Production Planning Problem with Sequence Dependent Setups

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

September 20, 2004

Abstract

Each of n items (products) is to be processed on two machines in order to satisfy known demands in each of T periods. Only one item can be processed on each machine at any given time. Each switch from one item to another requires sequence dependent setup time. The objects are to minimize the total setup time and the sum of the costs of production, storage and setup. Combining these two problems, we define a bicriteria mixed 0-1 integer programming problem called Production Planning Problem with Sequence Dependent Setups. We develop a heuristics based on tabu search for solving the problem. At the end, some computational results are presented.

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

1 Introduction

In recent years machine scheduling problems became very actual and important problems in the field of Operational Research. These are the following problems: suppose that we have to perform a number of jobs by using a number of machines. To perform the jobs, each of them must be processed in the order given by a sequence. The processing of a job requires processing time. Each machine can process only one job at a time. Given a cost function by which the cost of each possible solution can be measured, we want to find a processing order on each machine such that the corresponding cost is minimized.

A special class of scheduling problems arises from production planning. Its objective is to find an optimal schedule of jobs that allows production runs for several items (products, jobs) that are made on a single or several machines in such a way that setup and inventory costs or production completion time or average workload are minimized. Such a problem is called Lot Sizing Problem.

The classical Production Lot Sizing Problem is defined as follows: each of n items (products, jobs) is to be processed on a single machine in order to satisfy known demands in each of T periods. The object is to minimize the sum of the costs of production, storage and set up.

As manufacturing techniques become increasingly more complex, solving a lot sizing problem becomes much more difficult. First, limitations on machines lead to the so-called Capacitated Lot Sizing Problem. Second,the number of periods considered is another difficulty in solving lot sizing problems. Further, each switch from one item to another may require sequence dependent setup time. A setup may imply two kinds of machine consumption. One is setup cost, expressed in monetary terms; the other is setup time, consuming a certain amount of machine-hours.

In this paper, we consider the lot sizing problem with availability of two limited machines, existence of multiple items, multiple periods in the planning horizon, setup times and sequence dependent setup costs. The objects are to minimize the total setup time and the sum of the costs of production, backlog, storage and setup. A setup means deducting the setup time from the available machine-hours and deducting the setup costs from the corporate revenue which is trying to be maximized. Having the machine capacities expressed in time units, it is important to minimize the setup time because in this way there is more time available for the production. Having more time for the production, we are more able to satisfy demands in each of T periods. Also, in reality, there are many situations where the setup time is large, the setup cost small and vice versa. This problem can be formulated as a bicriteria mixed 0-1 integer programming problem called Production Planning Problem with Sequence Dependent Setups.

2 Formulation of the problem

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

1. Index:

- i - item type, $i = 1, 2, ..., n$
- j - machine type, $j = 1, 2, ..., m$
- 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
- a_{ij} - the consumption of machine j per unit of item i

3. Variables:

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

and

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

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

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

The bicriteria mixed 0-1 integer programming formulation is

$$F(x,y,s,w,z) = min \sum_{i=1}^{n} \sum_{l=1}^{n} \sum_{t=1}^{T} \sum_{j=1}^{m} u_{iltj} w_{iltj}$$

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

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

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

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

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

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

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

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

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

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

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 is obvious that minimizing the setup time, we have more time for the production. Having more time for the production, the feasibility set of the considered problem is larger and it is easier to satisfy the required demand. The constraints (4), (5) i (6) represent the quadratic constraints $w_{iltj} = y_{i(t-1)j} \cdot y_{itj}$. The constraints (7) refer to a single mode of production.

This is a NP-hard problem and in order to solve it, we introduce several heuristics based on tabu search. The motivation for the generation of the starting solutions is as follows. If $z_{ij} = 1$, $\forall i, j$ (every item 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 as large as possible. But, by fixing z_{ij} to 0 or 1, we decrease the number of variables and constraints (see the constraints (8) and (3)). Assuming the case of two machines, $m = 2$, the proposed heuristics are described in the following section.

3 Heuristics

We introduce four different versions of heuristics based on tabu search in order to solve the problem. All of them generate the starting solutions and construct the neighborhood points in the same manner. What makes them different is the aspiration criterion and the criterion for choosing the next solution.

In order to generate the starting solutions of the tabu search, first we devise a way to assign the items to the machines for production. Let $\mathcal{N}_1$ be the set of all items that are going to be produced on machine 1, let $\mathcal{N}_2$ be the set of all items that are going to be produced on machine 2, and let $\mathcal{N}_0$ be auxiliary set. We allow an item to be produced on both machines as well. Because of machine deterioration, without a loss of generality we can assume that setup times u_{iltj} are proportional by some coefficient of proportionality $r > 1$, i.e. $u_{il,t+1,j} = ru_{iltj}$ for certain factor r. Therefore we fix time to be $t = 1$. Then we fix the machine. If for the sum of the setup times on the first machine for pair of items i and l inequality $u_{il11} + u_{li11} \leq k$ holds for certain value of k, then we assign both items i and l to be produced on the first machine, i.e. $i, l \in \mathcal{N}_1$. If it doesn't hold, we put i and l to the set $\mathcal{N}_0$. We do the same for the second machine, i.e. if $u_{il12} + u_{li12} \leq k$, then we assign items i and l to be produced on the second machine, $i, l \in \mathcal{N}_2$. If not, we put i and l to set $\mathcal{N}_0$. After we've done that, the items from the set $\mathcal{N}_0$ are assigned to the machines randomly. Now all the values of variables z_{ij} are fixed, and size of the problem is reduced significantly. 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 solution 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 items i and l assigned to be produced on the first machine for which the setup time u_{ilt1} is maximum. Then we construct the first neighborhood point by leaving item l to be produced on the first machine and sending item i to be produced on the second machine, and second neighborhood point by leaving item i to be produced on the first machine and sending item l to be produced on the second machine. The other two neighborhood points are obtained by finding the pair of items assigned to be produced on the second machine for which the setup time u_{ilt2} is maximum, and by sending either item i or l to be produced on the first machine.

