Journal of Applied Sciences 8 (17): 3032-3037, 2008
ISSN 1812-5654

The General Flowshop Scheduling Problem: Mathematical Models

S.J. Sadjadi, M.B. Aryanezhad and Mohsen Ziaee
Department of Industrial Engineering, Iran University of Science and Technology,
Narmak, Tehran, Iran

Abstract: In this study, consider three general flowshop scheduling problems: (1) with the objective function of the total weighted tardiness and the assumption of having ready times for jobs, (2) with the objective function of the makespan and the constraints of time lags and (3) with the makespan as objective and the constraints of the Sequence Dependent Setup Times (SDST). We present a Mixed Integer Linear Programming (MILP) model for each of them. The modeling formulations of this paper can generate the non-permutation schedules and cover the missing operations assumption. The implementation of the proposed formulations is demonstrated using some numerical examples.

Key words: Scheduling, non-permutation flowshop scheduling problem, time lags, sequence dependent setup times, mixed integer linear programming

INTRODUCTION

There are many mathematical formulations for modeling the flowshop scheduling problem in the literature (Stafford *et al*., 2002, 2005; Rios-Mercado and Bard, 2003). But these models generate only the permutation schedules. One reason is that in a m-machine n-job non-permutation flowshop scheduling problem, the total number of feasible schedules tends to $(n!)^m$ and in the permutation case, the number of feasible solutions is reduced to n! (Pugazhendhi *et al*., 2002), since in the problem under the permutation assumption, the sequence of jobs on all machines are the same. Therefore, modeling and solving the non-permutation problems are much more difficult. Another reason is that there is little improvement made by non-permutation schedules over permutation schedules with respect to completion-time based criteria. But the improvement is significant with respect to some special constraints such as SDST, missing operations of jobs, time lags and so on, or due-date based criteria like maximum tardiness or total weighted tardiness (Liao *et al*., 2006). However, there is a lack of mathematical programming models for the non-permutation flowshop scheduling problem with these assumptions and objectives in the literature.

An important assumption in the flowshop scheduling problem which is seen in many of the real problems is the missing operations of jobs which allow the jobs to pass some machines with any processing. Pugazhendhi and Rajendran have done extensive researches on this kind of problem and presented several heuristics and meta-heuristics for solving the problem with some objectives separately, for example see (Pugazhendhi *et al*., 2002; Rajendran and Ziegler, 2001). But there is not any mathematical formulation to model the problem. Although, it can use the regular flowshop scheduling models for this type of problem by setting the processing time of missing operations to zero, but they do not have the desired performance with respect to the objective function value, because of the specific type of this problem which would be described later. Therefore, the missing operations assumption is highlighted in this paper and its proposed models.

In this study, consider three non-permutation flowshop scheduling problems: (1) with the objective function of the total weighted tardiness and the jobs having ready times, (2) with the objective function of the makespan and the constraints of the start-start and stop-start time lags (Fondrevelle *et al*., 2006), (3) with the objective function of the makespan and the constraints of SDST. These problems are very applicable in the real world and usually denoted as $F/r_i/\Sigma w_i T_i$, $F/tl/C_{max}$, $F/ST_{sd}/C_{max}$, respectively (Allahverdi *et al*., 2008). Here, present a MILP model for each problem. The modeling formulations of this paper can generate the non-permutation schedules and cover the missing operations assumption. They are also linear models and linearity may lead to lower complexity and higher performance. Finally, the conclusion remarks of this study are presented at the end to summarize the contribution of the study.

Corresponding Author: Mohsen Ziaee, Department of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran Tel: +98-581-2222766 Fax: +98-21-77240482

General notations used throughout the study are:

I : The No. of jobs
K : The No. of machines
i : Denotes i'th job, i = 1,...,I
k : Denotes k'th machine, k = 1,...,K
T_{ik} : The processing time of job i on machine k,
x_{ijk} : Binary variable taking value 1 if i'th job on k'th machine is processed in j'th order and 0 otherwise. If $t_{ik} = 0$, then the operation really does not exist. However, a rank is given.
q_{ijk} : The completion time of job i on machine k if it is processed in j'th order.

