Computers & Industrial Engineering 60 (2011) 602–605

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Unrelated parallel-machine scheduling with deteriorating maintenance activities

T.C.E. Cheng [a], Chou-Jung Hsu [b], Dar-Li Yang [c,*]

[a] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[b] Department of Industrial Engineering and Management, Nan Kai University of Technology, Nan-Tou 542, Taiwan
[c] Department of Information Management, National Formosa University, Yun-Lin 632, Taiwan

A R T I C L E I N F O

Article history:
Received 15 May 2010
Received in revised form 10 September 2010
Accepted 31 December 2010
Available online 8 January 2011

Keywords:
Scheduling
Deteriorating maintenance activity
Assignment problem
Total completion time
Total machine load

A B S T R A C T

We study the problem of unrelated parallel-machine scheduling with deteriorating maintenance activities. Each machine has at most one maintenance activity, which can be performed at any time throughout the planning horizon. The length of the maintenance activity increases linearly with its starting time. The objective is to minimize the total completion time or the total machine load. We show that both versions of the problem can be optimally solved in polynomial time.

1. Introduction

Production scheduling and preventive maintenance planning are fundamental operational problems in the manufacturing industry. The simultaneous consideration of these two problems has received increasing attention from the scheduling research community and the corresponding scheduling problem is commonly known as "machine scheduling with availability constraints". Most of the studies on this problem assume that the maintenance time is constant and known in advance. However, some studies assume that the maintenance time is constant while the starting time of the maintenance activity is a decision variable, which can take place within a known time interval. On the other hand, Lee and Leon (2001) consider single-machine scheduling with a rate-modifying activity. The rate-modifying activity is optional, which, if performed, changes the production rate of the machine. For more information, the reader may refer to the surveys on this subject by Schmidt (2000), Ma, Chu, and Zuo (2010) and Lee (2004).

It is noted that all of the above studies assume that the length of the maintenance activity is a constant regardless of the machine conditions. However, in real production, the length of the maintenance activity performed on a machine may depend on the state (e.g., running time) of the machine. For example, in the timber industry, log band mills are one of the main machines in a sawmill. Generally, the saw of a log band mill is maintained by an auto band saw sharpener to sustain the sharpness of its teeth. Specifically, the earlier a log band mill undergoes the teeth sharpening maintenance activity, the less blunt are its teeth, so the less time is needed to sharpen them. The motivation for this study stems from a sawmill that cuts various sizes and shapes of wood. Each log band mill is operated by a skilled worker. Typically, several log band mills are simultaneously available and a job (log) can be processed by any one of them. The conditions of the log band mills and workers are different. Therefore, the jobs have different processing times depending on the log band mills selected to process them.

Kubzin and Strusevich (2005) study a two-machine flow shop scheduling problem with no-wait in process to minimize the makespan with a maintenance period on one of the machines. They assume that the length of the maintenance activity depends on its starting time and provide a polynomial time approximation scheme for the problem. Kubzin and Strusevich (2006) consider the two-machine open shop and flow shop scheduling problems to minimize the makespan. They assume that each machine has to be maintained exactly once during the planning horizon and the length of each of maintenance activity depends on its starting time. They show that the open shop problem is polynomially solvable and the flow shop problem NP-hard, for which they present a fully polynomial approximation scheme and a fast 3/2-approximation algorithm. Mosheiov and Sidney (2010) study a single-machine scheduling problem with an option to perform a deteriorating maintenance activity. The objectives are to minimize the makespan, total completion time, maximum lateness, number of tardy jobs, and total earliness, tardiness, and due-date cost. They introduce polynomial time solutions for all these problems. Yang and Yang (2010) consider a single-machine scheduling problem

* Corresponding author. Fax: +886 5 6327291.
E-mail address: dlyang@nfu.edu.tw (D.-L. Yang).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.12.017

with a position-dependent aging effect described by a power function and maintenance activities that have variable maintenance durations. The objective is to find jointly the optimal maintenance frequency, the optimal maintenance positions, and the optimal job sequence to minimize the makespan. They show that the problem can be optimally solved in polynomial time.

