Computers & Industrial Engineering 114 (2017) 22–30

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing total tardiness in a two-machine flowshop scheduling problem with availability constraint on the first machine

Ju-Yong Lee[a,*], Yeong-Dae Kim[b]

[a] System Technology Team, Semiconductor Business, Samsung Electronics Co., Ltd., Yongin-Si, Gyeonggi-Do 17113, Republic of Korea
[b] Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Yuseong-gu, Daejeon 34141, Republic of Korea

ARTICLE INFO

Keywords:
Scheduling
Flowshop
Machine availability constraint
Branch and bound algorithm
Total tardiness

ABSTRACT

This paper deals with a two-machine flowshop problem in which the machine at the first stage requires preventive maintenance activities that have to be started within a given cumulative working time limit after the previous maintenance. That is, a maintenance activity can be started at any time unless the cumulative working time after the end of the previous maintenance exceeds the given limit. For the problem with the objective of minimizing total tardiness, we develop dominance properties and lower bounds for this scheduling problem as well as a heuristic algorithm, and suggest a branch and bound algorithm in which these properties, lower bounds, and heuristic algorithm are used. Computational experiments are performed to evaluate the algorithm and the results are reported.

1. Introduction

Over the past decades, a large number of researchers have studied flowshop scheduling problems. In a typical flowshop problem, m machines are arranged in a series and n jobs visit the machines in the same order, that is, each job consists of m operations and the kth operation of all jobs are processed on machine k for $k = 1, \ldots, m$. In most research on operations scheduling problems including flowshop problems, it is assumed that machines are continuously available at all times. However, this assumption oversimplifies the scheduling problems in real manufacturing systems since in reality there should be preventive maintenance, which has a significant impact on a variety of performance aspects such as productivity, reliability, and profitability. Note that delayed maintenance may lead to machine failure or deterioration in the quality of outputs. For these reasons, preventive maintenance tasks such as inspection, repair, replacement, cleaning, lubrication, adjustment and alignment are required to be performed for keeping machines in good condition and decrease the probability of machines' failures (Cui & Lu, 2017). Therefore, operations scheduling with maintenance is very important in operating manufacturing systems.

This study investigates a scheduling problem of minimizing total tardiness of jobs in a two-machine flowshop. In this problem, the machine in the first stage requires preventive maintenance within a given cumulative working time limit after the previous maintenance (since any delay of the maintenance activity increases the risk of machine failure significantly). That is, a maintenance activity can be (or should be) started at any time before the cumulative working time after the end of the previous maintenance exceeds a given time limit. This type of maintenance is called *flexible maintenance* since the start times of maintenance activities are not fixed but flexible. This scheduling problem can be denoted by $F2/fm/T$ according to the nomenclature of Graham, Lawler, Lenstra, and Rinnooy-Kan (1979), where $F2$, fm, and T represent the two-machine flowshop, flexible maintenance, and total tardiness, respectively. In the literature, constraints on machines due to maintenance activities are treated as a machine availability constraint.

For scheduling problems with preventive and flexible maintenance tasks, a large number of studies have been performed on a single-machine problem with various objectives. Among others, Qi, Chen, and Tu (1999), Yang, Ma, Xu, and Yang (2011), and Mashkani and Moslehi (2016) present branch and bound (BnB) algorithms, and Akturk, Ghosh, and Gunes (2003) and Gurel and Akturk (2008) propose heuristic algorithms for minimizing total completion time, while Mosheiov and Sarig (2009) develop a dynamic programming (DP) algorithm and a heuristic algorithm for minimizing total weighted completion times. Also, Chen (2006) gives mixed binary-integer programming models and develops a heuristic algorithm for minimizing mean flow time, Chen (2008a, 2008b) presents mixed binary-integer programming models for minimizing the total tardiness and for minimizing makespan, respectively, while Sbihi and Varnier (2008) suggest a heuristic and a BnB algorithm for minimizing maximum tardiness. Recently, Luo, Cheng, and Ji (2015), Zhu, Li, and Zhou (2015) and Ying, Ju, and Chen (2016) consider problems with flexible maintenance and various scheduling

* Corresponding author.
E-mail address: jy7433.lee@samsung.com (J.-Y. Lee).

http://dx.doi.org/10.1016/j.cie.2017.10.004
Received 4 March 2017; Received in revised form 2 October 2017; Accepted 3 October 2017
Available online 05 October 2017

measures.

For flowshop scheduling problems with machine availability constraints, most studies have focused on the makespan criterion (Ying, Chu, & Zuo, 2010). Also, the problems can be classified into two cases according to the characteristics of jobs: those of resumable and non-resumable jobs. In the resumable-job case, it is assumed that if a job is not completed before starting a maintenance activity, processing of the job can be continued after the maintenance. Unlike the resumable-job case, in the non-resumable job case, processing of a job that is preempted by a maintenance activity should be restarted from the beginning after the maintenance activity. In the following, we briefly review literature on two-machine flowshop problems of the three cases with the makespan measure.

