



MINIMIZING THE MAKESPAN IN A TWO-STAGE FLOWSHOP SCHEDULING PROBLEM WITH A FUNCTION CONSTRAINT ON ALTERNATIVE MACHINES

Chou-Jung Hsu

Lecturer, Department of Industrial Engineering and Management, Nan Kai Institute of Technology, Nantou, Taiwan 542, R.O.C., jrsheu@nkc.edu.tw

Wen-Hung Kuo

Assistant Professor, Department of Information Management, National Formosa University Yun-Lin, Taiwan 632, R.O.C.

Dar-Li Yang

Professor, Department of Information Management, National Formosa University Yun-Lin, Taiwan 632, R.O.C.

Maw-Sheng Chern

Professor, Department of Industrial Engineering and Engineering Management, National Tsing-Hua University, Hsin-Chu, Taiwan 300, R.O.C.

Follow this and additional works at: <https://jmstt.ntou.edu.tw/journal>



Part of the [Engineering Commons](#)

Recommended Citation

Hsu, Chou-Jung; Kuo, Wen-Hung; Yang, Dar-Li; and Chern, Maw-Sheng (2006) "MINIMIZING THE MAKESPAN IN A TWO-STAGE FLOWSHOP SCHEDULING PROBLEM WITH A FUNCTION CONSTRAINT ON ALTERNATIVE MACHINES," *Journal of Marine Science and Technology*. Vol. 14: Iss. 4, Article 3.

DOI: 10.51400/2709-6998.2082

Available at: <https://jmstt.ntou.edu.tw/journal/vol14/iss4/3>

This Research Article is brought to you for free and open access by Journal of Marine Science and Technology. It has been accepted for inclusion in Journal of Marine Science and Technology by an authorized editor of Journal of Marine Science and Technology.

MINIMIZING THE MAKESPAN IN A TWO-STAGE FLOWSHOP SCHEDULING PROBLEM WITH A FUNCTION CONSTRAINT ON ALTERNATIVE MACHINES

Acknowledgements

The authors would like to thank the two anonymous referees whose constructive comments have led to a substantial improvement in the presentation of the paper. This research is supported in part by the National Science Council of Taiwan, Republic of China, under grant number NSC-94-2213-E-212-013.

MINIMIZING THE MAKESPAN IN A TWO-STAGE FLOWSHOP SCHEDULING PROBLEM WITH A FUNCTION CONSTRAINT ON ALTERNATIVE MACHINES

Chou-Jung Hsu*, Wen-Hung Kuo**, Dar-Li Yang***, and Maw-Sheng Chern****

Key words: scheduling, function constraint, alternative machines

ABSTRACT

In this paper, we consider a two-stage flowshop scheduling problem with a function constraint on alternative machines. The objective is to minimize the makespan. We show that the proposed problem is NP-hard and provide some heuristic algorithms and computational experiments. In addition, from the experimental results, the modification of Johnson's rule combined with the First-Fit rule is the best heuristic algorithm of the proposed heuristic algorithms.

INTRODUCTION

A flowshop scheduling problem has been one of classical problems in production scheduling since Johnson [6] proposed the well-known Johnson's rule in the two-stage flowshop makespan scheduling problem. Yoshida and Hitomi [9] further considered the problem with setup times. Yang and Chern [8] extended the problem to a two-machine flowshop group scheduling problem. Ho and Gupta [5] and Cepek, *et al.* [2] proposed some efficient algorithms to solve various flowshop scheduling problems with dominant machines which can be found in many flexible manufacturing systems and process industries. However, these studies just considered one machine in each stage. Thus, mul-

iple machines in one or both stages were considered in other literatures. Kim, *et al.* [7] considered a batch scheduling problem for a two-stage flowshop with identical parallel machines at each stage. Brah and Loo [1] studied a flowshop scheduling problem with multiple processors. The objective is to minimize the makespan and mean flow time. These machines of each stage are identical in the previous studies. However, machines are not all identical at each stage in many real production systems. Thus, Futatsuishi, *et al.* [3] further studied a multi-stage flowshop scheduling problem with alternative operation assignments.

In this paper, we consider a two-stage flowshop scheduling problem with alternative machines. We focus on the functions of these alternative machines. Moreover, we show that the proposed problem is NP-hard and provide some heuristic algorithms and computational experiments.

Two contributions are made by this paper. First, the research of two-stage flowshop scheduling problem with a function constraint and alternative machines is new and practically useful. Second, the sequencing rules and heuristics developed in this paper are straightforward and easy to implement. We consider that this investigation provides not only a new model but also a new direction in the two-stage flowshop scheduling problems.