We extend the model proposed by Mosheiov and Sidney (2010) to the unrelated parallel-machine setting. The objectives are to minimize the total completion time and the total machine load. We show that both versions of the problem can be optimally solved in polynomial time.

2. Notation and problem formulation

A set $J = \{J_1, J_2, \ldots, J_n\}$ of n jobs are to be processed on m unrelated parallel machines M_j, $j = 1, 2, \ldots, m$. Let n_j denote the number of jobs assigned to M_j and $P(n, m) = (n_1, n_2, \ldots, n_m)$ denote a job-allocation vector, where $\sum_{j=1}^{m} n_j = n$. We assume, as in most practical situations, that $m < n$. The jobs are non-preemptive and they are all available for processing at time zero. Each machine can handle at most one job at a time and cannot stand idle until the last job assigned to it has finished processing. Each machine requires at most one maintenance activity, which can be performed at any time throughout the planning horizon. We say that the sole maintenance activity of M_j (which is a rate-modifying activity), if it exists, is in position k_j ($0 \leqslant k_j \leqslant n_j$) if it is scheduled immediately after the completion time of the job scheduled in the k_jth position on M_j. Let a_{ij} (b_{ij}) denote the processing time of job J_i ($i = 1, 2, \ldots, n$) if it is scheduled before (after) the maintenance activity on M_j. Then the actual processing time of job J_i if it is scheduled in the rth position on M_j is $p_{ijr} = a_{ij}$ for $r \leqslant k_j$ and $p_{ijr} = b_{ij}$ for $r > k_j$, respectively, where $a_{ij} \geqslant b_{ij} > 0$ because the efficiency of job processing improves after machine maintenance.

We say that a maintenance activity, given it is scheduled immediately after the processing of the job scheduled in position k_j of M_j, is a deteriorating maintenance activity (DMA) of M_j if its length increases linearly with its starting time: $TMA_j = T_j + \delta_j t_j$, where $T_j > 0$, $\delta_j \geqslant 0$, and t_j is the starting time of the DMA of M_j. The objectives are to minimize the total completion time and the total machine load. For convenience, we use $\sum C_i$ to denote the total completion time and $\sum C^j_{\max}$ the total machine load, where C_i denotes the completion time of J_i and $C^j_{\max}$ denotes the makespan of M_j. Using the standard three-field notation for scheduling problems (Lee & Lin, 2001; Pinedo, 2008), we denote our scheduling problem as $Rm|p_{ijr}, TMA_j = T_j + \delta_j t_j|\gamma$, where γ, $\gamma \in \left\{\sum C_i, \sum C^j_{\max}\right\}$, is the objective function to be minimized.

3. Problems analysis

We first consider the special case of the above problem that involves a single machine, i.e., the $1|p_i = (a_i, b_i), TMA = T_0 + \delta t_k|\gamma$ problem. Assume that the DMA is scheduled immediately after the processing of the job scheduled in the kth position of the machine and the length of the DMA is $TMA = T_0 + \delta t_k$, where $T_0 > 0$, $\delta \geqslant 0$, and t_k is the starting time of the DMA. If the job sequence is $\pi = (J_1, J_2, \ldots, J_k, DMA, J_{k+1}, \ldots, J_n)$, then the total completion time and the makespan of π are, respectively,

$$\sum_{i=1}^{n} C_i = \left\{ \sum_{i=1}^{k} [(n-i+1) + \delta(n-k)]a_i + \sum_{i=k+1}^{n} (n-i+1) b_i \right\} + (n-k)T_0$$

and

$$C_{\max} = \left[\sum_{i=1}^{k} (1+\delta) a_i + \sum_{i=k+1}^{n} b_i \right] + T_0$$

3.1. Minimizing the total completion time

In this sub-section we consider the $Rm|p_{ijr}, nr, TMA_j = T_j + \delta_j t_j|\sum C_i$ problem. Before developing the results, we introduce a useful lemma and a proposition, which are used throughout the rest of this paper.

