Computers & Industrial Engineering 56 (2009) 306–311

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Heuristic algorithms for two-machine flowshop with availability constraints ☆

Li-Man Liao *, Chin-Hui Tsai

Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung 411, Taiwan

ARTICLE INFO

Article history:
Received 15 August 2006
Received in revised form 7 January 2008
Accepted 5 June 2008
Available online 13 June 2008

Keywords:
Scheduling
Two-machine flowshop
Machine availability
Makespan

ABSTRACT

The majority of the scheduling studies carry a common assumption that machines are available all the time. However, machines may not always be available in the scheduling period due to breakdown or preventive maintenance. Taking preventive maintenance activity into consideration, we dealt with the two-machine flowshop scheduling problem with makespan objective. The preventive maintenance policy in this paper was dependent on the number of finished jobs. The integer programming model was proposed. We combined two recent constructive heuristics, HI algorithm and H algorithm, with Johnson's algorithm, and named the combined heuristic H&J algorithm. We also developed a constructive heuristic, HD, with time complexities O(n^2). Based on the difference in job processing times on two machines, both H&J and HD showed good performance, and the latter was slightly better. The HD algorithm was able to obtain the optimality in 98.88% of cases. We also employed the branch and bound (B&B) algorithm to obtain the optimum. With a good upper bound and a modified lower bound, the proposed B&B algorithm performed significantly effectively.

1. Introduction

Most studies on scheduling assume that machines are available throughout the planning horizon. However, in practice, machines are not always available (Pinedo, 2002). That is, machines may not be available during the scheduling horizon due to breakdown (stochastic) or preventive maintenance (deterministic). Taking preventive maintenance activity into consideration, we dealt with a flowshop scheduling problem with limited machine availability.

In capital-intensive industry, production generally proceeds on a continuous basis and the availability of production centers at all time is very important. Nevertheless, maintenance activities have to be performed. Possible events that necessitate maintenance operations include: (1) the occurrence of a failure (failure-based maintenance); (2) the elapse of a certain amount of time or usage (use-based maintenance); and (3) the tested condition of a unit (condition-based maintenance) (Art, Knapp, & Lawrence, 1998). For recent surveys of problems with limited machine availability, refer to Sanlaville and Schmidt (1998) and Schmidt (2000). However, research on these problems has started only recently.

Johnson's rule is well known for the case of continuous machine availability, making the problem of minimizing the makespan easy to solve for two machines. Lee (1997) proved the problem to be NP-hard when an interval of non-availability (or hole, for short) occurs, and then developed a pseudo-polynomial dynamic programming algorithm to optimally solve the problem. Lee presented two heuristic algorithms. The first heuristic had a worst-case error bound of 1/2 for the case in which the hole occurred on the first machine. The second heuristic with a worst-case error bound of 1/3 for the case in which the hole occurred on the second machine. Similarly, Cheng and Wang (2000) studied the problem with the holes occurred on the first machine. Their heuristic had a worst-case error bound of 1/3. Breit (2004), studying the holes occurring on the second machine, proposed a heuristic with a worst-case error bound of 1/4. Cheng and Wang (1999) considered a special case where the availability constraint is imposed on each machine, but the two availability constraints are consecutive.

Lee (1999) considered the two-machine flowshop problem under the assumption that if a job cannot be finished before the next down period of a machine, then the job must be restarted partially when the machine becomes available again. His model was called semi-resumable. The model contained two important special cases: resumable where the job can be continued without any penalty and non-resumable where the job must be totally restarted. Lee also developed a pseudo-polynomial dynamic programming algorithm to optimally solve the problem and proposed heuristic algorithms with an error bound analysis.

Blazewicz, Breit, Formanowicz, Kubiak, and Schmidt (2001) studied a two-machine flowshop problem where machines are unavailable in given time intervals. They analyzed two constructive heuristics, Johnson's algorithm and look-ahead heuristic, and a heuristic based on simulated annealing (SA). Blazewicz et al. concluded that the SA-based heuristic is a more effective approach.

☆ This manuscript was processed by Area Editor John W. Fowler.

* Corresponding author. Tel.: +886 4 24363048; fax: +886 4 24363039.
E-mail address: Liao507@cjhunag.idv.tw (L.-M. Liao).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.06.003

Kubiak, Blazewicz, Formanowicz, Breit, and Schmidt (2002) proved that no polynomial time heuristic with a finite worst-case bound may exist when at least two holes are allowed to occur. Their study also showed that makespan minimization becomes NP-hard in the strong sense even if arbitrary number of holes occur on one machine only. Most important, Kubiak et al. proved two important properties of optimal schedules for the two-machine flowshop, a theory which serves as the framework of the current paper. They further developed a branch and bound algorithm based on the proposed properties.

Some papers stated that machines are available in time windows, which is true in computer systems. Aggoune, Mahdi, and Portman (2001) and Aggoune (2004) considered a flowshop problem with availability constraints, and provided two approaches to dealing with the maintenance activities: either starting time of the maintenance tasks are fixed or the maintenance tasks must be performed on a given time window. Aggoune et al. proposed a heuristic based on genetic algorithm to solve the makespan and the total weighted tardiness minimization problems. Aggoune developed a heuristic based on genetic algorithm and tabu search to solve the makespan minimization problem.

Most studies on machine availability take into consideration the elapse of a certain amount of time or usage (use-based maintenance). However, Dell'Amico and Martello (2001) considered a practical assembly line for printed circuit boards. They asserted that the machine is not available after processing a fixed number of jobs to allow for time precision adjustment of the machines. That is, the time periods of preventive maintenance activities are dependent on the number of finished jobs. Liao, Chen, and Lin (2007) provided an algorithm to solve two parallel machines where there are one or more unavailability intervals for each machine. The algorithm had exponential time complexities, but it could optimally solve the various-sized problems in reasonable computation time.