PROBLEM 1

In this problem, the objective is to minimize the total weighted tardiness. It assume that the jobs may reach to the shop after the time zero. Let r_i, d_i and q_i be the ready time, the due date and the completion time (i.e. the completion time of the final operation of job i on the last machine) of job i, respectively, then the tardiness of job i is denoted as D_i, $D_i = \max(0, q_i - d_i)$. The proposed MILP model is as follows. M is an upper bound of the makespan (It has not to be tight and can be set to the sum of all processing times).

$$\min z = \sum_{i=1}^{I} (D_i w_i)$$

Subject to:

$$\sum_{j=1}^{I} (q_{ij(k+1)} - q_{ijk}) \geq t_{i(k+1)} \qquad i = 1,..,I; k = 1,..,(K-1) \quad (1)$$

$$\sum_{\substack{i \in 1..I, \\ t_{ik} \neq 0}} (q_{i(j+1)k} - t_{ik}.x_{i(j+1)k} - q_{ijk}) \geq 0 \qquad j = 1,..,(I-1); k = 1,..,K \quad (2)$$

$$D_i - \sum_{j=1}^{I} q_{ijK} + d_i \geq 0 \qquad i = 1,..,I \quad (3)$$

$$D_i \geq 0 \qquad i = 1,..,I \quad (4)$$

$$\sum_{i=1}^{I} x_{ijk} = 1 \qquad j = 1,..,I; k = 1,..,K \quad (5)$$

$$\sum_{j=1}^{I} x_{ijk} = 1 \qquad i = 1,..,I; k = 1,..,K \quad (6)$$

$$q_{ijk} - M.x_{ijk} \leq 0 \qquad \forall i, j, k \quad (7)$$

$$\sum_{j=1}^{I} q_{ij1} - t_{i1} - r_i \geq 0 \qquad i = 1,..,I \quad (8)$$

$$x_{ijk} \text{ belong to } \{0,1\} \qquad \forall i, j, k \quad (9)$$

Note that the sum $\sum_{j \in 1..I} q_{ijk}$ equals the completion time of operation (i, k). Similarly, $\sum_{i \in 1..I} q_{ijk}$ is the completion time of the operation ranked j on machine k. The constraint sets 1 and 2 insure that a job does not start on a machine until it finishes processing on the previous machine and its predecessor has completed processing on that machine. In the constraint set 2, if the processing time of operation (i, k) is equal to zero ($t_{ik} = 0$), then the constraint set 2 does not effect on this operation and so the completion time of the operation is determined based on the other constraints. The constraint sets 3 and 4 are related to the jobs tardiness described earlier. The constraint set 5 insures that in each machine, each sequence position is filled with only one job and the constraint set 6 insures that in each machine, each job is assigned to only one position in the job sequence. Constraint set 7 bind pairs (x_{ijk}, q_{ijk}); if $q_{ijk} > 0$, then $x_{ijk} = 1$. The constraint set 8 insures that each job on the first machine does not start earlier than its ready time.

Numerical example: Let us consider, the following instance of a 5-machine, 4-job problem:

$$t_{ik} = \begin{bmatrix} 2 & 0 & 3 & 1 & 5 \\ 3 & 5 & 7 & 4 & 2 \\ 2 & 1 & 5 & 6 & 9 \\ 0 & 10 & 5 & 8 & 4 \end{bmatrix}$$

$$w^T = [3 \quad 2 \quad 4 \quad 1]$$

$$r^T = [7 \quad 2 \quad 1 \quad 5] \qquad d^T = [21 \quad 27 \quad 15 \quad 39]$$

To compare the best solution with the best permutation solution of the problem, Here, first present the optimal permutation solution of the above test problem which is shown in Fig. 1. The objective value of this solution is equal to 66.