PROBLEM DESCRIPTION AND COMPLEXITY

The flowshop scheduling problem is described as follows. There are m types of jobs. The total number of all jobs is n . Each job must be processed at two stages. Moreover, the processing time of jobs of the same type at the first stage or the second stage may be different. There are m alternative machines at the first stage and a common processing machine at the second stage (see Figure 1). These alternative machines at the first stage have the following property. Without the loss of

Paper Submitted 10/20/05, Accepted 03/07/06. Author for Correspondence: Chou-Jung Hsu. E-mail: jrsheu@nkc.edu.tw.

*Lecturer, Department of Industrial Engineering and Management, Nan Kai Institute of Technology, Nantou, Taiwan 542, R.O.C.

**Assistant Professor, Department of Information Management, National Formosa University Yun-Lin, Taiwan 632, R.O.C.

***Professor, Department of Information Management, National Formosa University Yun-Lin, Taiwan 632, R.O.C.

****Professor, Department of Industrial Engineering and Engineering Management, National Tsing-Hua University, Hsin-Chu, Taiwan 300, R.O.C.

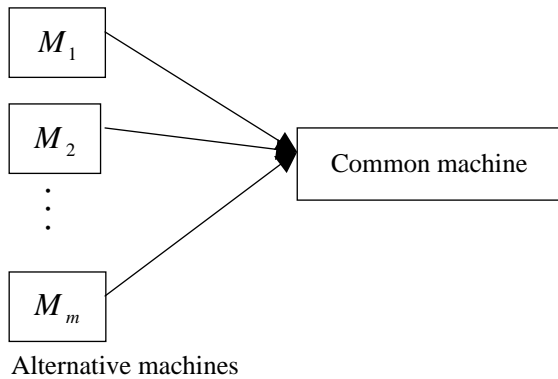


Fig. 1. The proposed problem model.

generality, it is assumed that the machines are numbered according to the technical functions, i.e. machine 1 can process jobs of type 1, machine 2 can process jobs of type 1 and type 2, ..., and machine m can process jobs of type 1, type 2, ..., and type m . We call these machines increasing dominating function machines. The objective of this scheduling problem is to minimize the makespan. Such scheduling problems can be applied to many real production systems. For example, a product has to undergo a manufacturing process and painting process. There are several manufacturing machines and one painting line in a production system. In addition, some of the manufacturing machines are old and the others are new. Generally, the new machines have more technical functions than the old ones. Therefore, the new machines can process more job-types than the old ones.

The problem is developed using the following notations. Additional notations will be introduced when needed throughout the paper.

n : the number of all jobs.

m : the number of alternative machines at the first stage.

M_k : the k -th machine which is numbered according to technical functions at the first stage, $k = 1, 2, \dots, m$.

M_c : the common processing machine at the second stage.

n_k : the number of jobs of type k , $k = 1, 2, \dots, m$. ($n_1 + n_2 + \dots + n_m = n$).

J_{ij} : the j -th job of type i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n_i$.

p_{ij} : the processing time of J_{ij} at the first stage.

q_{ij} : the processing time of J_{ij} at the second stage.

Theorem 1. The proposed problem is NP-hard in the strong sense.

Proof. We show that the 3-partition [4] problem reduces

to this problem. Considering the following well-known NP-complete problem:

3-partition: Give positive integers $a_1, a_2, \dots, a_{3n}, B$, and for each $J \in A = \{1, 2, \dots, 3n\}$ such that $B/4 < a_j < B/2$ and $\sum_{j \in A} a_j = nB$, does there exist disjoint sets A_1, A_2, \dots, A_n of A such that $\sum_{j \in A_1} a_j = \sum_{j \in A_2} a_j = \dots = \sum_{j \in A_n} a_j = B$?

For a given instance of 3-partition, $a_1, a_2, \dots, a_{3n}, B$, an instance of the proposed problem is constructed as follows. Let all jobs be the jobs of type 1.

$p_{1j} = a_j$ and $q_{1j} = 0$ for $j = 1, 2, \dots, 3n$, $m = n$, and $A = \{1, 2, \dots, 3n\}$, where $\sum_{j \in A} a_j = nB$.

We will show that 3-partition problem has a solution if and only if the above instance has an optimal schedule with the minimum makespan $C_{\max} = B$.

