total setup time listed and commented. Sumichrast (1987) considers the problem of minimizing the total setup time and scheduling parallel processors in a make-to-stock environment with sequence dependent setup costs.

Generally, in a basic bilevel programming problem, the first decision-making level (upper-level) is called the leader and the second level (lower-level) the follower. The leader makes a decision by first optimizing his objective function. The follower observes the leader's decision and reacts in a way that is optimal for him. A vast majority of research on bilevel programming has concentrated on the linear version of the problem. Studies of this kind can be found in, for example, Vicente and Calamai (1994) or in Vicente, Savard and Judice (1996).

One of the first papers in the bilevel mixed integer programming dealing with the heuristics were the works of Moore and Bard (1989) and Wen and Huang (1996). Moore and Bard developed a basic implicit enumeration scheme that finds good fesible solutions. A series of heuristics are then proposed in an effort to strike a balance between accurancy and speed. Wen and Huang applied the Simple Tabu Search Algorithm to solve a very simple problem. The most recent work in this area proposing a similar model to the one considered in this paper is the paper of Cao and Chen (2006). They addressed a capacitated plant selection problem in a decentralized manufacturing environment where the principal firm and the auxiliary plants operate independently in an organizational hierarchy. The problem was considered as a bilevel mixed 0-1 integer programming problem with the leader's goal to minimize the opportunity costs of over-setting production capacities in the opened plants and the follower's costs of production and transportation among the opened plants and the principal firm. They transformed the problem and then applied the available software for small dimensions of the problem.

Ben-Ayed and Blair (1989) proved that even bilevel linear programming problems are NP-hard. Therefore, solving a bilevel mixed-integer programming problem needs a heuristics.

Following the structure of the model we are considering here, the follower has to solve a kind of a deterministic capacitated lot sizing problem with sequence independent setup costs and sequence dependent setup times, on two machines. This is one of the most difficult lot-sizing problem and even the capacitated lot sizing problem has been shown to be NP-hard (Bitran and Yanasse, 1982). When setup times are included in the model, finding a feasible solution to the capacitated lot sizing problem also becomes an NP-complete problem (Garey and Johnson, 1979). Based on these results, it is very hard to propose an algorithm giving the optimal solution. Therefore, research on developing effective heuristics is needed. A good review of models and algorithms in this area is given in Karimi, Fatemi Ghomi and Wilson (2003). Also, some previous work on lot sizing and scheduling is presented in Drexl and Kimms (1997) and Wolsey (1997). These papers list a number of authors working on these problems. However, very few of them consider the problems including sequence dependent setup times. Hung, Chen, Shih and Hun (2003)considered the production planning problems with setups, but the setups were not sequence dependent. Gupta and Magnusson (2005) have studied a version of the problem with sequence dependent setup costs and times and proposed a heuristics to solve it. Their goal was to minimize the setup and inventory costs, while for the large scale instances of the problem, the setup times are fixed, positive and the same.

In the next section , we will formulate the model defined as a bilevel mixed 0-1 integer programming problem. The objective of the leader is to assign the products to the machines in order to minimize the total setup time, while the objective of the follower is to minimize the production, storage and setup cost of the machine. The considered problem is called Production Planning Problem with Sequence Dependent Setups. In Section 3 we introduce and explain the proposed heuristics based on tabu search. First we are defining the way of constructing the initial solutions and after that, the way of constructing their neighborhoods. At the end of this section the heuristics computer implementation is presented. The numerical results and the conclusions are reported in Section 4, and finally, in Section 5 we discuss the future work.

2 Formulation of the problem

In order to formulate the above described problem, let us introduce the following notations:

1. Index:

- i - product type, $i = 1, 2, ..., n$
- j - machine type, $j = 1, 2$
- t - planning period, $t = 1, 2, ..., T$

2. Parameters:

- d_{it} - the demand for product i in period t
- p_{itj} - the unit production cost of product i in period t on machine j
- h_{it} - the unit storage cost for product i in period t
- k_{itj} - the fixed setup cost for item i in period t on machine j
- u_{iltj} - the setup time from item i to item l in period t on machine j
- c_{tj} - the capacity of machine j in period t expressed in units of time
- a_{ij} - the consumption of machine j per unit of item i expressed in units of time