This paper dealt with the two-machine flowshop scheduling problem with makespan objective. The preventive maintenance policy was dependent on "the number of finished jobs". We combined two recent constructive heuristics, HI algorithm (Cheng & Wang, 1999, 2000) and H algorithm (Breit, 2004), with Johnson's algorithm, and named the heuristic H&J algorithm. We also developed a constructive heuristic, HD, which is based on the difference in the jobs' processing times on two machines. In order to evaluate the performance of H&J and HD, we further developed a branch and bound algorithm with a modified lower bound. Compared with the optimum solution, H&J was able to obtain optimality in 1562 out of 1600 instances (97.63%), and HD was able to obtain optimality in 1582 out of 1600 instances (98.88%). Both H&J and HD showed good performance, and the latter was slightly better.

The rest of the paper is organized as follows. Section 2 defines the terminology and constructs an integer programming model. Section 3 addresses basic properties of optimal solution and the development of two constructive heuristics, H&J and HD algorithms. A branch and bound algorithm (B&B algorithm) with a modified lower bound is constructed in Section 4. The performance of HD algorithm is evaluated in Section 5. The final section draws the conclusions of this work.

2. Terminology and integer programming model

Given n jobs to be processed in a two-machine flowshop, we define the following notations:

J_j	job j, $j = 1,\ldots,n$
$J_{[j]}$	the job at the jth position of schedule
M_i	machine i, $i = 1, 2$
$p_{i,j}$	processing time for J_j on M_i
t_i	length of hole on M_i
x_i	the number of finished jobs, the preventive maintenance policy on M_i
$h_{i,[j]}$	the index of M_i is available after the jth job. $h_{i,[j]} = 1$ if M_i is not available (hole) after the jth job; $h_{i,[j]} = 0$ if M_i is available after the jth job, i.e., $h_{i,[j]} = 1$ if j/x_i is integer; $h_{i,[j]} = 0$, otherwise.
$z_{j,k}$	the index of job j is scheduled at the kth position. $z_{j,k} = 1$ if job j is scheduled at the kth position; $z_{j,k} = 0$, otherwise.
$\sum$	schedule of jobs $1,\ldots,n$.
$J_k^{\sum}$	For a given schedule $\sum$ the holes partition jobs into disjoint subsets, the subset contains jobs completed on M_1 between starting points of the kth and the $(k+1)$th holes.
$C_{i,j}$	the completion time for J_j on M_i.
$C_{i,[l]}$	the completion time of the lth ranked job on M_i.
$C_{1,\max}$	makespan', $C_{1,\max} = C_{1,[n]} = \max\{C_{1,j}, j = 1,\ldots,n\}$.
$C_{\max}$	makespan, $C_{\max} = C_{2,[n]} = \max\{C_{2,j}, j = 1,\ldots,n\}$.
S_a	set of jobs before executing forward insert.
S_b	set of jobs before executing backward insert.
d_j	difference in processing time for J_j on M_1 and M_2, $d_j = p_{1,j} - p_{2,j}$.

In order to describe the problem clearly, an integer programming model is presented. The decision variables and auxiliary variables are $z_{j,k}$ and $C_{i,[l]}$, respectively. The parameters are $p_{i,j}$, t_i, x_i and $h_{i,[j]}$. The mixed integer programming model with $n^2 + 2n$ variables, including n^2 binary variables and $2n$ variables, and $5n$ constraints is formulated. The model is formulated as follows.

Objective function:

$$\min C_{2,[n]}$$

Subject to:

$$\sum_{j=1}^{n} z_{j,k} = 1; \quad k = 1,2,\ldots,n \tag{1}$$

$$\sum_{k=1}^{n} z_{j,k} = 1; \quad j = 1,2,\ldots,n \tag{2}$$

$$C_{1,[l]} = \sum_{k=1}^{l}\sum_{j=1}^{n}(z_{j,k} \times p_{1,j}) + \sum_{k=1}^{l}(h_{1,[k-1]} \times t_1); \quad l = 1,2,\ldots,n \tag{3}$$

$$C_{2,[l]} \geqslant C_{1,[l]} + \sum_{j=1}^{n}(z_{j,l} \times p_{2,j}); \quad l = 1,2,\ldots,n \tag{4}$$

$$C_{2,[l]} \geqslant C_{2,[l-1]} + h_{2,[l-1]} \times t_2 + \sum_{j=1}^{n}(z_{j,l} \times p_{2,j}); \quad l = 1,2,\ldots,n \tag{5}$$

$$h_{i,[j]} \in \{0,1\}; \quad i = 1,2; j = 1,2,\ldots,n \tag{6}$$

$$z_{j,k} \in \{0,1\}; \quad j,k = 1,2,\ldots,n \tag{7}$$

Constraint (1) specifies that exactly only one job can be scheduled to position k for any job j. Constraint (2) specifies that job j has to be scheduled to exactly one position. Constraint (3) defines the completion time of the lth ranked job on M_1. Constraints (4) and (5) insure that a job's completion time on M_2 is no earlier than that job's completion time on M_1 plus that job's processing time on M_2 and its previous job's completion time on M_2 plus that job's processing time on M_2.

3. The proposed solution methods

In this section, two critical properties of optimal schedules are described and two heuristics, H&J algorithm and HD algorithm, are proposed.

3.1. Basic properties

The two properties of optimal schedules, Lemmas 1 and 2, which were initially provided by Kubiak et al. (2002), were used