Computers & Industrial Engineering 101 (2016) 361–371

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Parallel machine scheduling with maintenance activities

Jaewook Yoo, Ik Sun Lee *

School of Business, Dong-A University, Seo-gu, Busan 602-760, Republic of Korea

ARTICLE INFO

Article history:
Received 14 August 2013
Received in revised form 18 September 2016
Accepted 19 September 2016
Available online 21 September 2016

Keywords:
Scheduling
Maintenance
Parallel machine
Problem complexity

ABSTRACT

This paper considers a problem of scheduling on parallel machines where each machine requires maintenance activity once over a given time window. The objective is to find a coordinated schedule for jobs and maintenance activities to minimize the scheduling cost represented by either one of several objective measures including makespan, (weighted) sum of completion times, maximum lateness and sum of lateness. The problem is proved to be NP-hard in the strong sense in each case of the objective measures. Some restricted cases of the problem are also characterized for their complexities, for which the associated dynamic programming algorithms are derived.

1. Introduction and problem description

The majority of machine scheduling models often assume that machines are available all the time for processing jobs over their associated planning horizon. However, this assumption is not realistic in many manufacturing situations, since machines require maintenance activity periodically to prevent malfunctions. During the associated maintenance activity machines are not available for processing jobs. Maintenance encompasses activities including installation, vehicles, equipment, or some physical assets enabling effective work. Preventive maintenance is an activity that a priori prevents potential faults resulting in malfunctions and also prevents critical non-availability of the system. Note that maintenance costs cover a big percentage of the total operating costs, making it very reasonable to include maintenance activities in the production schedule (Ángel-Bello, Álvarez, Pacheco, & Martínez, 2011).

In the airline industry, planned maintenance activities can reduce production time by as much as 15% (Laalaoui & M'Hallah, 2016). Moreover, especially in semiconductor manufacturing, it is often observed that machines are idle while waiting for maintenance personnel to do preventive maintenance, even though jobs are waiting. Thus, the operations managers have to create their production schedule carefully so as to minimize their costs while avoiding unexpected resource unavailability. Obviously, careful coordination between maintenance activity and job processing would result in a better schedule, which is the motivation for this study. Here, the authors consider a coordinated scheduling model that takes into account such associated machine maintenance activities.

In the literature, scheduling problems with maintenance activities incorporated can be classified into "*fixed*" and "*coordinated*" models. The first model considers the maintenance activity durations, which are known and fixed in advance, so that the starting and completion times of the maintenance activity are given. The problem of scheduling jobs with this type of maintenance has often been referred to in the literature as "*scheduling with machine availability constraints*". Ángel-Bello et al. (2011), Hfaiedh, Sadfi, Kacem, and Hadj-Alouane (2015), Laalaoui and M'Hallah (2016), Molaee, Moslehi, and Reisi (2011), and Sadfi, Penz, Rapine, Blazewicz, and Formanowicz (2005) have studied various single machine problems subject to various types of machine availability constraints. Fu, Huo, and Zhao (2011), Gedik, Rainwater, Nachtmann, and Pohl (2016), Liao and Sheen (2008), Mellouli, Sadfi, Chu, and Kacem (2009), and Wang and Cheng (2015) have studied various parallel machine problems allowing various types of unavailable intervals for machines. Cheng and Wang (1999), Cheng and Wang (2000), Kubiak, Blazewicz, Formanowicz, Breit, and Schmidt (2002), Kubzin, Potts, and Strusevich (2009), and Lee (1997, 1999) have studied a two machine flow shop problem allowing various types of unavailable intervals for machines.

The second model is concerned with simultaneously determining when to conduct each maintenance activity and when to process each job. Some research has been conducted on scheduling maintenance activities and jobs jointly. For example, Graves and Lee (1999) and Cassady and Kutanoglu (2003) have studied single machine problems allowing maintenance activities to be scheduled jointly with jobs. Aggoune (2004) has studied a flowshop machine

* Corresponding author. Tel.: +82 51 200 7425.
E-mail addresses: jyoo@dau.ac.kr (J. Yoo), lis1007@dau.ac.kr (I.S. Lee).

http://dx.doi.org/10.1016/j.cie.2016.09.020
0360-8352/

problem allowing maintenance activities to be made within any given time window. Costa, Cappadonna, and Fichera (2016), Lee and Chen (2000), and Sun and Li (2010) have studied some parallel machine problems, subject to the constraint that maintenance activity on each machine should be made within a given time window. Specifically, Lee and Chen (2000) have studied a parallel machine problem to minimize the weighted sum of completion times of jobs. They have proved that the problem is NP-hard and have derived a branch and bound algorithm based on the column generation approach. Sun and Li (2010) have researched two two-machine parallel machines with the makespan or sum of completion times. Costa et al. (2016) have developed a genetic algorithm for a parallel machine problem.