3. Variables:

- x_{itj} - the amount of product i produced in period t on machine j
- s_{it} - the inventory (stock) of product i in period t

and

$$y_{itj} = \begin{cases} 1, & \text{if machine } j \text{ is set up for product } i \text{ in period } t \\ 0, & \text{otherwise} \end{cases}$$

$$w_{iltj} = \begin{cases} 1, & \text{if machine } j \text{ is set up for product } l \text{ in period } t \\ & \text{and was set up for product } i \text{ in period } t-1 \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{if product } i \text{ is produced on machine } j \\ 0, & \text{otherwise} \end{cases}$$

The bilevel mixed 0-1 integer programming problem formulation is

$$min_z F(w) = \sum_{i=1}^{n} \sum_{l=1,l\neq i}^{n} \sum_{t=1}^{T} \sum_{j=1}^{2} u_{iltj} w_{iltj}$$

subject to

$$min f(x,y,s) = \sum_{i=1}^{n} \sum_{t=1}^{T} (\sum_{j=1}^{2} p_{itj} x_{itj} + h_{it} s_{it} + \sum_{j=1}^{2} k_{itj} y_{itj})$$

$$s_{i,t-1} + \sum_{j=1}^{2} x_{itj} = d_{it} + s_{it} \quad \forall i,t \tag{1}$$

$$\sum_{i=1}^{n} \sum_{l=1,l\neq i}^{n} u_{iltj} w_{iltj} + \sum_{i=1}^{n} a_{ij} x_{itj} \leq c_{tj} \quad \forall t,j \tag{2}$$

$$x_{itj} \leq M y_{itj} \quad \forall i,t,j \tag{3}$$

$$w_{iltj} \geq y_{i(t-1)j} + y_{ltj} - 1 \quad \forall i,l,t,j, i \neq l \tag{4}$$

$$w_{iltj} \leq y_{i(t-1)j} \quad \forall i,l,t,j, i \neq l \tag{5}$$

$$w_{iltj} \leq y_{ltj} \quad \forall i,l,t,j, i \neq l \tag{6}$$

$$\sum_{i=1}^{n} y_{itj} \leq 1, \quad \forall t,j \tag{7}$$

$$y_{itj} \leq z_{ij}, \quad \forall i,t,j \tag{8}$$

$$x_{it}, s_{it} \geq 0, \quad y_{itj}, w_{iltj}, z_{ij} \in \{0,1\}$$

where M is the upper bound on the production capacities. The constraints (1) represent the flow conservation constraints for each item in each period. Also, the constraints (2) describe the capacity limitations for each machine in each period. Here it is obvious that by minimizing the setup time, we have more time for the production. With more time for production, the feasibility set of the considered problem is larger and it is easier to satisfy the required demand. The constraints (4), (5) and (6) represent the quadratic constraints $w_{iltj} = y_{i(t-1)j} \cdot y_{ltj}$. If the machine j is set up for product i in period $t-1$, and it is set up for product l in period t, then $y_{i(t-1)j} = 1$, $y_{ltj} = 1$ and $w_{iltj} = y_{i(t-1)j} \cdot y_{ltj} = 1 \cdot 1 = 1$. This fact follows from the constraints (4), too. If the machine is set up for product i in period $t-1$, and it is not set up for product l in period t, then $y_{i(t-1)j} = 1$, $y_{ltj} = 0$ and $w_{iltj} = y_{i(t-1)j} \cdot y_{ltj} = 1 \cdot 0 = 0$. This fact follows from the constraints (6), too. Also, if the machine is not set up for product i in period $t-1$, and it is set up for product l in period t, then $y_{i(t-1)j} = 0$, $y_{ltj} = 1$ and $w_{iltj} = y_{i(t-1)j} \cdot y_{ltj} = 0 \cdot 1 = 0$. This fact follows from the constraints (5), too.At the end, if the machine is not set up for product i in period $t-1$, and it is not set up for product l in period t , then $y_{i(t-1)j} = 0$, $y_{ltj} = 0$ and $w_{iltj} = y_{i(t-1)j} \cdot y_{ltj} = 0 \cdot 0 = 0$. This fact follows from the constraints (5) and (6), too. The constraints (7) refer to a single mode of production.