Lemma 1. *The number of nonnegative integer solutions to $l_1 + l_2 + \cdots + l_m = n$ is $C(m-1+n, n)$.*

Proof. See Mott, Kandel, and Baker (1986). □

Proposition 1. *The number of nonnegative integer solutions to $l_1 + l_2 + \cdots + l_m \leqslant n$ is $C(m+n, n) = \frac{(m+n)!}{n!(m)!}$ and is bounded from above by $\frac{(2n)^m}{m!}$.*

Proof. Let $l_{m+1} = n - (l_1 + l_2 + \cdots + l_m) \geqslant 0$, which implies that $l_1 + l_2 + \cdots + l_{m+1} = n$, the number of nonnegative integer solutions to which is $C(m-1+n, n)$ by Lemma 1. It is evident that $C(m+n, n) = \frac{(m+n)!}{n!m!} = \frac{(m+n)(m+n-1)\cdots(n+1)}{n!m!} \leqslant \frac{(2n)^m}{m!}$. This completes the proof of Proposition 1.

We say that the DMA of M_j is in position l_j ($0 \leqslant l_j \leqslant n$) if it is scheduled immediately before the job scheduled in position l_jth *to the last job* on M_j. That is, $l_j = n$ means that we perform the DMA before we process any job on M_j and $l_j = 0$ implies that we do not perform it on M_j in the current planning horizon. Let $(l_1, l_2, \ldots, l_m)$ denote a position-allocation vector of the DMAs. Define $y_{ijs} = 1$ if J_i is in position sth to the last job processed on M_j and $y_{ijs} = 0$ otherwise. Then we express the total completion time as

$$\sum C_i(l_1, l_2, \ldots, l_m) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[\sum_{s=1}^{l_j} s b_{ij} y_{ijs} + \sum_{s=l_j+1}^{n} (s + \delta_j l_j) a_{ij} y_{ijs} \right] + \sum_{j=1}^{m} l_j T_j \tag{1}$$

If the position (l_j) of the DMA of each machine is known in advance, then the last term on the RHS of (1) is a constant. To minimize (1) is equivalent to minimizing $\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\sum_{s=1}^{l_j} s b_{ij} y_{ijs} + \sum_{s=l_j+1}^{n}(s + \delta_j l_j) a_{ij} y_{ijs}\right]$. The problem can be formulated as the following $n \times nm$ Constrained Asymmetric Assignment Problem (CAAP), where we seek to assign the n jobs to nm positions on each machine, leaving a total of $nm - n$ positions unassigned:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[\sum_{s=1}^{l_j} s b_{ij} y_{ijs} + \sum_{s=l_j+1}^{n} (s + \delta_j l_j) a_{ij} y_{ijs} \right]$$

$$\text{subject to} \sum_{i=1}^{n} y_{i1s} = 1, \quad s = 1, 2, \ldots, l_1 \tag{2}$$

$$\sum_{i=1}^{n} y_{i1s} \leqslant 1, \quad s = l_1 + 1, \quad l_1 + 2, \ldots, n \tag{3}$$

$$\sum_{i=1}^{n} y_{i2s} = 1, \quad s = 1, 2, \ldots, l_2 \tag{4}$$

$$\sum_{i=1}^{n} y_{i2s} \leqslant 1, \quad s = l_2 + 1, \quad l_2 + 2, \ldots, n \tag{5}$$

$$\sum_{i=1}^{n} y_{ims} = 1, \quad s = 1, 2, \ldots, \quad l_m \tag{6}$$

$$\sum_{i=1}^{n} y_{ims} \leqslant 1, \quad s = l_m + 1, \quad l_m + 2, \ldots, n \tag{7}$$

$$\sum_{j=1}^{m} \sum_{s=1}^{n} y_{ijs} = 1, \quad i = 1, 2, \ldots, n \tag{8}$$

$$\sum_{i=1}^{n} y_{ij1} \geqslant \sum_{i=1}^{n} y_{ij2} \geqslant \cdots \geqslant \sum_{i=1}^{n} y_{ijn}, \ j = 1, 2, \ldots, m \tag{9}$$

$$y_{ijs} \in \{0, 1\}, \quad i = 1, 2, \ldots, n; \ j = 1, 2, \ldots, m; \quad s = 1, 2, \ldots, n \tag{10}$$