Computers & Operations Research 40 (2013) 353–361

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Two-agent singe-machine scheduling with release times to minimize the total weighted completion time

T.C.E. Cheng [a], Yu-Hsiang Chung [b], Shan-Ci Liao [c], Wen-Chiung Lee [c,*]

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

ARTICLE INFO

Available online 23 July 2012

Keywords:
Scheduling
Total weighted completion time
Maximum lateness
Two agents

ABSTRACT

In many management situations multiple agents pursuing different objectives compete on the usage of common processing resources. In this paper we study a two-agent single-machine scheduling problem with release times where the objective is to minimize the total weighted completion time of the jobs of one agent with the constraint that the maximum lateness of the jobs of the other agent does not exceed a given limit. We propose a branch-and-bound algorithm to solve the problem, and a primary and a secondary simulated annealing algorithm to find near-optimal solutions. We conduct computational experiments to test the effectiveness of the algorithms. The computational results show that the branch-and-bound algorithm can solve most of the problem instances with up to 24 jobs in a reasonable amount of time and the primary simulated annealing algorithm performs well with an average percentage error of less than 0.5% for all the tested cases.

1. Introduction

In traditional scheduling models, there is a single criterion for all the jobs. However, in the real world, customers are distinct in that they pursue different objectives. For instance, Baker and Smith [1] give an example of a prototype shop being shared by a research and development (R&D) department, which tests new designs, and by a manufacturing engineering department, which runs experiments to improve the robustness of production processes. In this context, the R&D customers might be concerned about meeting due-dates, while the engineering customers might be more concerned about quick response time. Kim et al. [2] point out that in project scheduling the major concern is with negotiation to resolve conflicts whenever the agents find their own schedules unacceptable. Schultz et al. [3] remark that in telecommunications services, the problem is to satisfy the service requirements of individual agents, who compete for the use of a commercial satellite to transfer voice, image, and text files for their clients. Agnetis et al. [4] present examples of scheduling involving multiple agents competing on the usage of common processing resources in different application environments and methodological fields, such as artificial intelligence, decision theory, and operations research.

Agnetis et al. [4] and Baker and Smith [1] first introduced the multi-agent problems into the scheduling field. They considered scheduling models with two agents and their objective functions include the total weighted completion time, the number of tardy jobs, and the maximum of regular non-decreasing functions of the job completion times. Later, Yuan et al. [5] revised the dynamic programming recursion formulae in Baker and Smith [1] and derived a polynomial-time algorithm for the same problem. Ng et al. [6] considered a single-machine two-agent problem where the objective is to minimize the total completion time of one agent, given that the number of tardy jobs of the other agent cannot exceed a certain number. They showed that the problem is NP-hard under high multiplicity encoding and can be solved in pseudo-polynomial time under binary encoding. Cheng et al. [7] studied the feasibility model involving more than two agents on a single machine where each agent's objective function is to minimize the total weighted number of tardy jobs. They showed that the general problem is strongly NP-complete. Agnetis et al. [8] studied the scheduling problems where several agents compete to perform their respective jobs on one sharing processing resource and the cost function depends on the job completion times only. They analyzed the complexity of various problems arising from different combinations of the cost functions of each agent. In particular, they investigated the problem of finding schedules whose cost for each agent does not exceed a given bound. Cheng et al. [9] examined scheduling problems on a single machine involving more than two agents where the objective functions are of the max-form. In addition, Liu and Tang [10] discussed two two-agent scheduling problems with the consideration of a simple linear deterioration effect on a single machine. Their objective functions are the maximum lateness and the total completion time of the jobs of one agent, given a bound on the maximum of a non-decreasing function of the job completion times of the other agent. Moreover, Lee

* Corresponding author.
E-mail address: wclee@fcu.edu.tw (W.-C. Lee).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.07.013

et al. [11] discussed a scheduling problem on a single machine involving more than two agents in which each agent is responsible for his own set of jobs and wishes to minimize the total weighted completion time of his own set of jobs. They reduced this NP-hard problem to a multi-objective short-path problem. They also provided an efficient approximation algorithm with a reasonably good worst-case ratio. Agnetis et al. [12] developed branch-and-bound algorithms for several single-machine scheduling problems with two competing agents. They used Lagrangian duals to derive bounds for the branch-and-bound algorithm in strongly polynomial time. Lee et al. [13] studied a two-agent scheduling problem in a two-machine permutation flowshop where the objective is to minimize the total tardiness of the jobs of one agent, given that the number of late jobs of the other agent is zero. Recently, Leung et al. [14] generalized the two-agent single-machine problems proposed by Agnetis et al. [4] to the case of multiple identical parallel machines. In addition, they also considered the situations where the jobs may have different release dates and preemptions may or may not be allowed. Lee et al. [15] studied a two-agent problem with a common linear deterioration rate for all the jobs on a single machine. They provided the optimal solution and several heuristic algorithms for the problem of minimizing the total weighted completion time of the jobs of one agent, given that no tardy jobs are allowed for the other agent. Liu et al. [16] considered a two-agent single-machine problem with linear aging or learning effects. For the objective of minimizing the total completion time of one agent, given that the maximum cost of the other agent cannot exceed an upper bound, they presented polynomial-time algorithms. Wan et al. [17] considered several two-agent problems with controllable job processing times on a single machine or two identical machines in parallel. For several problems of minimizing the objective of one agent, including the total completion time plus the compression cost, the maximum tardiness plus the compression cost, the maximum lateness plus the compression cost, and the total compression cost, subject to a given upper bound on the objective function of the other agent, they provided the NP-hard proofs for the more general problems and presented polynomial-time algorithms for several special cases. Lee et al. [18] considered a two-agent scheduling problem in a two-machine permutation flowshop. Their objective is to minimize the total completion time of the jobs of one agent with the constraint that no tardy jobs are allowed for the other agent. They developed a branch-and-bound and simulated annealing algorithm to derive the optimal and near optimal solutions, for the problem, respectively.