This is a NP-hard problem and in order to solve it, we introduce heuristics based on tabu search. The motivation for the generation of the starting solutions was the fact as follows. If $z_{ij} = 1$, $\forall i, j$ (every product can be produced on every machine), the dimension of the problem is very large. We have many variables and many constraints. The feasibility set is large as possible. But, fixing z_{ij} to 0 or 1, we decrease the number of variables and constraints (see the constraints (3) and (8)). The proposed heuristics is described in the following section.

3 Heuristics

In order to generate the starting solutions of the tabu search, first we devise a way to assign the products to the machines for production. Let $\mathcal{N}_1$ be the set of all products that are going to be produced on machine 1, let $\mathcal{N}_2$ be the set of all products that are going to be produced on machine 2, and let $\mathcal{N}_0$ be an auxiliary set. We allow a product to be produced on both machines as well. Because of machine deterioration, we can assume without a loss of generality that setup times u_{iltj} are proportional, proportionality $r > 1$, i.e. $u_{il,t+1,j} = r u_{iltj}$ for a certain factor r. Therefore we fix time to be $t = 1$. Then we fix the machine. When we assign two products for production on the machine j, the sequence in which they are going to be produced is still not determined. Therefore we are considering the sum of the set-up times $u_{il1j} + u_{li1j}$. If for the sum of the setup times on the first machine for pair of products i and l inequality $u_{il11} + u_{li11} \leq k$ holds for certain value of k, then we assign both products i and l to be produced on the first machine, i.e. $i, l \in \mathcal{N}_1$. If the sum of the setup times for products i and l on the second machine is less than or equal to k, i.e. $u_{il12} + u_{li12} \leq k$, then we assign products i and l to be produced on the second machine, $i, l \in \mathcal{N}_2$. If there exists a product i not assigned to any machine, we put it to set $\mathcal{N}_0 = \{i \notin \mathcal{N}_1 \bigcup \mathcal{N}_2\}$. After we have done this, the products from the set $\mathcal{N}_0$ are allowed to be produced on the machines, randomly. Now that we have sets $\mathcal{N}_1$ and $\mathcal{N}_2$, values of variables z_{ij} are fixed since $\mathcal{N}_j = \{i : z_{ij} = 1\}$, and the size of the problem is reduced significantly.

Figure 1.

For different values of k we get different starting solutions, and thus we achieve diversification of the search. Shortly we denote the values of the decision variables in the K-th starting solutions by $x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]}$.

For each solution we construct exactly four neighborhood points. First we find the pair of products i and l assigned to be produced on the first machine for which the setup time u_{ilt1} is maximal. Then we construct the first neighborhood point by leaving product l to be produced on the first machine and sending product i to be produced on the second machine, and the second neighborhood point by leaving product i to be produced on the first machine and sending product l to be produced on the second machine. The other two neighborhood points are obtained by finding the pair of products assigned to be produced on the second machine for which the setup time u_{ilt2} is maximum, and by sending either product i or l to be produced on the first machine.

Figure 2.

The basic step of the search is being repeated until a fixed number of consecutive iterations is reached without providing any improvement of the leader's objective function.

To prevent reversal move, we introduce a tabu list in a form of the set TABU consisting of last L moves, where L is fixed and prescribed. The aspiration level $A(F(z))$ has been set to the value of the upper level objective function of the current solution z. When a solution z is found, it is allowed to go to a solution z' within neighborhood $N(z)$ by a tabu move only when the upper level objective value of $z\prime$ is larger than that of z. When a certain solution is found, we allow a move to another solution from its neighborhood by tabu move only if it satisfies aspiration criterion, i.e. if it leads to a smaller value of the leader's objective function than that of the aspiration level.

Let NS denote the total number of the generated starting points, and let K denote their counter, let NU denote the allowed number of uphill moves during the search part, and let I denote the counter to search among the neighborhood points.

The body of the heuristics looks like this:

Heuristics

Step 1 (Initialization)

Step 1A Set $K = 1$.

Step 1B Set $J = 0$. Set $I = 1$. If $K > NS$, go to Step 4. Otherwise, go to step 2A.