To understand the role of considering the missing operations assumption in improving the objective value, we also solve the problem 1 model without considering this assumption, i.e., the constraint set 2 changes as follows:

$$\sum_{i \in 1..I,} (q_{i(j+1)k} - t_{ik}.x_{i(j+1)k} - q_{ijk}) \geq 0 \qquad j = 1,..,(I-1); k = 1,..,K$$

The optimal solution of this model is shown in Fig. 2. In this solution, the objective value is equal to 60 and for two missing operations, i.e., (1, 2) and (4, 1), we have $q_{122} = 9$ and $q_{441} = 9$.

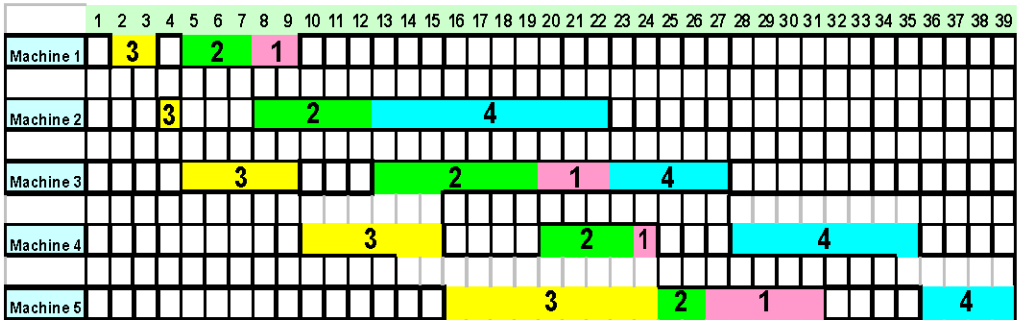

Fig. 1: The optimal permutation solution of the example of problem 1

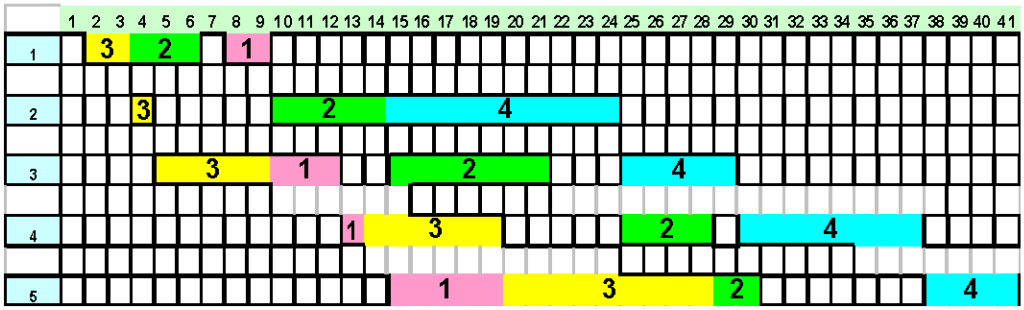

Fig. 2: The optimal solution of the example of problem 1 without considering the missing operations assumption

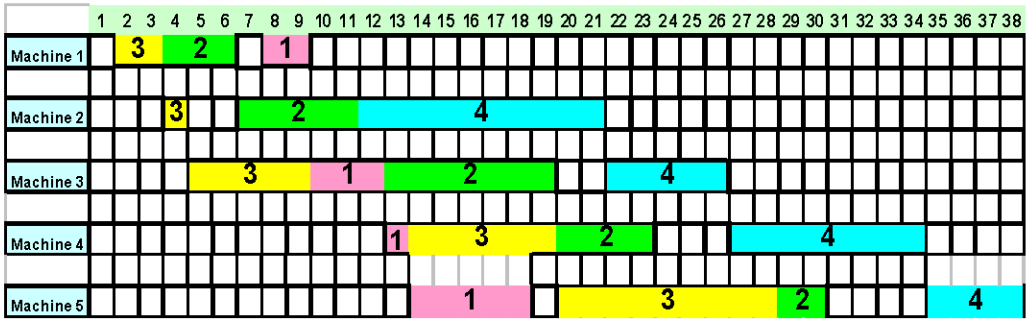

Fig. 3: The optimal solution of the example of problem 1

The optimal solution of the test problem is shown in Fig. 3 where, the objective value of this solution is equal to 58.