Under the assumption of the resumable-job case, Lee (1997) proves that the problem is NP-hard when a maintenance activity is performed on only one machine. Also, Lee (1997) develops a pseudo-polynomial dynamic programming algorithm and two heuristic algorithms. One heuristic, which is for a case in which a maintenance activity is performed on machine 1, has a worst case error bound of 1/2 and the other heuristic, for a case with a maintenance activity performed on machine 2, has a worst case error bound of 1/3. Later, for the first case, Cheng and Wang (2000) propose a heuristic algorithm with a worst case error bound of 1/3, while Breit (2004) presents an improved heuristic with a worst case error bound of 1/4 for the second case. On the other hand, for problems in which multiple maintenance activities are performed, Błażewicz, Breit, Formanowicz, Kubiak, and Schmidt (2001) develop heuristic algorithms and Kubiak, Błażewicz, Formanowicz, Breit, and Schmidt (2002) propose a branch and bound (BAB) algorithm. In addition, Kubzin, Potts, and Strusevich (2009) present a 3/2-approximation algorithm for a problem with multiple maintenance activities on the first machine.

Under the assumption of the non-resumable job case, Allaoui, Artiba, Elmaghraby, and Riane (2006) consider a problem in which a maintenance activity is required on machine 1 only and develop a dynamic programming algorithm and a heuristic algorithm, Allaoui, Lamouri, Artiba, and Aghezzaf (2008) consider that one of the two machines should be maintained once before a given maximum allowed continuously working time and propose some optimal solution properties, while Yang, Hsu, and Kuo (2008) propose a heuristic algorithm to solve a problem in which a maintenance activity is required on each of the two machines before completing a given number of jobs. Also, Hnaien, Yalaoui, and Mhadhbi (2015) investigates two mixed-integer programming models and a BAB algorithm for the problem in which a maintenance activity is performed at a given start time. On the other hand, Espinouse, Formanowlcz, and Penz (1999, 2001), Wang and Cheng (2001) and Cheng and Liu (2003) consider the non-resumable job case in a special flowshop, *no-wait flowshop*, in which jobs have to be processed on machine 2 immediately after they are completed on machine 1 without waiting.

In this paper, we present an optimal-solution algorithm minimizing total tardiness in a two-machine flowshop scheduling problem in which the machine in the first stage requires preventive maintenance within a given cumulative working time limit after the previous maintenance. Here, it is assumed that: (1) the cumulative working time between two successive maintenance activities should not exceed a given time limit; (2) the maximum processing time of all jobs is shorter than the cumulative working time limit; (3) the time required to perform maintenance activities are the same and given; (4) all jobs are ready at the beginning of the scheduling horizon; and (5) all jobs are not resumable and preemption of jobs is not allowed. Note that since preemption is not permitted in the problem, one and only one schedule can be obtained from a permutation of jobs, or a sequence of jobs. Therefore, only $n!$ sequences of jobs need to be considered to find an optimal solution of the problem.

We can prove that the problem considered in this study, $F2/fm/T$, is strongly NP-hard, since the ordinary two-machine flowshop total tardiness problem, $F2//T$, is NP-hard (Koulamas, 1994). Note that $F2/fm/T$ is the same as $F2//T$ if the cumulative working time limit is extremely large or the time needed to carry out maintenance activities are zero.

This paper is organized as follows. In the next section, we give a mixed integer-linear programming formulation for the problem considered in this study. In Section 3, we develop dominance properties for the problem. In Section 4, we suggest a branch and bound (BAB) algorithm, with lower bounds on the total tardiness of partial schedules used in the BAB algorithm, and a heuristic algorithm to obtain an initial upper bound for the BAB algorithm. In Section 5, computational experiments are performed to evaluate the performance of the suggested algorithms. In Section 6, we conclude this study with a short summary.

2. Mathematical formulation

We give a mathematical formulation of the problem considered in this study. In the formulation and throughout this paper, the following notation is used.

Indices and parameters

J	set of jobs, $J = \{1, \ldots, n\}$
i, j	indices for jobs
$[r]$	index for the rth job in a sequence, $r = 1, \ldots, n$
k	index for machines, $k = 1, 2$
p_{ik}	processing time of job i on machine k
d_i	due date of job i
W	cumulative working time limit
h	time required to perform a maintenance activity
L	a large number

Decision variables

$s_{[r]k}$	nonnegative real variable that is equal to the starting time of the rth job on machine k
$c_{[r]}$	nonnegative real variable that is equal to the completion time of the rth job
$T_{[r]}$	nonnegative real variable that is equal to the tardiness of the rth job
$e_{[r]}$	nonnegative real variable that is equal to the cumulative working time when the rth job is completed on machine 1
x_{ir}	binary variable that is equal to 1 if job i is the rth job in a sequence, 0 otherwise
$y_{[r]}$	binary variable that is equal to 1 if a maintenance task is performed immediately after the rth job on machine 1, 0 otherwise

Now, a mixed integer-linear programming formulation is given.

$$[\text{P}]\ \text{Minimize} \quad \sum_r T_{[r]}$$

$$\text{subject to} \quad \sum_r x_{ir} = 1 \quad \forall\, i \in J \tag{1}$$

$$\sum_i x_{ir} = 1 \quad \forall\, r \tag{2}$$

$$\sum_i p_{i1} x_{i1} = e_{[1]} \tag{3}$$

$$e_{[r-1]} + \sum_i p_{i1} x_{ir} - L y_{[r-1]} \leqslant e_{[r]} \quad r = 2,\ldots,n \tag{4}$$

$$\sum_i p_{i1} x_{ir} - L(1 - y_{[r-1]}) \leqslant e_{[r]} \quad r = 2,\ldots,n \tag{5}$$

$$e_{[r]} \leqslant W \quad \forall\, r \tag{6}$$

$$s_{[r]1} + \sum_i p_{i1} x_{ir} + h y_{[r]} \leqslant s_{[r+1]1} \quad r = 1,\ldots,n1 \tag{7}$$