Int. J. Production Economics 141 (2013) 327–334

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

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

Permutation flowshop scheduling to minimize the total tardiness with learning effects

Wen-Chiung Lee [a], Yu-Hsiang Chung [b,*]

[a] Department of Statistics, Feng Chia University, Taichung, Taiwan
[b] Department of Industrial & Engineering Management, National Chiao Tung University, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 4 January 2011
Accepted 1 August 2012
Available online 25 August 2012

Keywords:
Scheduling
Permutation flowshop
Total tardiness
Learning effect

ABSTRACT

Scheduling with learning effects has received considerable attention recently. Often, numbers of operations have to be done on every job in many manufacturing and assembly facilities. However, it is seldom discussed in the general multiple-machine setting, especially without the assumptions of identical processing time on all the machines or dominant machines. With the current emphasis of customer service and meeting the promised delivery dates, we consider a permutation flowshop scheduling problem with learning effects where the objective is to minimize the total tardiness. A branch-and-bound algorithm and two heuristic algorithms are established to search for the optimal and near-optimal solutions. Computational experiments are also given to evaluate the performance of the algorithms.

1. Introduction

In classical scheduling, the job processing times are assumed to be fixed and known throughout the entire process. However, this assumption might not reflect many real-life situations. For example, Biskup (1999) pointed out that repeated processing of similar tasks improves the worker skills; workers are able to perform setup, to deal with machine operations or software, or to handle raw materials and components at a greater pace. Biskup (1999) and Cheng and Wang (2000) were among the pioneers that brought the concept of learning effects into the scheduling field. Many researchers have devoted to this young but vivid area since. Biskup (2008) provided a comprehensive review of the scheduling models and problems with learning effects.

Recently, Wang (2007) considered some single-machine problems with the effects of learning and deterioration, and proved that the makespan and the sum of completion time problems remain polynomially solvable. He also showed that the weighted shortest processing time rule and the earliest due date rule provide the optimal schedules for the weighted sum of completion time and the maximum lateness problems in some special cases. Janiak and Rudek (2008) considered a scheduling problem in which each job provides a different experience to the processor. They relaxed one of the rigorous constraints, and thus each job can provide different experience to the processor in their model. They then formulated the job processing time as a non-increasing k-stepwise function that in general is not restricted to a certain learning curve, thereby it can accurately fit every possible shape of a learning function. Lee and Wu (2004) investigated a two machine flowshop scheduling problem with learning consideration to minimize the total completion time. They utilized the branch-and-bound algorithm incorporated with several dominance properties and lower bounds to obtain the optimal solution. An accurate heuristic algorithm was also proposed to obtain the near-optimal solution. Cheng et al. (in press) studied a two-machine flowshop scheduling problem with a truncated learning function to minimize the makespan. They utilized a branch-and-bound and three heuristic algorithms to derive the optimal and near-optimal solutions. Wang (2008) studied some single-machine problems with the sum-of-processing-time-based learning effect. He showed by examples that the classical optimal rules no longer provide the optimal solutions under the proposed model. He also provided the optimal solutions for some single-machine problems under certain conditions. Cheng et al. (2008), Lee and Wu (2009), Yin et al. (2009) and Zhang and Yan (2010) considered a variety of models in which the actual job processing time not only depends on its scheduled position, but also depends on the sum of the processing times of jobs already processed. They provided the optimal schedules for some single machine problems. Janiak and Rudek (2010) presented a new approach called multi-abilities learning that generalizes the existing ones and models. On this basis, they focused on the makespan problem and provided the optimal polynomial time algorithms for some special cases. Lee et al. (2010) investigated a single-machine problem with the learning effect and release times where the objective is to minimize the

* Corresponding author. Tel.: +886 3 928 395863; fax: +886 3 572 9101.
E-mail address: yhchung.iem96g@nctu.edu.tw (Y.-H. Chung).