However, in real-life situations, customer orders do not necessarily arrive at the same time. To the best of our knowledge, the multi-agent scheduling problem with consideration of release times has hardly been studied in the literature. In this paper we study a two-agent scheduling problem on a single machine with job release times where the objective is to minimize the total weighted completion time of the jobs of one agent, subject to the maximum lateness of the jobs of the other agent does not exceed a given limit. The research results on the classical single-machine scheduling problem with release times to minimize the total weighed completion time can be found in [19–23]. The rest of this paper is organized as follows: we present the problem formulation in the next section. We develop a branch-and-bound algorithm incorporating several dominance properties and a lower bound in Section 3. We propose a primary simulated annealing algorithm in Section 4. We introduce a secondary simulated annealing algorithm and report the computational results in Section 5. Finally, we conclude the paper in the last section and suggest topics for future research.

2. Problem description

The problem under study can be described as follows: there are n jobs to be processed on a single machine. Each job belongs to either one of the two agents, namely AG_1 and AG_2. Associated with each job j, there is a processing time p_j, a weight w_j, a release time r_j, a due date d_j, and an agent code I_j, where $I_j=1$ if $j\in AG_1$ and $I_j=2$ if $j\in AG_2$. Given a schedule S, let $C_j(S)$ be the completion time of job j, $L_j(S)=C_j(S)-d_j$ be the lateness of job j, and $L^2_{\max}(S)=\max_{j\in AG_2}\{L_j(S)\}$ be the maximum lateness of the jobs of agent AG_2. The objective of the problem is to find a schedule that minimizes the total weighted completion time of the jobs of AG_1 with the restriction that the maximum lateness of the jobs of agent AG_2 does not exceed a given upper bound M. Since the objective function is a regular scheduling performance measure, we use the terms schedule and sequence interchangeably in this paper. Using the conventional three-field notation for describing scheduling problems, we denote our problem as $1|L^2_{\max}\le M, r_j|\sum_{j\in AG_1} w_jC_j$.

3. A branch-and-bound algorithm

When the number of jobs of agent AG_2 is zero, the problem under consideration reduces to the classical single-machine scheduling problem with release times to minimize the total weighted completion time, which is NP-hard [24]. So it is justified that we deploy the branch-and-bound technique to solve the problem under study. In this section we first provide several dominance properties, followed by a lower bound to speed up the search process in the branch-and-bound scheme, and finally the description of the branch-and-bound algorithm.

3.1. Dominance properties

First, we provide some results that help reduce the size of solution space of the problem or determine the optimal schedule under certain conditions. Since the correctness of both Theorems 1 and 2 are easy to establish, we omit their proofs.

Theorem 1. *If there is a job i such that $r_i+p_i\le r_j$ for any other job j, then there is an optimal sequence in which job i is scheduled first.*

Let S^* be a sequence in which the jobs of AG_1 are scheduled first in non-decreasing order of their release times, followed by the jobs of AG_2 in non-decreasing order of their due dates. Let $C_{[k]}(S^*)$, $r_{[k]}(S^*)$, and $d_{[k]}(S^*)$ denote the completion time, release time, and due date of the kth scheduled job under schedule S^*, respectively.

Theorem 2. *If $C_{[k]}(S^*)\le r_{[k+1]}(S^*)$ for the jobs of AG_1 and $C_{[k]}(S^*)-d_{[k]}(S^*)\le M$ for the jobs of AG_2 , $k=1, \ldots, n$, then S^* is an optimal schedule.*

Next, we provide several adjacent dominance properties. Suppose that S and S' are two job schedules and the difference between S and S' is a pairwise interchange of two adjacent jobs i and j. That is, $S=(\pi,i,j,\pi')$ and $S'=(\pi,j,i,\pi')$, where π and π' each denote a partial sequence. In addition, let t denote the completion time of the last job in π. Depending on whether jobs i and j are from agent AG_1 or AG_2, we consider the following four cases:

Case 1. Both jobs i and j are from agent AG_1 in sequences S and S'.

To show that S dominates S', it suffices to show that $C_j(S)-C_i(S')\le 0$ and $w_iC_i(S)+w_jC_j(S)<w_jC_j(S')+w_iC_i(S')$ in this case.

Property 1.1. *If* $\max\{t,r_i\}\le r_j$, $\max\{t,r_i\}+p_i\ge r_j$ *, and* $p_i/w_i<p_j/w_j$ *, then S dominates S′ .*

Proof. Since $\max\{t,r_i\}\le r_j$, the completion times of jobs j and i in S' are

$$C_j(S')=r_j+p_j \tag{1}$$

and

$$C_i(S')=r_j+p_j+p_i \tag{2}$$