Step 2 (Choice)

Step 2A Generate K-th starting point $z_{[K]}$. Fix $z_{[K]}$ and obtain $x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}$ by solving the follower's problem (1)-(8). If $K = 1$, set

$$x^* = x_{[K]}, y^* = y_{[K]}, s^* = s_{[K]}, w^* = w_{[K]}, z^* = z_{[K]}$$

and set

$$F^* = F(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$$

Go to Step 2B.

Step 2B If $I \leq 4$, find I-th neighborhood point $z'_{[I]}$ and obtain $x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}$ by solving the follower's problem (1)-(8). Go to Step 2C.

If $I > 4$, go to Step 2D.

Step 2C If such obtained point is tabu and fails to satisfy the aspiration condition, set $I = I + 1$ and go to Step 2B. Otherwise, record

$$\overline{F}_{[I]} = F(x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}, z'_{[I]}),$$

set $I = I + 1$, and go to Step 2B.

Step 2D As new current solution $(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$ chose the point $(x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}, z'_{[I]})$ for which

$$\overline{\overline{F}} = \min\left\{\overline{F}_{[I]} : I = 1, ..., 4\right\}$$

Set B to an index of that point.

Step 3 (Update)

Step 3A Update tabu list $TABU$ and the value of the aspiration level. If $F^* \geq \overline{\overline{F}}$, set $J = J + 1$ and go to Step 3B.
If $F^* < \overline{\overline{F}}$, set $F^* = \overline{\overline{F}}$, set

$$x^* = x'_{[B]}, y^* = y'_{[B]}, s^* = s'_{[B]}, w^* = w'_{[B]}, z^* = z'_{[B]}$$

set $I = 1$, set $J = 0$ and go to Step 2B.

Step 3B If $J > NU$, set $K = K + 1$ and go to Step 1B. Otherwise, set $I = 1$ and go to Step 2B.

Step 4 (Termination) STOP. The solution is obtained.

Figure 3.

4 Computational results and conclusions

The heuristics is implemented in the AMPL programming language and uses CPLEX. All computations were performed on PC having Pentium IV 2.4 GHz processor and 1Gb RAM. The mixed-integer problem arising in Step

2B (the follower's problem (1)-(8)) is solved using CPLEX 8.0 mixed integer programming package, i.e. its branch and bound procedure.

Ten separate runs were made for each class of the problem. The data were generated according to uniform distribution using Excel, and taking into account the requirement that the costs increase as time passes, as well as requirements for setup costs. Namely, if the setup time required for switching from a product i to a product l is small, the setup time for switching from product l to product i is large and vice versa. Otherwise, the problem would not be the problem with sequence dependent setup times and would not describe the problem confronted by a pharmaceutical firm that we are considering here. Table 1 displays the size of the problem classes, as well as the CPU time (in seconds) required by the heuristics to reach termination.

Problem Class (T,n)	**No. of Variables**	**No. of Constraints**	**Average CPU time**
$(4,3)$	138	212	1.09375
$(4,4)$	216	376	1.73438
$(5,4)$	268	470	1.96875
$(6,4)$	320	564	2.54688
$(10,5)$	760	1470	4.9688
$(15,5)$	1135	2205	5.375
$(15,8)^*$	2536	5670	14.4844
$(20,5)$	1510	2940	6.09375

Table 1. Problem Classes together with CPU time

*The CPU time for the problem class $(15,8)$ is obtained by limiting the number of branch-and-bound nodes in CPLEX procedure for finding optimal solution of the mixed-integer follower's problem emerging in Step 2A and Step 2B (1)-(8). It should be mentioned that when the algorithm was run for this problem class without any limitations on the number of branch and bound nodes, it did not reach termination even after approximately 20 hours of working. However, the solution obtained by heuristics when a number of branch and bound nodes in MIP procedure for solving the follower's problem was limited was much better than the solution obtained after 20 hours of computing without limitations on the number of nodes, which shows that the heuristics itself had more influence on the quality of the solution.

