Computers & Operations Research 39 (2012) 1644–1651

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

# Solving a two-agent single-machine scheduling problem considering learning effect

Der-Chiang Li [a], Peng-Hsiang Hsu [a,b,*]

[a] Department of Industrial and Information Management, National Cheng Kung University, No. 1, University Road, Tainan City 70101, Taiwan, ROC
[b] Department of Business Administration, Kang-Ning Junior College, Taipei, Taiwan

ARTICLE INFO

Available online 25 October 2011

Keywords:
Scheduling
Single-machine
Two-agent
Learning effect
Genetic algorithm

ABSTRACT

Scheduling with multiple agents and learning effect has drawn much attention. In this paper, we investigate the job scheduling problem of two agents competing for the usage of a common single machine with learning effect. The objective is to minimize the total weighted completion time of both agents with the restriction that the makespan of either agent cannot exceed an upper bound. In order to solve this problem we develop several dominance properties and a lower bound based on a branch-and-bound to find the optimal algorithm, and derive genetic algorithm based procedures for finding near-optimal solutions. The performances of the proposed algorithms are evaluated and compared via computational experiments.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

Many scheduling problems are solved conventionally in a one-agent environment, but this assumption is impractical in some real life situations. Consequently, Curiel et al. [1] and Hamers et al. [2] proposed that a scheduling problem can be formulated as a sequencing game in a multi-agent environment, while minimizing the cost savings with certain distribution rules to encourage the agents to cooperate. Schultz et al. [3] pointed out that in telecommunication services the task of S-UMTS is to fulfill the service demands from different agents, who compete for the use of a microwave system to transfer voice and web browsing files for their clients.

Baker and Smith [4] were among the pioneers that brought the concept of multiple agents into the scheduling field. They introduced a linear combination of a scheduling problem with two agents and different basic criteria. Agnetis et al. [5] then introduced some application environments and methodological fields in which multiple agents competed for the usage of shared processing resources. They analyzed the complexity of several two-agent single-machine problems with different combinations of objective functions, in which each agent had individual criterion to optimize. Yuan et al. [6] further revised the two polynomial-time dynamic programming recursions in Baker and Smith's research [4] for the two-family jobs problem. Cheng et al. [7] presented a strong NP-complete problem in which multi-agent jobs were scheduled on a single machine and the objective of each agent was to minimize the total weighted number of tardy jobs. Ng et al. [8] considered a two-agent scheduling problem on a single machine in which the objective was to minimize the total completion time of one agent with the restriction that the number of tardy jobs of the other agent was bounded. Agnetis et al. [9] investigated the complexity of some scheduling problems in which several agents with non-preemptive job sets were competing to perform their respective jobs on a single shared processing resource. Cheng et al. [10] considered a single machine multi-agent scheduling problem in which the agent's objective functions have the type of max-form and proved NP-hard for some minimalistic models. Agnetis et al. [11] designed branch-and-bound algorithms by a Lagrangian approach which provided a good bound to deal with two-agent scheduling problems for minimizing total weighted completion time of one agent subject to various constraints on the second agent's performance. According to their study, the scheduling problem could be solved in a strongly polynomial time. Lee et al. [12] focused on a multi-agent scheduling problem on a single machine in which each agent tried to minimize the total weighted completion time of its own set of jobs. They introduced some approximate results with the fully polynomial time approximation schemes and provided an efficient approximation algorithm with a reasonable worst case ratio. More recently, Leung et al. [13] generalized the single machine scheduling problems with multiple agents based on the results of Agnetis et al. [5] and considered the model in a parallel machines environment. Mor and Mosheiov [14] focused on a scheduling problem with two competing agents to minimize the maximum earliness cost of one agent. They showed that an upper bound on the maximum earliness cost of the agent yielding a polynomial time solution.

In addition to the multi-agent scheduling environment, learning effects have also received a lot of research attention since

* Corresponding author.
E-mail address: r38991036@mail.ncku.edu.tw (P.-H. Hsu).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.09.018

Wright [15] first examined them in the aircraft industry, and, Biskup [16] and Cheng and Wang [17] brought the concept of such effects into the field of scheduling. Since then, scheduling with consideration of learning has received growing attention. For example, Mosheiov [18] extended Biskup's [16] model and found that some optimal scheduling problems could be solved in polynomial time. Mosheiov and Sidney [19] further studied a job-dependent learning curve where the learning in the production process of some jobs was faster than those of others. Bachman and Janiak [20] studied some single machine with learning effect scheduling models for total weighted completion time problems. They showed that there were polynomial time solutions in some special cases and proved the strong NP-hardness of the makespan minimization problem with release time for two different models of job processing time. In addition, Kuo and Yang [21], Koulamas and Kyparisis [22] developed two different types of general sum-of-processing-times-based learning of all jobs already processed. Biskup [23] provided a comprehensive survey of research on scheduling with learning effects, and for more recent results can be found in papers by Ghodratnama et al., Janiak and Rudek, Ji and Cheng, Koulamas, Okołowski and Gawiejnowicz, Toksarı and Wu et al. [24–30].

However, works on the multi-agent scheduling problems with learning effects are rather limited. Liu et al. [31] studied the optimal polynomial time algorithms for a two-agent single machine scheduling model with position-dependent processing time aging and learning effect, with the objective of minimizing the total completion time of the first agent with a maximum cost limit of the second agent. Cheng et al. [32] studied a two-agent single machine scheduling problem with a sum-of-processing-times-based learning effect function as the control parameter, and developed algorithms to minimize the total weighted completion time of the jobs considering that no tardy job is allowed for the second agent. In this paper, we investigate a rather general problem where two agents are included with position-based learning competing for the usage of a common single machine. The objective is to minimize the total weighted completion time of both agents with the restriction that the makespan of either agent cannot exceed an upper bound. The complexity of the proposed problem might be still an open question.