The basic step of the search is being repeated until a fixed number of consecutive iterations is reached without providing any improvement of certain objective function. The choice of the objective function depends on the version of the heuristics.

To prevent a move reversal, we introduce a tabu list in a form of the set TABU consisting of last L moves, where L is fixed and prescribed. The choice of the aspiration level has been set either to the value of the function F or f in a current solution, depending on the version of heuristics. When a certain solution is found, we allow a move to another solution from it's neighborhood by tabu move only if it satisfies aspiration criterion, i.e. if it leads to a smaller objective function value than that of the aspiration level.

Let NS denote the total number of generated starting points, 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 $J = 0$. 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 solve the problem by using objective function f. If $K = 1$, compute

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

Go to Step 2B.

Step 2B If $I \leq 4$, find I-th neighborhood point $z'_{[I]}$ and solve the problem by using objective function f in order to obtain $x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}$; go to Step 2C. Otherwise, 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 $(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$ chose the point $(x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}, z'_{[I]})$ for which

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

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. Otherwise, set $F^* = \overline{\overline{F}}$, 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 being obtained.

Heuristics H1

Aspiration criterion is determined according to the function F.

Heuristics H2

Aspiration criterion is determined according to the function f. Step 2A, 2C, 2D and 3A change into

Step 2A - H2 Generate K-th starting point $z_{[K]}$. Fix $z_{[K]}$ and solve the problem by using objective function f. If $K = 1$, compute

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

Go to Step 2B.

Step 2C - H2 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 - H2 As new $(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$ chose the point $(x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}, z'_{[I]})$ for which

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

Step 3A - H2 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. Otherwise, set $f^* = \overline{\overline{f}}$, set $I = 1$, set $J = 0$ and go to Step 2B.

Heuristics H3

Aspiration criterion is determined according to the function F. Step 2A, 2C, 2D and 3A are as in heuristics H2.

Heuristics H4

Aspiration criterion is determined according to the function $F + \lambda f$, where λ is a scaling factor. Step 2A, 2C, 2D and 3A change into

Step 2A - H4 Generate K-th starting point $z_{[K]}$. Fix $z_{[K]}$ and solve the problem by using objective function f. If $K = 1$, compute

$$G^* = Ff(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]}) + \lambda f(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$$

Go to Step 2B.

Step 2C - H4 If such obtained point is tabu and fails to satisfy the aspiration condition, set $I = I + 1$ and go to Step 2B. Otherwise, record

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

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

Step 2D - H4 As new $(x_{[K]}, y_{[K]}, s_{[K]}, w_{[K]}, z_{[K]})$ chose the point $(x'_{[I]}, y'_{[I]}, s'_{[I]}, w'_{[I]}, z'_{[I]})$ for which

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

Step 3A - H4 Update tabu list $TABU$ and the value of the aspiration level. If $G^* \geq \overline{\overline{G}}$, set $J = J + 1$ and go to Step 3B. Otherwise, set $G^* = \overline{\overline{G}}$, set $I = 1$, set $J = 0$ and go to Step 2B.

4 Computational results

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 is solved using CPLEX 8.0 mixed integer programming package, i.e. it's branch and bound procedure.

Ten separate runs were made for each class of the problem. Data were generated according to uniform distribution using Excel, and taking into account the requirements of increasing costs as time passes, as well as requirements on setup costs. 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 (n,T)	**No. of Variables**	**No. of Constraints**	**Average CPU time**
$(3, 4)$	138	262	8.4
$(4, 4)$	216	456	14.2
$(4, 5)$	268	562	16.86
$(4, 6)$	320	684	18.9

Table 1. Problem Classes together with CPU time

Table 2 shows comparison of the results obtained using different types of heuristics.

Problem Class (n,T)	**Avg. solution value of**	**Heuristics**			
		H1	**H2**	**H3**	**H4**
(3,4)	F	17	22	22	17
	f	23550	22271	22271	23550
(4,4)	F	25	28	28	25
	f	40112	38756	38756	40112
(4,5)	F	12	15	15	12
	f	10747	9176	9176	10747
(4,6)	F	65	88	88	65
	f	72973	67414	67414	72973

Table 2. Comparison of results using different types of heuristics

As Table 2 shows, when the heuristics favors function F (heuristics H1), then we obtain the solution having the smaller value of function F than the one obtained by heuristics H2 and H3, and vice versa, which proofs the existence of conflict between this two functions. By favoring either function F or f, the heuristics obtain two efficient solutions.

5 Future work

In order to solve more realistic problems, the problem instances of higher dimension will be considered and more computational results made. Also, some other heuristics based on Lagrange relaxation will be proposed and compered with the heuristics proposed and studied in this paper.

References

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

[2] F. Glover. Tabu Search - Part I. *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206 (1989)

[3] F. Glover. Tabu Search - Part II. *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4-32 (1990)

[4] Z. Lukač. Lot Sizing Problem With Equipment Replacement as a Bilevel Programming Problem. *Master Thesis*, Faculty of Economics, University of Zagreb, 2003.

[5] Z. Lukač, K. Šorić, V. Vojvodić-Rosenzweig. *Production Planning Problem with Different Setups as a Bilevel Programming Problem.* Paper presented at the EURO XX Conference, Rhodes, July 4-7, 2004

[6] 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