In the above optimal solution reported by a solver, q_{122} is equal to 9. Because the constraint set 2 of the model does not affect the missing operation (1, 2) and so the completion time of this operation is determined based on the other constraints. In this example, the optimal solution is a non-permutation schedule.

PROBLEM 2

In this problem, it is assumed that there are given time lags. These time lags are the waiting times between two consecutive operations of the same job. Here, assumes that the gap between the starting time of i'th job on k'th machine and its starting time on (k+1)'th machine (start-start time lag) must be greater or equal to $a_{i(k+1)}$ and the gap between the finishing time of i'th job on k'th machine and

its starting time on (k+1)'th machine (stop-start time lag) must be greater or equal to $b_{i(k+1)}$). Therefore, if job i starts on machine k at ($q_{ijk}-t_{ik}$) then it can start on machine k+1 at ($q_{ij'(k+1)}-t_{i(k+1)}$) such that,

$$a_{i(k+1)} \le (q_{ij'(k+1)} - t_{i(k+1)}) - (q_{ijk} - t_{ik}),$$

and similarly,

$$b_{i(k+1)} \le (q_{ij'(k+1)} - t_{i(k+1)}) - (q_{ijk})$$

The MILP model of this problem is as follows. y is the makespan, i.e., the largest completion time.

min z = y

subject to:

$$\sum_{i=1}^{I} q_{iik} - y \le 0 \qquad k = 1,..,K \qquad (10)$$

$$\sum_{i=1}^{I} (q_{i11} - t_{i1}.x_{i11}) \ge 0 \qquad (11)$$

$$\sum_{j=1}^{I} (q_{ij(k+1)} - q_{ijk}) - t_{i(k+1)} + t_{ik} - a_{i(k+1)} \ge 0 \qquad i,k: a_{i(k+1)} > 0 \qquad (12)$$

$$\sum_{j=1}^{I} (q_{ij(k+1)} - q_{ijk}) - t_{i(k+1)} - b_{i(k+1)} \ge 0 \qquad i,k: b_{i(k+1)} > 0 \qquad (13)$$

$$\sum_{j=1}^{I} (q_{ij(k+1)} - q_{ijk}) - t_{i(k+1)} \ge 0 \qquad i,k: a_{i(k+1)} = 0 \wedge b_{i(k+1)} = 0 \qquad (14)$$

And the constraints 2, 5 to 7 and 9 of the problem 1 model are added to the above constraints with no changes. Constraint set 10 affects the objective function. Note that the makespan may be not achieved on machine I because of missing operations on this machine. Hence these constraints involve all machines. Constraint (11) affects the completion time of the first operation on machine 1. The constraint sets 12 and 13 are related to the start-start and stop-start time lags described above. The constraint set 14 is the same as the constraint set 1 in the model of the first problem.

Numerical example: Consider a 5-machine 4-job test problem one more time where the values of processing times are the same as those of the example of problem 1 and the parameter values for the start-start and stop-start time lags constraints are as follows:

$$a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 5 & 0 & 0 & 3 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix} \end{array}, \quad b = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 8 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

If we have $a_{i(k+1)} > 0$ (or $b_{i(k+1)} > 0$) and $t_{ik} = 0$ or $t_{i(k+1)} = 0$, then this processing time of zero does not mean a missing operation, i.e., the operation really exists, but its processing time is very little and near to zero.

The optimal solution of the test problem is shown in Fig. 4.

And the optimal permutation schedule of the test problem is shown in Fig. 5.

