Information Sciences 256 (2014) 91–108

Contents lists available at ScienceDirect

Information Sciences

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

A branch-and-bound algorithm for a single machine sequencing to minimize the total tardiness with arbitrary release dates and position-dependent learning effects

Yunqiang Yin [a,b], Wen-Hung Wu [c], Wen-Hsiang Wu [d,*], Chin-Chia Wu [e]

[a] Fundamental Science on Radioactive Geology and Exploration Technology Laboratory, East China Institute of Echnology, Nanchang 330013, China
[b] School of Sciences, East China Institute of Technology, Fuzhou, Jiangxi 344000, China
[c] Department of Business Administration, Kang-Ning Junior College, Taipei, Taiwan
[d] Department of Healthcare Management, Yuanpei University, Hsinchu, Taiwan
[e] Department of Statistics, Feng Chia University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 27 August 2011
Received in revised form 19 April 2013
Accepted 28 July 2013
Available online 6 August 2013

Keywords:
Scheduling
Single-machine
Tardiness
Learning effect

ABSTRACT

This study considers an NP-hard problem of minimizing the total tardiness on a single machine with arbitrary release dates and position-dependent learning effects. A mixed-integer programming (MIP) model is first formulated to find the optimal solution for small-size problem instances. Some new dominance rules are then presented which provide a sufficient condition for finding local optimality. The branch-and-bound (B& B) strategy incorporating with several dominance properties and a lower bound is proposed to derive the optimal solution for medium- to-large-size problem instances, and four marriage-in-honey-bees optimization algorithms (MBO) are developed to derive near-optimal solutions for the problem. To show the effectiveness of the proposed algorithms, 3600 situations with 20 and 25 jobs, are randomly generated for experiments.

1. Introduction

Some simplest and most studied scheduling problems involve due date-based objectives on a single machine. These problems deal with scheduling multiple tasks that compete for service on a single resource, or with a machine to meet some objective concerning due dates of the tasks. A frequently encountered due date-based objective is to minimize the total tardiness of all the tasks where total tardiness scheduling models have received considerable and increasing attention from the scheduling community, due to their practical importance and relevance. It has been shown by Rinnooy Kan [35] that the total tardiness problem problem with release times on a single machine is NP-hard in the strong sense. Accordingly, most research has directed to aiming at providing a near-optimal solution in a short time. For the total tardiness objective, a lot of research efforts have been conducted on the problem with equal release dates $1||\sum T_j$ where powerful dominance rules were introduced by Emmons [18]. Most methods for solving $1||\sum T_j$ strongly rely on Emmons' dominance rules. For example, Potts and Van Wassenhove [34], Chang et al. [9] and Szwarc et al. [38], developed branch-and-bound methods using the Emmons rules coupled with the decomposition rule of Lawler [30] together with some other elimination rules. There are few results on problems with arbitrary release dates. Chu and Portmann [13] introduced a sufficient condition for finding local optimality which allowed them to build a dominant subset of schedules. Chu [14] also proposed a branch-and-bound method using some efficient dominance rules. Baptiste et al. [5] presented a branch-and-bound to minimize the total tardiness with

* Corresponding author. Tel.: +886 36102321; fax: +886 36102323.
E-mail address: wenhsiang_wu@yahoo.com.tw (W.-H. Wu).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2013.07.026

arbitrary release dates and introduced new lower bound and generalized some well-known dominance properties. Su and Chen [39] addressed a problem of minimizing total tardiness on a single machine with unequal release dates. They developed a branch and bound and heuristic procedures to solve the problem. The computational experimental results revealed that the proposed branch and bound algorithm was efficient in solving hard problems as well as easy problems. For recent studies in total tardiness problems, the reader may refer to a recent review paper by Koulamas [25].

On the other hand, in classical scheduling problems, the processing times of jobs are assumed to be constant values. Recent empirical studies in several industries have demonstrated that unit costs decline as companies produce more products and gain knowledge or experience. This phenomenon is well-known and as referred to "learning effect" in the literature. The learning effect on scheduling problem was first introduced by Biskup [7]. Since then, scheduling problem with learning effect have received increasing attention in the scheduling research community. For research on other scheduling models considering learning effects and under different machine environments, the readers can refer to Bachman and Janiak [4], Cheng et al. [10–12], Janiak and Rudek [19–21], Koulamas et al. [26], Kuo and Yang [27,28], Lai and Lee [29], Lee [31], Rudek [37], Wang and Guo [42], Wang et al. [43,44], Wang and Wang [45,46], Wu and Lee [47], Yang and Yang [50,49], Yin et al. [51,52], and the survey paper by Biskup [8].

Over the years quite a few research have been conducted on scheduling problems regarding learning effects with release times. Bachman and Janiak [4] considered the learning effect scheduling problem with release times where they proved the makespan problem is NP-hard in the strong sense for two different learning models. Wu and Liu [48] proposed a branch-and-bound algorithm and three two-stage heuristic algorithms for minimizing the makespan on a single machine with learning and unequal release times. Lee et al. [32] further provided exact and heuristic algorithms to solve the same problem. Toksarı [41] investigated a single-machine scheduling problem with unequal release times under learning effect and deteriorating jobs where the objective is to minimize the makespan and presented a branch-and-bound algorithm to derive the optimal solution.

With modeling a realistic production system in mind, this paper addresses a single-machine total tardiness scheduling problem with arbitrary release dates and position-dependent learning effects simultaneously. The purpose of this study is twofold. One is that such a problem has not been explored. Another is to provide procedures to solve the problem. To accomplish these two tasks,

(1) for small-size problem instances, we present an MIP model to find the optimal solution,
(2) for medium-to-large-size problem instances, we provide a branch-and-bound algorithm incorporating with several dominance properties and a lower bound to derive the optimal solution and also develop an MBO algorithm to derive the near-optimal solutions.

The remainder of this paper is organized as follows. In Section 2, the problem is formulated. In Section 3, a mixed integer programming model is presented to solve the problem. In Section 4, some dominance properties and two lower bounds are developed to enhance the search efficiency for the optimal solution, followed by descriptions of a marriage-in- honey-bees optimization algorithm and the branch-and-bound algorithm. In Section 5, a computational experiment is conducted to evaluate the performance of a branch-and-bound algorithm and the proposed the marriage in honey-bees optimization algorithm. A conclusion is provided in the last section.

2. Model formulation

In this section, the notations that are used throughout the paper will be introduced first, followed by the formulation of the problem.

n	the number of jobs
S	sequence of jobs
p_j	the normal processing time of job J_j
r_j	the release date of job J_j
d_j	the due date of job J_j
$p_{[k]}(S)$	the normal processing time of the job assigned in the kth position in S
$r_{[k]}(S)$	the release date of the job assigned in the kth position in S
$d_{[k]}(S)$	the due date of the job assigned in the kth position in S
$C_j(S)$	the complete time of job J_j in S
$C_{[k]}(S)$	the complete time of the job assigned in the kth position in S
$T_j(S) = \{C_j(S) - d_j, 0\}$	the tardiness of job J_j in S
$T_{[k]}(S) = \{C_{[k]}(S) - d_{[k]}, 0\}$	the tardiness of the job assigned in the kth position in S