The remainder of this paper is organized as follows. In the next section, a description of the problem is given, while several dominance properties and lower bounds relating to the branch-and-bound approach are introduced in Section 3. In Section 4, genetic algorithms for obtaining near-optimal solutions are described in detail. The computational experiments are conducted in Section 5, and the conclusions and discussions are presented in the last section.

## 2. Problem formulation

There are two agents $A$ and $B$, and $n$ jobs $J_1^X J_2^X,\ldots J_n^X$ to be scheduled on a single machine, where $X$ represents the agents, that is $X=A$ or $B$. Some of the jobs $J^A=\{J_1^A J_2^A,\ldots J_{n^A}^A\}$ belong to agent A and others $J^B=\{J_1^B J_2^B,\ldots J_{n^B}^B\}$ belong to agent B, where the numbers of jobs are $n^A$ and $n^B$, and $n=n^A+n^B$. Each of the jobs $J_i^X$ is assigned with a weight $w_i^X$ and a basic processing time $p_i^X$, where $1\le i\le n$. The real processing time $p_{ir}^X$ of job $J_i^X$ varies with position $r$ based on the learning effect, that is $p_{ir}^X=p_i^X r^a$, where $a$ is the learning ratio with $a<0$ and $r=1,2,\ldots,n$.

Moreover, let $C_i^X(S)$ and $C_{\max}^X$ be the completion time of $J_i^X$ in a job sequence $S$ and the makespan of $J^X$, respectively. The objective of this paper is to minimize the total weighted completion time while keeping the makespan of agent $A$ or $B$ less than a bound $U$. The problem can be notated by a triplet as $1\|\sum_{J_i^X\in J_i^A\cup J_i^B} w_i^X C_i^X(S): C_{\max}^A\le U$. Agnetis [5] showed that $1\|\sum w_i C_i^A(S): f_{\max}^B$ is binary NP-hard, even when the objective function of agent B is maximum completion time, i.e., $f_{\max}^B=C_{\max}^B$. It implies that the complexity of $1\|\sum w_i^{A\cup B} C_i^{A\cup B}(S): C_{\max}^B$ is at least binary NP-hard. Our proposed problem with learning consideration is more difficult than the $1\|\sum w_i^{A\cup B} C_i^{A\cup B}(S): C_{\max}^B$ problem. We will apply a branch-and-bound algorithm to obtain the optimal solution, and propose a genetic algorithm based procedure for the near-optimal solutions.

## 3. Dominances and low bound for a branch-and-bound algorithm

Firstly, we derive the optimal solution of the problem using a branch-and-bound algorithm with several dominance properties developed in order to speed up the searching process. In addition, we will come up with a lower bound to curtail the branching for more efficient processing.

### 3.1. Dominance properties

In this subsection we develop four dominance properties based on a pairwise interchange argument. Suppose that there are two job sequences $S_1=(\pi,i,j,\pi')$ and $S_2=(\pi,j,i,\pi')$, where $\pi$ and $\pi'$ denote the scheduled and unscheduled sequence, respectively. Moreover, in $S_1$, jobs $J_i^X$ and $J_j^X$ are in the $k$th and $(k+1)$th positions, respectively, while in $S_2$, jobs $J_i^X$ and $J_j^X$ are in the $(k+1)$th and $k$th positions. To show that $S_1$ dominates $S_2$ on time total completion, it suffices to check three regularity conditions, namely $C_j^X(S_1)<C_i^X(S_2)$, $[w_i^X C_i^X(S_1)+w_j^X C_j^X(S_1)]\le[w_i^X C_i^X(S_2)+w_j^X C_j^X(S_2)]$ and $C_j^X(S_1)\le U$. Thus, we set the following property and let $t$ denote the completion time in schedule sequence $\pi$ containing $(k-1)$ jobs as:

**Property 1.** *If jobs $J_i^X$, $J_j^X\in J^A$, $p_j^A/p_i^A\ge w_j^A/w_i^A>1$, and $t+p_i^A k^a+p_j^A(k+1)^a\le U$, then $S_1$ dominates $S_2$.*

**Proof.** By the definition, the completion times of jobs $J_i^X$ and $J_j^X$ from agent $A$ in $S_1$ and $S_2$ are, respectively, as

$$C_i^X(S_1)=t+p_i^A k^a \tag{1}$$

$$C_j^X(S_1)=t+p_i^A k^a+p_j^A(k+1)^a \tag{2}$$

$$C_j^X(S_2)=t+p_j^A k^a \tag{3}$$

$$C_i^X(S_2)=t+p_j^A k^a+p_i^A(k+1)^a \tag{4}$$

Since $p_j^A/p_i^A\ge w_j^A/w_i^A>1$ and $a<0$, we have

$$C_j^X(S_1)<C_i^X(S_2) \tag{5}$$

and

$$[w_i^X C_i^X(S_1)+w_j^X C_j^X(S_1)]\le[w_i^X C_i^X(S_2)+w_j^X C_j^X(S_2)] \tag{6}$$

and since $t+p_i^A k^a+p_j^A(k+1)^a\le U$, it implies that

$$C_j^X(S_1)<U \tag{7}$$

By Eqs. (5)–(7), we conclude that $S_1$ dominates $S_2$. □

The proofs of Properties 2–4 are omitted since they are similar to that of Property 1.

**Property 2.** *Given jobs $J_i^X$, $J_j^X\in J^B$, if $p_j^B/p_i^B\ge w_j^B/w_i^B>1$, then $S_1$ dominates $S_2$.*

Download English Version:

https://daneshyari.com/en/article/10348148

Download Persian Version:

https://daneshyari.com/article/10348148

Daneshyari.com