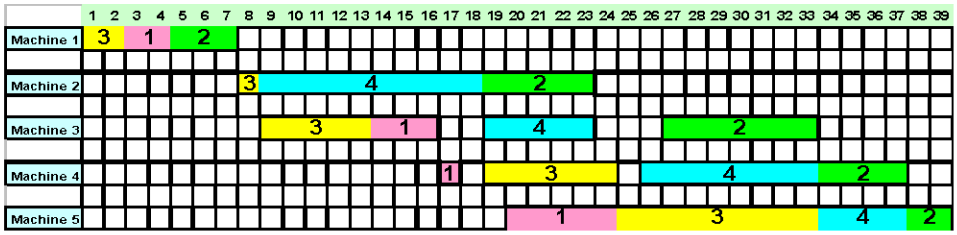

Fig. 4: The optimal solution of the example of problem 2

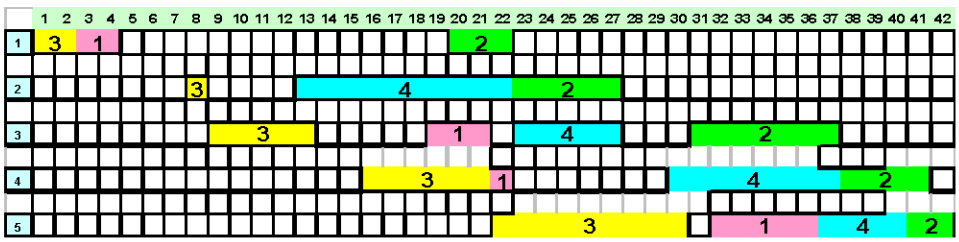

Fig. 5: The optimal permutation solution of the example of problem 2

PROBLEM 3

Here, it consider the case of SDST where the objective is to minimize the makespan. Let $s_{ii'k}$ be the setup time between two consecutive jobs i, i' on machine k. Therefore, the proposed model is formulated as follows:

min z = y

subject to:

$$\sum_{i=1}^{I} q_{ilk} - y \le 0 \qquad k = 1,..,K \qquad (15)$$

$$\sum_{i=1}^{I} (q_{i11} - t_{i1}.x_{i11}) \ge 0 \qquad (16)$$

$$\sum_{j=1}^{I} (q_{ij(k+1)} - q_{ijk}) - t_{i(k+1)} - \sum_{j=2}^{I}\sum_{i' \ne i} (s_{i'i(k+1)} x_{i'ij(k+1)}) \ge 0 \quad i = 1,..,I; k = 1,..,(K-1) \qquad (17)$$

$$\sum_{\substack{i \in 1..I, \\ t_{ik} \ne 0}} (q_{i(j+1)k} - t_{ik} x_{i(j+1)k} - q_{ijk} - \sum_{\substack{i' \in 1..I, \\ i' : i' \ne i, \\ t_{i'k} \ne 0}} (x'_{ii'jk} s_{i'ik})) \ge 0 \quad j = 1,..,(I-1); k = 1,..,K \qquad (18)$$

$$x_{i'(j-1)(k+1)} + x_{ij(k+1)} - x_{i'ij(k+1)} \le 1 \qquad i,i' = 1,..,I;\ j = 2,..,I; k = 1,..,(K-1) \qquad (19)$$

$$x_{i(j+1)k} + x_{i'jk} - x'_{ii'jk} \le 1 \qquad i,i' = 1,..,I;\ j = 1,..,(I-1); k = 1,..,K \qquad (20)$$

$$-x_{i'(j-1)(k+1)} - x_{ij(k+1)} + 2x_{i'ij(k+1)} \le 0 \quad i,i' = 1,..,I;\ j = 2,..,I; k = 1,..,(K-1) \qquad (21)$$

$$-x_{i(j+1)k} - x_{i'jk} + 2x'_{ii'jk} \le 0 \qquad i,i' = 1,..,I;\ j = 1,..,(I-1); k = 1,..,K \qquad (22)$$

$$x_{i'ij(k+1)}, x'_{ii'jk} \text{ belong to } \{0,1\} \qquad i,i', j = 1,..,I; k = 1,..,K \qquad (23)$$

