Information Sciences 340–341 (2016) 305–320

Contents lists available at ScienceDirect

Information Sciences

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

A lower bound for minimizing the total completion time of a three-agent scheduling problem

Yau-Ren Shiau [a], Wen-Chiung Lee [b], Yu-Sheng Kung [b], Jen-Ya Wang [c,*]

[a] Department of Industrial Engineering and System Management, Feng Chia University, Taichung, Taiwan, ROC
[b] Department of Statistics, Feng Chia University, Taichung, Taiwan, ROC
[c] Department of Computer Science and Information Management, Hungkuang University, Taichung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 15 August 2015
Revised 13 October 2015
Accepted 10 January 2016
Available online 15 January 2016

Keywords:
Discrete optimization
Job scheduling
Multiple agents
Maintenance activity
Branch-and-bound algorithm

ABSTRACT

In the field of job scheduling, multi-agent issues have been studied for many years. Most of researchers focused their attention only on two agents. However, there are more than two agents in the real-world scheduling problems. In this study, we consider a single-machine multi-agent scheduling problem with release time and maintenance activity. The objective is to minimize the first agent's total completion time given that the tardiness of jobs from the second agent does not exceed a limit and the maintenance activity from the third agent must be conducted within a specified time interval, i.e., maintenance window. A lower bound is proposed to accelerate the branch-and-bound algorithm. Computational experiments show the proposed lower bound performs well. The improvement ratio even reaches 1789%.

1. Introduction

In the real world, there might be different kinds of customers. Some customers cannot endure tardiness and would pay more for punctual deliveries; some customers care about cost more than time. Therefore, a good schedule would meet different requests from multiple kinds of customers at the same time. On the other hand, orders are not placed on the same date. So jobs are tagged with different release times. Once a release time is up, we can start processing the corresponding job. Moreover, machines need maintenance in order not to malfunction or not to disappoint customers. Job scheduling with completion time, release time, maintenance is thus called for. For more research into scheduling with completion time, release time, maintenance, please refer to [3,4,7,12,15].

In the last decade, researcher paid much attention to multi-agent scheduling. Baker and Smith [1] first explored a two-agent scheduling problem and discussed three criteria. They showed an example of model manufacturing and indicated that each department has its own goal. Cheng et al. [6] considered two-agent scheduling on a single machine, where each agent's objective is to minimize the total weighted number of tardy jobs. Lee et al. [10] considered a single-machine scheduling problem with a linear deterioration assumption where the objective is to minimize the total weighted completion time for the first agent with the restriction that no tardy job is allowed for the second agent. Pratap et al. [13] introduced a berth allocation problem which aims to minimize ship waiting time and deviation of customer priority at the same time. It is clear that multi-agent scheduling is an important issue. However, only two agents are considered in all the above studies.

* Corresponding author. Tel.: +886 426318652; fax: +886 426521921.
E-mail address: jywang@sunrise.hk.edu.tw, wang7160@gmail.com (J.-Y. Wang).

http://dx.doi.org/10.1016/j.ins.2016.01.031
0020-0255/

Table 2.1
The symbols used in this study.

Symbol	Description
n	Number of jobs
S	Schedule
α	Determined partial sequence of S
β	Undetermined partial sequence of S
AG^i	Set of jobs from agent i, where $i = 1, 2, 3$
M	Maximum acceptable tardiness for jobs in AG^2
B	Left endpoint of maintenance window
E	Right endpoint of maintenance window
r_j	Release time of job j
p_j	Processing time of job j
d_j	Due date of job j
$C_j^i(S)$	Completion time of job j from AG^i
$T_j(S)$	Tardiness of job j from AG^2
$C_j(S)$	Completion time of job j

A number of studies have recently discussed multi-agent scheduling with release time. Lee et al. [9] considered a two-agent scheduling problem on a single machine with release time. The first agent aims to minimize the total tardiness and the second agent manages to control its maximum tardiness lower than a specified limit. Yin et al. [17] considered another two-agent scheduling problem with release time. Subject to the maximum tardiness enforced on the second agent, the first agent aims to minimize its total tardiness. Cheng et al. [4] considered two-agent scheduling with release time on a single machine. The first agent aims to minimize the total weighted completion time and the constraint is a maximum tardiness for this second agent. Wu et al. [16] also considered the issues of multiple agents and release time on a single machine. The objective is to minimize the total completion time of the first agent, where the total completion time of the second agent cannot exceed a specified limit.

In the field of job scheduling, regular maintenance is also an important issue. Ji et al. [8] considered multiple maintenance activities needed to be scheduled on a single machine. Meanwhile, they minimized the total completion time. Low et al. [11] considered a specified time interval (maintenance window) when they minimized the total completion time. With the consideration, a small maintenance activity can be scheduled more flexibly within this window. Cheng et al. [5] considered a parallel-machine scheduling problem. In this study, the maintenance activities may deteriorate. Sun et al. [14] considered two kinds of maintenance, i.e., minor maintenance, overhaul maintenance. They needed to determine the minor maintenance rate and the overhaul maintenance times in order to minimize the total maintenance cost. Yin et al. [18] proposed a mixed Integer programming model to schedule ordinary jobs and a maintenance activity in order to minimize the total tardiness. All the above research shows maintenance is an important issue when we schedule jobs no matter what objective we have.

To our best knowledge, few researchers consider a scheduling problem with three agents or more at the same time. Therefore, we consider a three-agent problem with completion time, release time, and maintenance activity simultaneously. The objective of the problem is to minimize the first agent's total completion time given that the tardiness suffered by the second agent does not exceed a specified limit and the maintenance activity performed by the third agent must be conducted within a specified time interval. With these considerations, the proposed scheduling problem becomes more realistic and applicable than before.

2. Problem description

The three-agent scheduling problem is described as follows. There are n jobs and they are from three agents. The set of jobs from agent i is denoted by AG^i and AG^3 contains only one maintenance activity (i.e., job n) which must be conducted within a maintenance window $[B,E]$. Each job in AG^1 has a release time r_j and each job in AG^2 has a due date d_j. Let p_j denote the processing time of job j for $j = 1, 2, \ldots, n$. Let $C_j^i(S)$ denote the completion time of job j from AG^i and $T_j(S) = \max\{0, C_j^2(S) - d_j\}$ denote the tardiness of job j from AG^2. Note that no tardiness can be greater than a given limit M, i.e., $T_{\max}(S) = \max_{j \in AG^2}\{T_j(S)\} \le M$. Under the above assumption, the problem is to minimize the total completion time of jobs in AG^1, i.e., $1|T_{\max} \le M, B + p_n \le C_j^3(\pi) \le E| \min \sum C_j^1(\pi)$. The related symbols are summarized in Table 2.1.

Table 2.2 shows a problem instance and Fig. 2.1 illustrates the optimal schedule. Note that job 8 means the maintenance activity and it needs to be performed within [38,49]. There is an optimal schedule $S = (5, 6, 8, 2, 3, 4, 1, 7)$. For this optimal schedule, each tardiness is less than 10 and the total completion time is 239 (= 48 + 55 + 63 + 73).

3. Branch-and-bound algorithm

A branch-and-bound (B&B) algorithm is proposed to accelerate the execution speed. We first develop 24 dominance properties (Section 3.1) and two lower bounds (Section 3.2). Then, we present the pseudo code of B&B (Section 3.3). For the problem instances with $n \le 30$, B&B aims to provide the optimal solutions within 10 minutes.