(\Rightarrow) If 3-partition problem has a solution, then there exist disjoint sets A_1, A_2, \dots, A_n of A such that $\sum_{j \in A_1} a_j = \sum_{j \in A_2} a_j = \dots = \sum_{j \in A_n} a_j = B$. Let the jobs corresponding to A_1 be scheduled on machine M_i ($i = 1, 2, \dots, n$) at the first stage (see Figure 2). Note that in such a situation, each machine exactly processes three jobs. Therefore, the maximum completion time on M_i ($i = 1, 2, \dots, n$) is B . That is, the maximum completion time of the first stage is B . In the meantime, the common machine is idle. However, the processing times of all jobs at the second stage is 0, thus the makespan $C_{\max} = B$.

(\Leftarrow) If 3-partition problem has no solution, we will show that the makespan of any schedule for the above instance is greater than B . Assume that $m = n$ and 3-partition has no solution. Then we have at least one

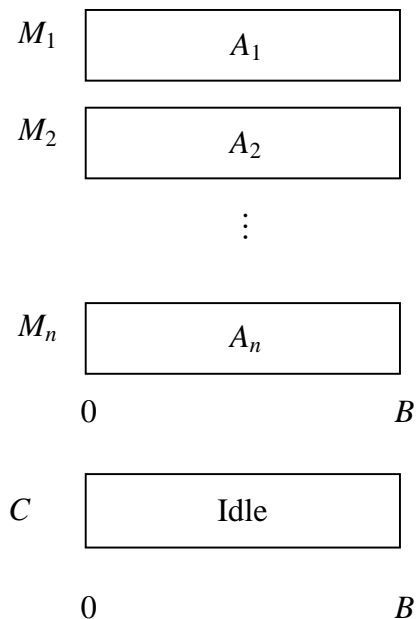


Fig. 2. Configuration of an optimal schedule.

disjoint sets of A , say A_l , in which $\sum_{i \in A_l} a_i < B$. Hence the sum of a_j in the other disjoint sets of A is greater than $(nB - B)$. Therefore, if the jobs corresponding to A_l be scheduled on one machine, then there at least exists one of the other $(n - 1)$ machines, on which the maximum completion time is greater than B . Thus, the makespan $C_{\max} = B$. Therefore, the makespan is greater than B if 3-partition problem has no solution.

This follows that 3-partition problem has a solution if and only if the optimal schedule of the above instance has the minimum makespan $C_{\max} = B$.

In this section, we will consider some heuristic algorithms with computational experiments. Our sequence-first, allocate-second heuristic approach decomposes the overall problem to exploit each of these two aspects. For the sequencing phase we have considered two rules to form a sequencing priority list. One is the Longest Processing Time (LPT) rule in which jobs are sequenced in non-increasing order of q_{ij} . The other is the modification of Johnson's rule. Second, we combine the two job sequencing methods with four dispatching rules to find an optimal or near-optimal schedule. That is, there are eight combinations of heuristic algorithms.

1. The modification of Johnson's rule

The steps of the modification of Johnson's rule are described as follows.

- Step 1. For the jobs yet to be sequenced, determine the minimum processing times of all p_{ij} and q_{ij} .
- Step 2. If the minimum is associated with p_{ij} , place the corresponding job in the earliest possible position in the sequencing priority list. In case of a tie, the job with the longest time of q_{ij} is chosen. If there is still a tie, choose the job with the least job type index, i.e. the number of machines to which the job can be assigned is maximal, and if still a tie, break it arbitrary. If the minimum is associated with q_{ij} , place the corresponding job in the latest possible position in the sequencing priority list. In case of a tie, select the job with the longest time of p_{ij} from the jobs tied. If still a tie, choose the job with the largest job type index, i.e. the number of machines to which the job can be assigned is minimal, if still tie again, break it arbitrary.
- Step 3. Repeat Step2 until all jobs are placed in the sequencing priority list.

2. Dispatching rules

Four dispatching rules are described as follows.

A. Type-Fix (TF):

- Step 1. Select the job in order of sequencing priority list, and assign the job to the corresponding machine according to the job type. i.e. a job of type 1 is assigned to machine 1, one of type 2 is assigned to machine 2, and so on.
- Step 2. Repeat Step 1 until all jobs are assigned.

B. First-Fit (FF):

- Step 1. Select the job from the sequencing priority list, say job j , and determine the base machine for job j . If the job j belongs to job type k , then the machine k is set to be the base machine for processing job j .
- Step 2. For machine $(k + 1)$ to machine m , find the first machine with the sum of processing times of jobs assigned so far less than that of the base machine. If there exists a such machine k' , then assign job j to the machine k' . Otherwise, assign the job to the base machine k .
- Step 3. Repeat Step 1 to 2 until all jobs are assigned.