And the constraints 5 to 7 and 9 of the problem 1 model are added to the above constraints with no changes. y is the makespan, i.e., the largest completion time. The constraint set 15 is related to the makespan. The constraint 16 insures that the first job on the first machine does not start earlier than the time zero. The constraint sets 17 and 18 insure that a job does not start on a machine until it has finished processing on the previous machine and its predecessor has completed processing on that machine and, that machine has set up for processing the job. The constraint sets 19 to 22 are added to the model for converting the nonlinear expressions to linear (Schrijver, 1998).

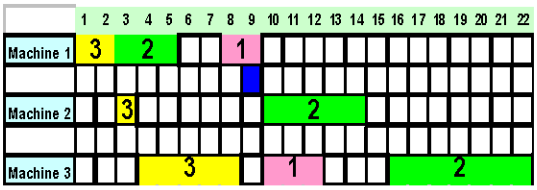

Fig. 6: The optimal solution of the example of problem 3

Numerical example: Here, consider a 3-machine 3-job example with the following data.

$$t_{ik} = \begin{bmatrix} 2 & 0 & 3 \\ 3 & 5 & 7 \\ 2 & 1 & 5 \end{bmatrix},$$

$S_{121} = 4,\ S_{123} = 3,\ S_{131} = 3,\ S_{132} = 5,$
$S_{211} = 2,\ S_{212} = 5,\ S_{231} = 4,\ S_{233} = 6,$
$S_{312} = 2,\ S_{313} = 1,\ S_{322} = 7.$

And all other $s_{ii'k}$ variables are equal to zero.

The optimal solution of this test problem is shown in Fig. 6.

CONCLUSION

In this study, it is presented some MILP models for some general flowshop scheduling problems. The proposed models of this paper can generate the non-permutation schedules and cover the missing operations assumption which can be found in many real world problems. The implementation of the proposed models on some numerical examples has shown that the proposed models could present better solutions compared with the optimal permutation solutions and even compared with the optimal non-permutation solutions which are obtained from the proposed models without considering the missing operations assumption.

REFERENCES

Allahverdi, A., C.T. Ng, T.C.E. Cheng and Y.M. Kovalyov, 2008. A survey of scheduling problems with setup times or costs. Eur. J. Operat. Res., 187: 985-1032.

Fondrevelle, J., O. Ammar and P. Marie-Claude, 2006. Permutation flowshop scheduling problems with maximal and minimal time lags. Comput. Operat. Res., 33: 1540-1556.

Liao, C.J., L.M. Liao and C.T. Tseng, 2006. A performance evaluation of permutation vs. non-permutation schedules in a flowshop. Int. J. Prod. Res., 44: 4297-4309.

Pugazhendhi, S., S. Thiagarajan, C. Rajendran and N. Anantharaman, 2002. Anantharaman performance enhancement by using nonpermutation schedules in flowline-based manufacturing systems. Comput. Ind. Eng., 44: 133-157.

Rajendran, C. and H. Ziegler, 2001. A performance analysis of dispatching rules and a heuristic in static flowshops with missing operations of jobs. Eur. J. Operat. Res., 131: 622-634.

Rios-Mercado R.Z. and J.F. Bard, 2003. The flow shop scheduling polyhedron with setup times. J. Combinatorial Optimization, 7: 291-318.

Schrijver, A., 1998. Theory of Linear and Integer Programming. 1st Edn., Wiley ISBN: 0 471 98232 6.

Stafford, E. F. and F.T. Tseng, 2002. Two models for a family of flowshop sequencing problems. Eur. J. Operat. Res., 142: 282-293.

Stafford, E.F., F.T. Tseng and J.N.D. Gupta, 2005. Comparative evaluation of MILP flowshop models. J. Operat. Res. Soc., 56: 88-101.