Computers & Operations Research 51 (2014) 208–217

Contents lists available at ScienceDirect

Computers & Operations Research

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

A scheduling problem with three competing agents

Wen-Chiung Lee [a], Jen-Ya Wang [b,*]

[a] Department of Statistics, Feng Chia University, Taichung, Taiwan, ROC
[b] Department of Computer Science and Information Management, Hungkuang University, Sha Lu, Taiwan, ROC

ARTICLE INFO

Available online 18 June 2014

Keywords:
Scheduling
Total weighted completion time
Makespan
Multiple-agent
Single-machine
Maintenance activity

ABSTRACT

Scheduling with multiple agents has become a popular topic in recent years. However, most of the research focused on problems with two competing agents. In this article, we consider a single-machine scheduling problem with three agents. The objective is to minimize the total weighted completion time of jobs from the first agent given that the maximum completion time of jobs from the second agent does not exceed an upper bound and the maintenance activity from the third agent must be performed within a specified period of time. A lower bound based on job division and several propositions are developed for the branch-and-bound algorithm, and a genetic algorithm with a local search is constructed to obtain near-optimal solutions. In addition, computational experiments are conducted to test the performance of the algorithms.

1. Introduction

In traditional scheduling, there are common goals to be minimized for all jobs. However, jobs or orders might come from several customers who have different objectives or priority requests. For example, Kubzin and Strusevich [1] pointed out in maintenance planning that the maintenance operations compete with real jobs for machine occupancy. Baker and Smith [2] claimed that the manufacturing department is concerned about finishing jobs before their due dates, while the research and development department is concerned about quick response time. Recently, Leung et al. [3] pointed out that several important scheduling problems, such as rescheduling or scheduling with availability constraints, can be formulated as two-agent scheduling problems.

Agnetis et al. [4] were the first authors to discuss scheduling problems with multiple agents. Many researchers have started their studies on this area, for instance, Yuan et al. [5], Cheng et al. [6,7], Ng et al. [8], Agnetis et al. [9,10]. Recently, Leung et al. [3] extended the single-machine two-agent problems of Agnetis et al. [4] to the case of multiple identical parallel machines. In addition, they discussed some single-machine problems where jobs may have different release dates, and preemptions may or may not be allowed. Lee et al. [11] studied a single-machine problem with deteriorating jobs and two competing agents. The objective is to minimize the weighted completion time of jobs from one agent given no tardy jobs are allowed for the other agent. They provided heuristic algorithms based on the weighted combination of the due date and the deterioration rate. Lee et al. [12] studied a two-agent two-machine flowshop problem where the objective is to minimize the total completion time of jobs from one agent given that no tardy job is allowed for the other agent. They used the simulated annealing approach to derive a heuristic solution. Wu et al. [13] studied a two-agent single-machine problem with learning effects. The objective is to minimize the total tardiness of jobs from the first agent given that no tardy job is allowed for the second agent. They developed heuristic algorithms based on the weighted combination of the due date and the processing time. For more recent development in scheduling with two competing agents, please refer to [14,15].

On the other hand, machines might become unavailable due to machines breakdown or preventive maintenance activity. Schmidt [16] was the first author to consider problems with maintenance period. Ma et al. [17] pointed out that a more realistic model should take machine maintenance activities into account. Recently, Wang and Wei [18] studied identical parallel machines scheduling problems with a deteriorating maintenance activity. They showed that problems to minimize the total absolute differences in completion times and the total absolute differences in waiting times remain polynomially solvable under the proposed model. Cheng et al. [19] studied unrelated parallel-machine scheduling with deteriorating maintenance activities. They proved problems to minimize the total completion time or the total machine load can be optimally solved in polynomial time. Lee and Kim [20] provided a heuristic algorithm for the single-machine problem that requires periodic maintenance with the objective of minimizing the number of tardy jobs. Yang et al. [21] studied the parallel-machine scheduling problem with aging effects and multi-maintenance activities simultaneously. They provided an algorithm for the problem to jointly determine the optimal maintenance frequencies