As it can be seen from Table 1, the number of products to be produced most significantly influences the dimension of the problem in terms of the number of variables, as well as in terms of the number of constraints. Furthermore, the computational time increases significantly as n increases, which can be seen by comparing the CPU times for problem classes $(20, 5)$ and $(15, 8)$, where CPU time for problem class $(15, 8)$ is obtained by limiting the number of branch and bound nodes..

Table 2 shows values of objective functions F and f obtained by solving one instance of each class of the problem for the corresponding one-criterion problem obtained by taking into account only the leader's or the follower's objective function respectively, and by solving the bilevel problem.

Problem Class (T,n)	**One-criterion problem**		**Bilevel problem**
	Objective F	**Objective f**	**Heuristics**
$(4, 3)$	$F = 1$ $f = 25681$	$F = 22$ $f = 22271$	$F = 9$ $f = 24205$
$(4, 4)$	$F = 8$ $f = 52000$	$F = 25$ $f = 40112$	$F = 18$ $f = 42655.7$
$(5, 4)$	$F = 15$ $f = 64743$	$F = 39$ $f = 57345$	$F = 39$ $f = 57445.8$
$(6, 4)$	$F = 10$ $f = 98095$	$F = 88$ $f = 67414$	$F = 58$ $f = 75565.3$
$(10, 5)$	$F = 1$ $f = 230949$	$F = 99$ $f = 163144$	$F = 54$ $f = 207906$
$(15, 5)$	$F = 35$ $f = 404808$	$F = 293$ $f = 306161$	$F = 214$ $f = 344164$
$(15, 8)^*$	$F = 134$ $f = 787943$	$F = 282$ $f = 611075$	$F = 206$ $f = 632781$
$(20, 5)$	$F = 135$ $f = 680318$	$F = 757$ $f = 409362$	$F = 679$ $f = 425351$

Table 2. Comparison of the results for the one-criterion problem and the bilevel problem

Once again, the solution of the bilevel problem for problem class $(15, 8)$ is obtained by setting the limit on the number of branch and bound nodes in solving the follower's mixed-integer problem emerging in Step 2A and Step 2B.

Table 3 shows the initial partitions of the tabu search heuristics for solving the bilevel problem and optimal partitions of products to the machines obtained by the heuristics for the same instances of problem classes as in Table 2.

Problem Class (T,n)	**Initial partition**	**Optimal partition**
$(4,3)$	$\mathcal{N}_1 = \{2\}$ $\mathcal{N}_2 = \{1,3\}$	$\mathcal{N}_1 = \{2,3\}$ $\mathcal{N}_2 = \{1\}$
$(4,4)$	$\mathcal{N}_1 = \{1,2,3\}$ $\mathcal{N}_2 = \{4\}$	$\mathcal{N}_1 = \{1,2,3\}$ $\mathcal{N}_2 = \{2,4\}$
$(5,4)$	$\mathcal{N}_1 = \{1,2,3\}$ $\mathcal{N}_2 = \{4\}$	$\mathcal{N}_1 = \{1,2,3,4\}$ $\mathcal{N}_2 = \{1,2,3,4\}$
$(6,4)$	$\mathcal{N}_1 = \{1,3,4\}$ $\mathcal{N}_2 = \{2\}$	$\mathcal{N}_1 = \{1,2\}$ $\mathcal{N}_2 = \{3,4\}$
$(10,5)$	$\mathcal{N}_1 = \{2,5\}$ $\mathcal{N}_2 = \{1,3,4\}$	$\mathcal{N}_1 = \{5\}$ $\mathcal{N}_2 = \{1,2,3,4\}$
$(15,5)$	$\mathcal{N}_1 = \{3,4,5\}$ $\mathcal{N}_2 = \{1,2,5\}$	$\mathcal{N}_1 = \{3,4\}$ $\mathcal{N}_2 = \{1,2,3,4,5\}$
$(15,8)^*$	$\mathcal{N}_1 = \{2,3,4,5,6,8\}$ $\mathcal{N}_2 = \{1,3,5,6,7,8\}$	$\mathcal{N}_1 = \{2,3,4,5,6,7\}$ $\mathcal{N}_2 = \{1,3,5,6,8\}$
$(20,5)$	$\mathcal{N}_1 = \{2,5\}$ $\mathcal{N}_2 = \{1,3,4\}$	$\mathcal{N}_1 = \{2,4,5\}$ $\mathcal{N}_2 = \{1,2,3,4,5\}$

Table 3. Partition of products to the machines in the starting and optimal solution

Partitions obtained by solving the one-criterion problem always consist of $\mathcal{N}_1 = \{1,..,n\}$, $\mathcal{N}_2 = \{1,..,n\}$, regardless of whether solving the follower's or leader's problem is solved.