C. Best-Fit (BF):

- Step 1. Select the job from the sequencing priority list, say job j , and determine the base machine for job j . If the job j belongs to job type k , the machine k is set to be the base machine for processing job j .
- Step 2. For machine $(k + 1)$ to machine m , find the machine k' with the minimum sum of processing time of jobs assigned so far. If the sum of machine k' is less than that of the base machine, assign job j to the machine k' . Otherwise, assign the job to the base machine k .
- Step 3. Repeat Step 1 to 2 until all jobs are assigned.

D. Random (RD):

- Step 1. Select the job from the sequencing priority list, say job j . If job j belongs to job type k , then choose a machine k' from machine k to machine m at random, and assign the job j to the machine k' .
- Step 2. Repeat Step 1 until all jobs are assigned.

It should be mentioned that the job sequence for the second stage is arranged in first come first service (FCFS) manner.

The following example demonstrates the usage of the four proposed dispatching rules.

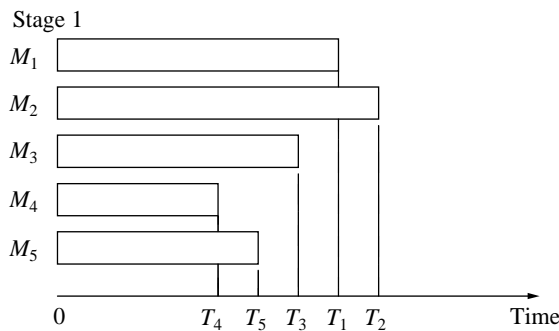


Fig. 3. The sum of processing times of all jobs assigned to each machine at stage 1.

Example 1. In a two-stage flowshop scheduling problem, there are five machines at the first stage. T_1 is the sum of processing times of all jobs assigned to machine M_1 , T_2 for machine M_2 , ... and T_5 for machine M_5 . Moreover, $T_4 < T_5 < T_3 < T_1 < T_2$ (see Figure 3). A job of type 2 is selected from the sequence. Which machine will the job be assigned?

Solution.

- A. **TF rule.** The job will be assigned to machine 2.
- B. **FF rule.** The job will be assigned to machine 3.
- C. **BF rule.** The job will be assigned to machine 4.
- D. **RD rule.** The job will be assigned to any one of available machines, i.e. the job will be assigned to machine 2, 3, 4 or 5.

3. The computational experiments

In order to evaluate the efficiency of the eight combinations of two job sequencing methods and four dispatching rules, we generate several groups of problems according to the following conditions.

- (1) m is equal to 2, 5, or 8.
- (2) p_{ij} is uniformly distributed over $[1, 20]$.
- (3) q_{ij} is uniformly distributed over $[1, 5]$, or $[1, 10]$.
- (4) n is equal to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250 or 300.
- (5) n_1, n_2, \dots, n_m are generated randomly according to the size of n .

There are 100 problems generated in each group. Hence, the total number of problems generated is 8,400. For each problem, a percentage of error $e = (C_h - C_{low})/C_{low}$ is computed. C_h is the makespan of a heuristic algorithm. C_{low} is the lower bound on the corresponding makespan and is estimated as theorem 2.

Theorem 2. A lower bound C_{low} of the makespan for the proposed problem is estimated as follows:

$C_{low} = \max \{LB1, LB2, LB3\}$, where

$$LB1 = \min_{i,j} \{p_{ij}\} + \sum_{i=1}^m \sum_{j=1}^{n_i} q_{ij}$$

$$LB2 = \sum_{j=1}^{n_m} p_{mj} + \min_j \{q_{mj}\}$$

$$LB3 = \min_j \{p_{mj}\} + \sum_{j=1}^{n_m} q_{mj}$$

Proof: Since each job has to go through two stages, the starting time of the common machine at the second stage begins after one of the jobs finishes its process at the first stage. Thus, the first term of $LB1$ is equal to the minimum waiting time of the common machine at the second stage, while the second term is equal to the total processing time of the common machine at the second stage. Thus, $LB1$ is a possible lower bound.

Since the machine M_m can process jobs of all types, it is highly possible that the total processing time of jobs on machine M_m is maximal. Thus, we derived lower bound $LB2$ and lower bound $LB3$ according to the total processing time on machine M_m . The first term of the lower bound $LB2$ is the maximum completion time of jobs on the machine M_m at the first stage. The second term represents the processing time of the job with the least processing time at the second stage. Therefore, $LB2$ is another possible lower bound.