0925-5273/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2012.08.014

makespan. Chang et al. (2009) studied a single machine scheduling problem, in which the learning/aging effect is considered. The objective is to determine the common due date and the sequence of jobs that minimizes a cost function.

Often numbers of operations have to be done on every job in many manufacturing and assembly facilities (Pinedo, 2002; Tseng and Lin, 2010; Wang et al., 2010; Zhao and Tang, 2012; Shabtay et al., 2012; Sun et al., 2012). However, it is seldom discussed in the general multiple-machine setting, especially without the assumptions of identical processing time on all the machines or dominant machines. Wang and Xia (2005) studied some permutation flowshop problems when the learning effect is present. They provided the worst-case bound of the shortest processing time rule for the makespan and the total completion time problems. They also showed that the makespan and the total completion time problems remain polynomially solvable for two special cases. Xu et al. (2008) provided heuristic algorithms for some permutation flowshop problems. They also analyzed the worst case bounds for the proposed algorithms. Wu and Lee (2009) considered a permutation flowshop scheduling problem to minimize the total completion time. They also analyzed the performance of the existing heuristic algorithms when the learning effect is present.

With the current emphasis of customer service and meeting the promised delivery dates, in this paper we consider a permutation flowshop scheduling problem to minimize the total tardiness with the learning effect. Although the classical problem without the consideration of learning effects has attracted the attention of numerous researchers due to its simple definition, most of the research focused on developing the heuristic or meta-heuristic algorithms due to the complexity of the problem. Recently, Vallada et al. (2008) provided a comprehensive review of the heuristic algorithms. To the best of our knowledge, Kim (1995) and Chung et al. (2006) were the only authors who derived the optimal schedules. In this paper, we provide a branch-and-bound and two heuristic algorithms when the learning effect is present. The rest of the paper is organized as follows. In the next section we describe the formulation of our problem. In Section 3, we construct a branch-and-bound algorithm using an elimination rule and a lower bound to speed up the search for the optimal solution. In Section 4, two heuristic algorithms are proposed to solve this problem. In Section 5, a computational experiment is conducted to evaluate the efficiency of the branch-and-bound algorithm and the performance of the heuristic algorithms. A conclusion is given in the last section.

2. Problem description

There are n jobs and m machines. For each job j, there are associated with m operations $O_{1j}, O_{2j}, \ldots, O_{mj}$ where operation O_{ij} must be processed on machine i, $i=1,2,\ldots,m$. Processing of operation $O_{i+1,\ j}$ can start only after operation O_{ij} is completed. Moreover, we focus on the permutation flowshop case which implies the job sequence is the same in all the machines. The normal processing time of operation O_{ij} is denoted by p_{ij} and the due date of job j is d_j. The actual processing time p_{ijr} of operation O_{ij} is a function of its position in a schedule. That is,

$$p_{ijr} = p_{ij} r^a, \quad i=1,2,\ldots,m; \quad r=1,2,\ldots,n,$$

if it is scheduled in the rth position and $a<0$ is the learning effect.

For a given schedule S, let $C_{ij}(S)$ denote the completion time of job j on machine i, $T_j(S)=\max{0,C_{mj}(S)-d_j}$ denote the tardiness of job j, and $C_{i[j]}(S)$ denote the completion time of the job scheduled in the jth position on machine i. The objective of this paper is to find a schedule that minimizes the total tardiness, a widely used performance measure in scheduling literature. That is, we want to find a schedule S^* such that $\sum T_j(S^*) \le \sum T_j(S)$ for any schedule S.

3. A branch-and-bound algorithm

The problem under study is NP-hard since it already is even without the learning effect (Pinedo, 2002). Thus, the branch-and-bound algorithm might be a good way to obtain the optimal solution. In this section, we first provide a dominance property, followed by the lower bound to speed up the search process, and finally the branch-and-bound algorithm.

3.1. Dominance property

Chung et al. (2006) gave a dominance property for the classical problem. In this subsection, we modified the property to take the learning effect into consideration. Before presenting the property, we first state a lemma from Chung et al. (2006).