Finally, for each class of the problem the bilevel problem is solved by slightly modified heuristics obtained by randomly generating starting points of the tabu search in Step 2A instead of constructing them as described in Section 3 (Figure 1). The starting points of the modified heuristics are obtained by randomly assigning products to the machines for production. However, such heuristics mainly produces solutions of poorer quality, and sometimes, unlike the initial heuristics, it does not even reach the feasible solution at all, due to the limitations of the search.

5 Future work

The results showed that the greatest computational burden on heuristics lies in solving the mixed integer follower's problem appearing in Step 2A and Step 2B, (1)-(8). Moreover, the solution obtained by heuristics when limiting the number of branch and bound nodes in the mixed integer procedure for solving these problems proved to produce the overall bilevel problem solution of better quality and in less computational time. Therefore, in order to solve the problem instances of higher dimension, we want to develop heuristics based on Lagrange relaxation for solving the follower's problem in Step 2A and 2B (1)-(8), and then compare the results obtained in such a way with the results of heuristics proposed and studied in this paper.

References

[1] A. Allahverdi, J.N.D. Gupta, T. Aldowaisan, A review of scheduling research involving setup consideration, *Omega, The International Journal of Management Science* 27 (1999) 219-239

[2] O. Ben-Ayed, C.E. Blair, Computational difficulties of bilevel linear programming, *Technical notes* (1989) 556-560

[3] G.R. Bitran, H.H. Yanasse, Computational complexity of the capacitated lot size problem, *Management Science* 28 (1982) 1174-1186

[4] D. Cao, M. Chen, Capacitated plant selection in a decentralized manufacturing environment: A bilevel optimization approach", *European Journal of Operational Research* 169 (2006) 97-110

[5] A. Drexl, A. Kimms, Lot sizing and scheduling - Survey and extentions, *European Journal of Operational Research* 99 (1997) 221-235

[6] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, *Freeman, San Francisco, CA* (1979)

[7] F. Glover. Tabu Search. *Kluwer Academic Publishers*, Dordrecht, 1997.

[8] Diwakar Gupta, Thorkell Magnusson, The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times, *Computers & Operations Research* 32 (2005) 727-747

[9] Y. Hung, C. Chen, C. Shih, M. Hung. Using tabu search with ranking candidate list to solve production planning problems with setups, *Computers & Industrial Engineering* 45 (2003) 615–634

[10] B. Karimi, S.M.T. Fatemi Ghomi, J.M. Wilson, The capacitated lot sizing problem: a review of models and algorithms, *Omega, The International Journal of Management Science* 31 (2003) 365-378

[11] Z. Lukač, K. Šorić, V. Vojvodić Rosenzweig, Production Planning Problem with Sequence Dependent Setups, *Proceedings of 10th International Conference on Operational Research*, September 22-24, 2004, Trogir, Croatia, 25-33

[12] J.T. Moore, J.F. Bard, The mixed integer linear bilevel programming problem, *Operations research* 38 (1990) 911-921

[13] R.T. Sumichrast, Scheduling parallel processors to minimize setup time,*Compters & Operations Research* 14 (1987) 305-313

[14] L. Vicente, P. Calamai, Bilevel and multilevel programming: A bibliography review, *Journal of Global Optimization* 5 (1994) 291-306

[15] L. Vicente, G. Savard, J. Judice, Discrete Linear Bilevel Programming Problem, *Journal of optimization theory and application* 89 (1996) 597-614

[16] P. Wen, D. Huang, A simple Tabu search method to solve the mixed integer linear bilevel program, *European Journal of Operational Research* 88 (1996) 354-362

[17] Laurence A. Wolsey, MIP modelling of changovers in production planning and scheduling problems, *European Journal of Operational Research* 99 (1997) 154-165