This paper considers a coordinated scheduling model on parallel machines where each machine requires maintenance activity once over a given time window, as in Lee and Chen (2000). Moreover, two different maintenance activities are considered. The first one allows more than one machine to be put under maintenance simultaneously if necessary and is called "*independent case*". The second one, called "*dependent case*", allows only one machine to be put under maintenance at any time point due to insufficient maintenance resources (equipment or person); hence, the maintenance activity duration on machines cannot be overlapped onto each other. The model proposed here considers several different objectives of minimizing scheduling costs, each being represented by either one of several objective measures including makespan, (weighted) sum of completion times, maximum lateness and sum of lateness. Each of these scheduling problems is proved to be NP-hard in the strong sense and then some solution properties are characterized. Therewith, solution algorithms are derived using a dynamic programming (DP) approach. A few restricted cases of the problems are also analyzed for their complexities.

The proposed problem is stated in detail as follows: there are n jobs available at time zero to be scheduled on m identical parallel machines without preemption. Maintenance on each machine must be completed exactly once within the given time length T, that is, during the given time window $[0,T]$, where the maintenance activity requires a maintenance time length t, while any job processing is allowed after time T. It is assumed that $T \geqslant t$ and $T \geqslant mt$ in the independent and dependent cases, respectively. processing time, weight, due date, and completion time of job j are denoted by p_j, w_j, d_j and C_j, respectively. It is assumed that t, T, p_j's, w_j's and d_j's have integer values. Moreover, this paper does not allow any preemption, so that a job should not be allowed to start until completing its associated maintenance activity if there is not enough time to complete any job processing before starting maintenance activity on the machine, which may incur an occurrence of machine idle time.

The standard classification scheme for scheduling problems (Pinedo (1995)) $\alpha|\beta|\gamma$ is adapted in this paper where α indicates the scheduling environment, β describes the job characteristics or restrictive requirements, and γ defines the objective function to be minimized. Accordingly, the proposed problem is represented by an identical parallel machines problem with α = "P". For β, the problem considers "*ind*", "*dep*", "$p_j = p$", "$d_j = d$" and "$m = q$" constraints, where "*ind*", "*dep*", "$p_j = p$", "$d_j = d$" and "$m = q$" indicate the independent case, the dependent case, all identical processing times case, all identical due dates case, and the q identical parallel machines case, respectively. Moreover, for γ, the objective function of the proposed problem may be represented by one of the following:

$$C_{max} = \max_{1\leqslant j\leqslant n} C_j \quad \text{(makespan)},$$

$$\sum_{j=1}^{n} C_j = \text{sum of completion times},$$

$$\sum_{j=1}^{n} w_j C_j = \text{weighted sum of completion times},$$

$$L_{max} = \max_{1\leqslant j\leqslant n}\{C_j - d_j, 0\} \quad \text{(maximum lateness)},$$

$$\sum_{j=1}^{n} L_j = \sum_{j=1}^{n} \max(C_j - d_j, 0) \quad \text{(sum of lateness)}.$$

Table 1 provides the complexities of all of the tested scheduling problems associated with various objective measures. In the table, the complexity orders represent the computational time complexities of the associated DP algorithms, which are derived in Sections 2.1 and 4, where R_1, R_2, R_3 and R_4 are derived in Theorems 10 and 11 in Section 2.

2. General case analysis

This section will prove that the problems P|ind|γ and P|dep|γ are NP-hard in the strong sense, where $\gamma \in \left\{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\right\}$.

Table 1
Complexities of the scheduling problems.

Objective	Additional problem characteristics	Independent case	Dependent case
C_{max}		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m = q\ (\geqslant 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j = p$	$O(n)$	*Unknown*, $O(m(n+1)^{3m})$
$\sum_{j=1}^{n} C_j$		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m = q\ (\geqslant 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j = p$	$O(n)$	*Unknown*, $O(m(n+1)^{3m})$
$\sum_{j=1}^{n} w_jC_j$		Unary NP-hard, $O(nmR_1^{2m}R_2{}^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m = q\ (\geqslant 2)$	Binary NP-hard, $O(nqR_1^{2q}R_2{}^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j = p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
L_{max}		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m = q\ (\geqslant 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j = p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
	$d_j = d$	Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
$\sum_{j=1}^{n} L_j$		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m = q\ (\geqslant 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j = p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
	$d_j = d$	Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$