Lemma 1. $\max{0,a-b} \ge \max{0,a-c} - \max{0,b-c}$ for arbitrary real numbers a, b, and c.

Property 1. Suppose that $S_1=(\sigma_1,\pi)$ and $S_2=(\sigma_2,\pi)$ are two sequences where σ_1 and σ_2 are partial sequences which contains the same set of s jobs. If

$$\sum_{j=1}^{s} T_{[j]}(\sigma_2) - \sum_{j=1}^{s} T_{[j]}(\sigma_1) \ge (n-s)\max{0, \max_{1\le i\le m} C_{i[s]}(\sigma_1) - C_{i[s]}(\sigma_2)},$$

then S_1 dominates S_2.

Proof. By definition, the completion time of the nth job of S_1 on machine m is

$$C_{m[n]}(S_1) = \max_{1\le i\le m} C_{i[n-1]}(S_1) + \sum_{l=i}^{m} p_{l[n]} n^a = C_{i_1[n-1]}(S_1) + \sum_{l=i_1}^{m} p_{l[n]} n^a$$

for some i_1 where $1 \le i_1 \le m$. Similarly, the completion time of the nth job of S_2 on machine m is

$$C_{m[n]}(S_2) = \max_{1\le i\le m} C_{i[n-1]}(S_2) + \sum_{l=i}^{m} p_{l[n]} n^a = C_{i_2[n-1]}(S_2) + \sum_{l=i_2}^{m} p_{l[n]} n^a$$

for some i_2 where $1 \le i_2 \le m$. Thus, we have

$$\begin{aligned} C_{m[n]}(S_1) - C_{m[n]}(S_2) &= [C_{i_1[n-1]}(S_1) + \sum_{l=i_1}^{m} p_{l[n]} n^a] - [C_{i_2[n-1]}(S_2) + \sum_{l=i_2}^{m} p_{l[n]} n^a] \\ &\le [C_{i_1[n-1]}(S_1) + \sum_{l=i_1}^{m} p_{l[n]} n^a] - [C_{i_1[n-1]}(S_2) + \sum_{l=i_1}^{m} p_{l[n]} n^a] \\ &\le C_{i_1[n-1]}(S_1) - C_{i_1[n-1]}(S_2) \le \max_{1\le i\le m} C_{i[n-1]}(S_1) - C_{i[n-1]}(S_2) \end{aligned}$$

By an induction argument, we have

$$C_{m[j]}(S_1) - C_{m[j]}(S_2) \le \max_{i\le i\le m} C_{i[s]}(S_1) - C_{i[s]}(S_2) \text{ for } j=s+1,\ldots,n \tag{1}$$

From Lemma 1 and Eq. (1), the difference between the total tardiness of S_1 and S_2 is

$$\begin{aligned} \sum_{j=1}^{n} T_{[j]}(S_2) - \sum_{j=1}^{n} T_{[j]}(S_1) &= \sum_{j=1}^{s} T_{[j]}(S_2) - \sum_{j=1}^{s} T_{[j]}(S_1) - [\sum_{j=s+1}^{n} T_{[j]}(S_1) - \sum_{j=s+1}^{n} T_{[j]}(S_2)] \\ &= \sum_{j=1}^{s} T_{[j]}(S_2) - \sum_{j=1}^{s} T_{[j]}(S_1) - \sum_{j=s+1}^{n} (\max{0,C_{m[j]}(S_1)-d_{[j]}} \\ &\quad -\max{0,C_{m[j]}(S_2)-d_{[j]}}) \ge \sum_{j=1}^{s} T_{[j]}(S_2) - \sum_{j=1}^{s} T_{[j]}(S_1) \\ &\quad -(n-s)\max{0, \max_{1\le i\le m} C_{i[s]}(S_1) - C_{i[s]}(S_2)} \end{aligned}$$

It implies that S_1 dominates S_2 and this completes the proof.

Property 1. can be simplified to the case of two adjacent jobs which is stated without proof.