Finally, the first term of $LB3$ is equal to the minimum waiting time of the common machine at the second stage, while the second term is equal to the total processing time of the jobs of type m on the common machine at the second stage. Hence, $LB3$ is also a possible lower bound. Thus, the maximum of the three possible lower bounds provides a valid lower bound.

To evaluate the overall performances of the heuristic algorithms, we compute the means of all the average percentages of errors for different numbers of jobs. There are 1,400 (100×14) test problems for each problem type. The results of computational experiments are shown in Table 1. From Table 1, it can be seen that the performances of the four dispatching rules combined with the modification of Johnson's rule are better than those combined with the LPT rule. In addition, the performance of the First-Fit rule is the best one of the four dispatching rules. It can also be seen that most of the mean percentages of errors decrease as the range of the processing times of jobs at the second stage increases. This implies that if the processing time variations of jobs at the second stage are larger, the heuristic algorithms may generate better solutions.

CONCLUSIONS

In this paper, we consider a two-stage flowshop

Table 1. The mean of all the average percentages of errors for each problem type

Sequencing		The LPT rule				A modification of Johnson's rule			
q_{ij}	NM	TF	FF	BF	RD	TF	FF	BF	RD
[1, 5]	2	0.1578	0.128	0.128	0.532	0.1577	0.1311	0.1311	0.5282
	5	0.0651	0.0317	0.0334	0.587	0.063	0.0318	0.0323	0.6024
	8	0.0254	0.0089	0.0092	0.2375	0.0227	0.0068	0.0069	0.2421
[1, 10]	2	0.0456	0.0366	0.0366	0.3707	0.0424	0.0335	0.0335	0.3655
	5	0.0084	0.0072	0.0072	0.0363	0.0025	0.0014	0.0015	0.0286
	8	0.0051	0.0039	0.004	0.0098	0.0012	0.0008	0.0008	0.0075
Average		0.0342	0.024	0.0243	0.197	0.0322	0.0228	0.0229	0.1971

Note: NM means # of machines at stage 1.

scheduling problem with alternative machines. We focus on the functions of these alternative machines. Moreover, we show that the proposed problem is NP-hard in the strong sense. Eight combinations of heuristic algorithms and the associated computational experiments are provided. From the results of the computational experiments, the performance of the modified Johnson's rule combined with the First-Fit dispatching rule is the best heuristic algorithm of the proposed algorithms.

In the future research, it is worthwhile to study other objectives, such as total completion time or maximum lateness. It is also interesting to investigate an extension of this problem, i.e. a flowshop scheduling problem with alternative machines at both stages.

ACKNOWLEDGEMENTS

The authors would like to thank the two anonymous referees whose constructive comments have led to a substantial improvement in the presentation of the paper. This research is supported in part by the National Science Council of Taiwan, Republic of China, under grant number NSC-94-2213-E-212-013.

REFERENCES

1. Brah, S.A. and Loo, L.L., "Heuristics for Scheduling in a Flow Shop with Multiple Processors," *European Journal of Operational Research*, Vol. 113, No. 1, pp. 113-122 (1999).
2. Cepek, O., Okada, M., and Vlach, M., "Nonpreemptive Flowshop Scheduling with Machine Dominance," *European Journal of Operational Research*, Vol. 139, No. 2, pp. 245-261 (2002).
3. Futatsuishi, Y., Watanabe, I., and Nakanishi, T., "A Study of the Multi-Stage Flowshop Scheduling Problem with Alternative Operation Assignments," *Mathematics and Computers in Simulation*, Vol. 59, No. 3, pp. 73-79 (2002).
4. Graey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, pp. 60-62 (1979).
5. Ho, J.C. and Gupta, J.N.D., "Flowshop Scheduling with Sominant Machines," *Computers and Operations Research*, Vol. 22, No. 2, pp. 237-246 (1995).
6. Johnson, S.M., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," *Naval Research Logistics Quarterly*, Vol. 1, No. 1, pp. 61-68 (1954).
7. Kim, J.S., Kang, S.H., and Lee, S.M., "Transfer Batch Scheduling for a Two-Stage Flowshop with Identical Parallel Machines at Each Stage," *Omega*, Vol. 25, No. 1, pp. 547-555 (1997).
8. Yang, D.L. and Chern, M.S., "Two-Machine Flowshop Group Scheduling Problem," *Computers and Operations Research*, Vol. 27, No. 10, pp. 975-985 (2000).
9. Yoshida, T. and Hitomi, K., "Optimal Two-Stage Production Scheduling with Setup Times Separated," *AIIE Transactions*, Vol. 11, No. 1, pp. 261-273 (1979).