Computers & Industrial Engineering 87 (2015) 580–589

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Two-agent two-machine flowshop scheduling with learning effects to minimize the total completion time

Yau-Ren Shiau [a], Ming-Shua Tsai [b], Wen-Chiung Lee [b,*], T.C.E. Cheng [c]

[a] Department of Industrial Engineering and System Management, Feng Chia University, Taichung, Taiwan
[b] Department of Statistics, Feng Chia University, Taichung, Taiwan
[c] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 24 February 2014
Received in revised form 27 March 2015
Accepted 23 May 2015
Available online 17 June 2015

Keywords:
Scheduling
Total completion time
Two agent
Two-machine flowshop
Learning effects

ABSTRACT

We study a two-agent scheduling problem in a two-machine permutation flowshop with learning effects. The objective is to minimize the total completion time of the jobs from one agent, given that the maximum tardiness of the jobs from the other agent cannot exceed a bound. We provide a branch-and-bound algorithm for the problem. In addition, we present several genetic algorithms to obtain near-optimal solutions. Computational results indicate that the algorithms perform well in either solving the problem or efficiently generating near-optimal solutions.

1. Introduction

In classical scheduling problems the job processing times are always assumed to be fixed and known over the entire scheduling process. However, in many realistic situations, employees can process jobs more efficiently over time because they have accumulated experience in performing similar jobs repeatedly. This is known as the *learning effect* in the scheduling literature. Biskup (2008) provides a comprehensive review of research on scheduling with learning effects. For more recent studies on this stream of research, the reader may refer to Cheng, Lai, Wu, and Lee (2009), Cheng, Kuo, and Yang (2013), Lai and Lee (2011), Rudek (2012), Zhang, Yan, Huang, and Tang (2012), Zhang, Sun, and Wang (2013), Zhu, Sun, Chu, and Liu (2011), among others.

In many manufacturing and assembling processes, a job often consists of a number of operations that must be processed in a certain order, e.g., flowshop processing (Pinedo, 2008). However, scheduling with learning effects in the flowshop environment is relatively unexplored. Lee and Wu (2004) consider a two-machine flowshop problem to minimize the total completion time. They propose a branch-and-bound and a heuristic algorithm to derive the optimal and near-optimal solutions, respectively. Koulamas and Kyparisis (2007) introduce a sum-of-processing-time-based model in which the learning effect is expressed as the sum of the processing times of the jobs already processed. They solve the two-machine flowshop problem with ordered processing times under this learning model to minimize the makespan. Wu and Lee (2009) study the multi-machine permutation flowshop problem to minimize the total completion time. They provide a branch-and-bound algorithm that could solve instances with up to 16 jobs. Rudek (2011) studies the two-machine flowshop problem to minimize the makespan. He shows that an optimal solution is not necessarily a permutation schedule when the learning effect is taken into consideration. Furthermore, he proves that both the permutation and non-permutation versions of the problem are NP-hard even if the learning effect is a step learning curve. Li, Hsu, Wu, and Cheng (2011) study a two-machine flowshop scheduling problem where the learning effect is truncated. They develop a branch-and-bound solution method and three simulated annealing algorithms to solve the problem to minimize the total completion time. Kuo, Hsu, and Yang (2012) consider some flowshop problems with time-dependent learning effects. They provide some heuristic algorithms for the problem and analyze their worst-case performance.

On the other hand, the jobs that are to be scheduled might come from different customers, and their goals to pursue might not be the same. Baker and Smith (2003) give an example in which the goal of the manufacturing department is to finish jobs before their deadlines, whereas the goal of the research and development department is to get quick response. Kubzin and Strusevich (2006) point out that in maintenance planning, the maintenance

* Corresponding author. Tel.: +886 4 24517250x4016; fax: +886 4 24517092.
E-mail address: wclee@fcu.edu.tw (W.-C. Lee).

http://dx.doi.org/10.1016/j.cie.2015.05.032
0360-8352/

activities and the real jobs will compete for the machine occupancy. Baker and Smith (2003) and Agnetis, Mirchandani, Pacciarelli, and Pacifici (2004) initiate research on multi-agent scheduling. Their studies have inspired an abundance of subsequent research on this subject. For more studies on this line of research, the reader may refer to Agnetis, Pacciarelli, and Pacifici (2007), Cheng, Ng, and Yuan (2006), Cheng, Ng, and Yuan (2008), and Leung, Pinedo, and Wan (2010).

Recently, Lee, Chen, Chen, and Wu (2011) consider the two-agent scheduling two-machine flowshop problem to minimize the total completion time of the jobs of one agent, subject to no tardy job is allowed for the other agent. Liu, Yi, and Zhou (2011) solve some two-agent single-machine problems with increasing linear deterioration where the goal is to minimize the criterion of one agent, given that the criterion of the other agent cannot exceed a certain bound. Nong, Cheng, and Ng (2011) study a two-agent single-machine problem to minimize the sum of the maximum weighted completion time of the jobs of one agent and the total weighted completion time of the jobs of the other agent. They provide a 2-approximation algorithm and show that the case is NP-hard when the number of jobs of the first agent is fixed. Mor and Mosheiov (2010) consider single-machine batch scheduling to minimize the total completion time of the jobs of one agent, subject to an upper bound on the maximum completion time of the jobs of the other agent. Wu, Huang, and Lee (2011) study two-agent single-machine scheduling with learning effects. The objective is to find a schedule that minimizes the total tardiness of the jobs of one agent, subject to no tardy job for the other agent. In two-agent scheduling research, most studies focus on the single-machine setting with learning effects or the two-machine flowshop setting without learning effects. To the best of our knowledge, our study is the first to consider two-agent permutation flowshop scheduling with learning effects.