* Corresponding author. Postal address: No. 1018, Sec. 6, Taiwan Boulevard, Shalu District, Taichung City 43302, Taiwan, ROC. Tel.: +886 4 26348652x5417; fax: +886 4 26521921.
E-mail address: jywang@sunrise.hk.edu.tw (J.-Y. Wang).

http://dx.doi.org/10.1016/j.cor.2014.06.011
0305-0548/

and the optimal positions of the maintenance activities. Rustogi and Strusevich [22] presented polynomial-time algorithms for single machine problems with generalized positional deterioration effects and machine maintenance. For extensive studies regarding scheduling with maintenance activities, readers can refer to Lee [23] and Ma et al. [17].

Although the multiple agent scheduling problems have been widely discussed recently, the majority of the research focused on problems with two competing agents. As pointed by Leung et al. [3], the maintenance activities can be viewed as jobs from one agent. Thus, problems with more than two agents are more desirable and realistic. To the best of our knowledge, two-agent scheduling problems with the consideration of maintenance activity have never been studied before. In this paper, we consider a single-machine scheduling problem with three competing agents. The objective is to minimize the total weighted completion time of jobs from the first agent given that the maximum completion time of jobs from the second agent does not exceed an upper bound and the maintenance activity from the third agent must be performed within a specified period of time. This is motivated by the heat treatment process in the air tool manufacturing industries [24]. The hammers (jobs) need to go through a vacuum carburizing furnace (machine) in order to improve their hardness. Hammers might come from the manufacturing department to make air tools, or from the research department to test new products. Meanwhile, the vacuum carburizing furnace needs to change the oil rings and/or heating bar (maintenance activity) to prevent oil ring leak and/or to be able to achieve the designed temperature.

The rest of the paper is organized as follows. In the next section, the problem is stated. In Section 3, a lower bound is established, several dominance propositions are given, and a branch-and-bound algorithm is developed. In Section 4, a genetic algorithm with a local search is proposed. In Section 5, a computational experiment is conducted to evaluate the proposed algorithms. A conclusion is given in the final section.

2. Problem definition

The scheduling problem is described as follows. There are n jobs from three agents, i.e., AG_1, AG_2, or AG_3, and each job j has a processing time p_j. All jobs are available at time zero. Agent AG_3 has one maintenance job that needs to be performed within the specified interval $[A, B]$, where A and B are positive constants. Without loss of generality, we assume that the last job, job n is the maintenance activity. Moreover, each job j from AG_1 has a weight w_j. Jobs from AG_2 must be completed before M, where M is a positive constant. For each job j in a schedule π, its completion time is denoted by $C_j(\pi)$. The objective of this problem is to find a schedule that minimizes $\sum_{j \in AG_1} w_j C_j(\pi)$ subject to $\max_{j \in AG_2} \{C_j(\pi)\} \le M$ and $A + p_n \le C_n(\pi) \le B$.

3. Branch-and-bound algorithm

When there is no job from AG_2, all jobs from AG_1 have equal weights and $B - A = p_n$, it reduces to the total completion time with maintenance activity, which is proved to be NP-complete [23]. In this section, a lower bound and several dominance propositions are proposed to accelerate the execution speed of the branch-and-bound algorithm. First, we introduce the lower bound. Second, several dominance propositions are developed to eliminate unnecessary nodes or determine the sequence of the unscheduled jobs in the branching tree.

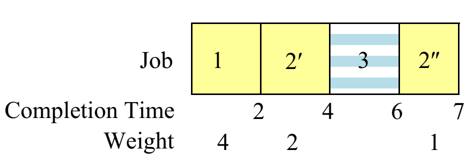

Fig. 1. An example of lower bound.