We consider two-agent scheduling with learning effects in the two-machine flowshop environment. The objective is to find a schedule that minimizes the total completion time of the jobs of one agent, subject to the maximum tardiness of the jobs of the second agent cannot exceed a given bound. The rest of the paper is organized as follows: In the next section we introduce and formulate the problem. In Section 3 we provide a branch-and-bound algorithm to optimally solve the problem. In Section 4 we propose several genetic algorithms. In Section 5 we present the results of extensive computational experiments conducted to evaluate the performance of the proposed algorithms. We conclude the paper and suggest topics for future research in the final section.

2. Problem description

There are n jobs available at time zero that need to be processed on two machines in the same order, i.e., each job must be processed first on machine 1 and then on machine 2. Jobs come from either agent 1 (AG_1) or agent 2 (AG_2). For job j, there is a normal processing time a_j on machine 1, a normal processing time b_j on machine 2, a due date d_j, and an agent code I_j, where $I_j = 1$ if $j \in AG_1$ or $I_j = 2$ if $j \in AG_2$. Following the learning effect model proposed by Lee (2011), we can compute the actual processing time of job j on machines 1 and 2, if it is scheduled in the rth position of a sequence (or schedule), as follows:

$$a_{j[r]} = a_j L_r = a_j \prod_{k=0}^{r-1} l_k, \tag{1}$$

and

$$b_{j[r]} = b_j L_r = b_j \prod_{k=0}^{r-1} l_k, \tag{2}$$

where $l_0 = 1$ and $0 < l_k \leqslant 1$ for $k = 1, \ldots, n$. Under a schedule S, let the completion times of job j on machines 1 and 2 be $C_{1j}(S) = C_{1[r-1]}(S) + a_{j[r]}$ and $C_{2j}(S) = \max\{C_{2[r-1]}(S), C_{1j}(S)\} + b_{j[r]}$, respectively, if it is scheduled in the rth position, where $C_{1[r-1]}(S)$ and $C_{2[r-1]}(S)$ denote the completion times of the jobs in the $(r-1)$th position on machines 1 and 2, respectively, and $T_j(S) = \max\{0, C_{2j}(S) - d_j\}$ be the tardiness of job j. We consider the two-agent two-machine flowshop scheduling problem with learning effects to minimize the total job completion time of agent AG_1, given that the maximum job tardiness of agent AG_2 cannot exceed a bound M. Using the three-field notation introduced in Agnetis et al. (2004), we denote the problem as $F2|LE;LE|\sum C_j; T_{\max}$.

3. A branch-and-bound algorithm

When the jobs are all from agent AG_1 with no learning effect, problem $F2|LE;LE|\sum C_j; T_{\max}$ is the classical two-machine flowshop problem to minimize the total completion time, which is known to be NP-hard (Gonzalez & Sahni, 1978). So problem $F2|LE;LE|\sum C_j; T_{\max}$ is NP-hard, too, and we resort to using the branch-and-bound method to find the optimal solution for the problem.

3.1. Dominance properties

In this subsection we provide several dominance properties for the total completion time criterion to reduce the search space in the branch-and-bound algorithm. Suppose that S and S' are two schedules of jobs, and the difference between them is a pairwise interchange of two adjacent jobs i and j, i.e., $S = (\pi, i, j, \pi')$ and $S' = (\pi, j, i, \pi')$, where π and π' each denote a partial sequence. Furthermore, suppose there are $r-1$ jobs in π, and t_1 and t_2 are the completion times of the last job in π on machines 1 and 2, respectively. Since jobs i and j might come from agent AG_1 or AG_2, we classify the situations into the following cases.

Case 1: Both jobs belong to agent AG_1.

To show that S dominates S', it suffices to show that $C_{1j}(S) \leqslant C_{1i}(S')$, $C_{2j}(S) \leqslant C_{2i}(S')$, and $C_{2i}(S) + C_{2j}(S) < C_{2i}(S') + C_{2j}(S')$.

Property 1. If $a_i \leqslant a_j$, $b_j > b_i$, $t_1 + a_i L_r \geqslant t_2$, and $a_j l_r \leqslant b_i$, then S dominates S'.

Proof. Since $t_1 + a_i L_r \geqslant t_2$ and $a_j l_r \leqslant b_i$, the completion times of jobs i and j in S on machines 1 and 2 are

$$C_{1i}(S) = t_1 + a_i L_r$$
$$C_{2i}(S) = \max\{t_1 + a_i L_r, t_2\} + b_i L_r = t_1 + a_i L_r + b_i L_r$$
$$C_{1j}(S) = t_1 + a_i L_r + a_j L_{r+1}$$

and

$$C_{2j}(S) = \max\{C_{1j}(S), C_{2i}(S)\} + b_j L_{r+1} = t_1 + a_i L_r + b_i L_r + b_j L_{r+1}.$$

Since $a_i \leqslant a_j$ and $t_1 + a_i L_r \geqslant t_2$, the completion times of jobs i and j in S' on machines 1 and 2 are

$$C_{1j}(S') = t_1 + a_j L_r$$
$$C_{2j}(S') = \max\{t_1 + a_j L_r, t_2\} + b_j L_r = t_1 + a_j L_r + b_j L_r.$$
$$C_{1i}(S') = t_1 + a_j L_r + a_i L_{r+1}$$