3.1. Lower bound

To obtain a lower bound, we borrowed an idea, *job division*, from Posner [25]. With this idea, we can perform a swap operation, like the bubble sort [26], on a schedule π to obtain a lower bound. An example to illustrate the lower bound is given in Fig. 1. There are 2 jobs from AG_1 with processing time $p_j = 2$, 3 and $w_j = 4$, 3 for $j = 1$ and 2 and there is a maintenance job 3 with processing time 2 which must be performed in $[A, B] = [4,6]$. For a simple schedule in Fig. 1, job 1 is processed first with a completion time of 2. On the other hand, the maintenance activity must be performed during [4,6] and a division of job 2 is inevitable. To obtain the lower bound, we divide job 2 into jobs 2′ and 2″ such that job 2′ can be filled into the interval [2,4] and the corresponding weights are also divided into $w_{2'} = 2$ and $w_{2''} = 1$. Unlike the concept of job preemption, we have two completion times for job 2, and they are $C_{2'} = 4$, $C_{2''} = 7$. Thus, the lower bound of the total weighted completion time of jobs 1, 2′ and 2″ is $2 \times 4 + 4 \times 2 + 7 \times 1 = 23$. The concept is described in the following definition.

Definition 1. (Job Division) If job i with processing times p_i is divided into two virtual jobs i' and i'' with processing times $p_{i'}$ and $p_{i''}$, respectively, where $p_{i'} + p_{i''} = p_i$, then its weight w_i is divided into $w_{i'} = w_i p_{i'} / p_i$ and $w_{i''} = w_i p_{i''} / p_i$.

In the following properties, we develop the lower bound for a schedule $\pi = (\alpha, \beta)$, where jobs in partial sequence α are scheduled. Depending on whether the maintenance activity is scheduled, and whether there are unscheduled jobs from agent AG_2, we divide it into 7 cases. In addition, we use $t = \max_{j \in \alpha} \{C_j(\pi)\}$ to denote the completion time of the last job in α, and an indicator function $I(Z) = 1$ if event Z occurs, and 0 otherwise.

Property 1. *If job $n \in \beta$, $AG_2 \cap \beta \ne \phi$, and $B - p_n \le M \le B + \sum_{j \in AG_2 \cap \beta} p_j$, then there is a virtual schedule $\pi^+ = (\alpha, \gamma, g', \delta, g'', \zeta)$ with a total weighted completion time of*

$$\sum_{j \in AG_1 \cap \alpha} w_j C_j(\pi^+) + \sum_{j \in \gamma} w_j C_j(\pi^+) + I(M - t$$
$$< \sum_{j \in \beta} p_j)[\lambda w_g C_{g'}(\pi^+) + (1-\lambda) w_g C_{g''}(\pi^+) + \sum_{j \in \zeta} w_j C_j(\pi^+)],$$

where γ and ζ are subsequences of jobs from AG_1, δ is a subsequence of jobs from $AG_2 \cup AG_3$, job $g \in AG_1$ with $p_{g'} = \lambda p_g$, and $\lambda = (M - t - \sum_{j \in \gamma} p_j - \sum_{j \in AG_2 \cap \beta} p_j - p_n) / p_g$.

Property 2. *If job $n \in \beta$, $AG_2 \cap \beta = \phi$, and $B - t > p_n$, then there is a virtual schedule $\pi^+ = (\alpha, \gamma, g', n, g'', \zeta)$ with a total weighted completion time of*

$$\sum_{j \in AG_1 \cap \alpha} w_j C_j(\pi^+) + \sum_{j \in \gamma} w_j C_j(\pi^+) + I(B - t$$
$$< \sum_{j \in \beta} p_j)[\lambda w_g C_{g'}(\pi^+) + (1-\lambda) w_g C_{g''}(\pi^+) + \sum_{j \in \zeta} w_j C_j(\pi^+)],$$

where γ and ζ are subsequences of jobs from AG_1, job $g \in AG_1$ with $p_{g'} = \lambda_g$, and $\lambda = (B - t - \sum_{j \in \gamma} p_j - p_n) / p_g$.

Property 3. *If job $n \in \beta$, $AG_2 \cap \beta \ne \phi$, $B - p_n > M$, and $\sum_{j \in \beta} p_j \le B - t$, then there is a virtual schedule $\pi^+ = (\alpha, \gamma, g', \delta, g'', \zeta, n)$ with a total*