Applied Mathematical Modelling 39 (2015) 2587–2601

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

A honey-bees optimization algorithm for a two-agent single-machine scheduling problem with ready times

Yunqiang Yin [a], Wen-Hung Wu [b], T.C.E. Cheng [c], Chin-Chia Wu [d], Wen-Hsiang Wu [e,*]

[a] Faculty of Science, Kunming University of Science and Technology, Kunming 650093, China
[b] Department of Business Administration, Kang-Ning Junior College, Taipei, Taiwan
[c] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[d] Department of Statistics, Feng Chia University, Taichung, Taiwan
[e] Department of Healthcare Management, Yuanpei University of Medical Technology, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form 5 July 2014
Accepted 23 October 2014
Available online 21 November 2014

Keywords:
Scheduling
Two agents
MBO algorithm

ABSTRACT

In this paper we consider two agents that compete on the use of a common processor. Each of the agents has a set of jobs that have to be processed on the same machine without preemption. Each of the agents wants to minimize an objective function that depends on the completion time of its own jobs. In addition, each job has different release dates. In the presence of unequal release dates, it is sometimes advantageous to form a non-full batch, while in other situations it is a better strategy to wait for future job arrivals in order to increase the fullness of the batch. The objective is to find a schedule that performs well with respect to the objectives of both agents. To solve this difficulty problem, we construct a branch-and-bound solution scheme incorporating these bounds and some dominance rules for the optimal solution. In view of the advantage of combining local and global searches in the honey-bees optimization algorithm, we attempt to use a marriage in honey-bees optimization algorithm (MBO) to find near-optimal solutions. We conduct extensive computational experiments to evaluate the performance of the algorithms.

1. Introduction

A branch of artificial intelligence, multi-agent simulation provides a promising approach to deal with multi-stakeholder management systems such as allocation of a common pool of resources to multiple stakeholders. It provides a framework in which stakeholders' (or agents') interactions and decision-making behaviors can be analyzed. Examples of such a system are given by Purnomo and Guizol [1] in the context of forest plantation co-management, and by Bessonov et al. [2] concerning blood cell population dynamics. More examples in different application environments and different methodological fields, such as artificial intelligence, decision theory, and operations research, can be found in Agnetis et al. [3].

Scheduling with multiple agents has received considerable research attention recently. Baker and Smith and Agnetis et al. [4,5] pioneer multi-agent scheduling research. Baker and Smith [4] investigate a linear combination of the objectives of the two agents. Agnetis et al. [5] study the complexity of several two-agent single-machine problems in which one agent seeks to optimize its objective, given that the objective of the other agent cannot exceed a certain value. Cheng et al. [6] consider a feasibility model of multi-agent single-machine scheduling where each agent's objective is to minimize its total weighted

* Corresponding author.
E-mail address: wenhsiang_wu@yahoo.com.tw (W.-H. Wu).

http://dx.doi.org/10.1016/j.apm.2014.10.044

number of tardy jobs. Cheng et al. [7] study multi-agent single-machine scheduling where the objectives of the agents are of the max-form. Ng et al. [8] consider two-agent single-machine scheduling where the objective is to minimize the total completion time of the first agent, subject to the number of tardy jobs of the second agent cannot exceed a given number. Agnetis et al. [3] analyze the complexity of some multi-agent single-machine scheduling problems and propose solution algorithms. Studying multi-agent single-machine scheduling to minimize the total weighted completion time, Lee et al. [9] provide fully polynomial-time approximation schemes and an efficient approximation algorithm with a reasonable worst-case bound. Leung et al. [10] generalize the single-machine problems proposed by Agnetis et al. [5] to the environment with multiple identical machines in parallel and different release dates. Wan et al. [11] consider several two-agent scheduling problems with controllable job processing times, where two agents share either a single machine or two identical machines in parallel for processing their jobs. Mor and Mosheiov [12] deal with a scheduling problem with two competing agents to minimize the minmax and minsum earliness objectives. Nong et al. [13] consider two-agent single-machine scheduling to minimize the total cost, which comprises the maximum weighted completion time of one agent and the total weighted completion time of the other agent. They present a 2-approximation approximation algorithm for the problem, show that the case where the number of jobs of the first agent is fixed is NP-hard, and devise a polynomial-time approximation scheme for this case. Cheng et al. [14] consider two-agent single-machine scheduling involving deteriorating jobs and learning effects simultaneously. In the proposed model, given a schedule, they assume that the actual processing time of any job of the first agent is a function of position-based learning, while the actual processing time of any job of the second agent is a function of position-based deterioration. Cheng et al. [15] investigate two-agent single-machine scheduling in which the actual processing time of a job in a schedule is a function of sum-of-processing-times-based learning and a control parameter of the learning function. The objective is to minimize the total weighted completion time of the jobs of the first agent, subject to no tardy job for the second agent. Wu et al. [16] study two-agent single-machine scheduling in which sum-of- processing-times-based learning and job deteriorating co-exist. Luo et al. [17] consider two-agent two-machine flow shop scheduling. They investigate two models where one is a weighted-sum optimization model and the other is a constrained optimization model.

Despite the popularity of studies of multi-agent scheduling, research on multi-agent scheduling with job release dates is relatively uninvestigated. Yin et al. [18] address a two-agent single-machine scheduling problem with different ready times. Their objective function is to minimize the tardiness of one agent, subject to the constraint that the lateness of the other agent is less than upper bound. They use a mixed integer programming model and branch-and-bound algorithm to solve the problem. In addition, they propose a honey-bees optimization algorithm to obtain approximate solutions. Yin et al. [19] address a two-agent scheduling problem on a single machine where the objective is to minimize the total weighted earliness cost of all jobs, while keeping the earliness cost of one agent below or at a fixed level. An application of multi-agent scheduling with job release dates arises in the shipping industry ([20–23]). Specifically, ships belonging to different shipping companies (multiple agents) call at a port, which needs to determine the sequence to serve the ships that arrive at different times. In this case, the single machine is the port that processes jobs with ready times, which are the ships that arrive at different times. Motivated by this observation, we consider a two-agent single-machine scheduling problem with job release dates, where the goal is to minimize the weighted sum of the completion times of the jobs of the first agent, subject to the maximum lateness of the jobs of the second agent cannot exceed a given limit. We present a branch-and-bound (BAB) solution scheme and a marriage in honey-bees optimization (MBO) algorithm to solve the problem optimally and approximately, respectively. The contributions of this study are that we first discuss a polynomially solvable case of the proposed problem, then we develop nine dominance properties and a lower bound for the optimal solution in the theoretical part and propose an MBO algorithm to find near-optimal solutions in the technical part.

The rest of the paper is organized as follows: In Section 2 we introduce and formulate the problem. In Section 3 we discuss the complexity and solvability of the problem. In Section 4 we present a BAB solution scheme and an MBO algorithm to solve the problem. We discuss the results of computational experiments conducted to evaluate the performance of the algorithms in Section 5. We conclude the paper and suggest some topics for future research in the last section.

2. Optimization problem formulation

In this section we first introduce the notations used throughout the paper and then give the formulation of the problem.

n denotes the number of jobs;
S denotes a sequence of jobs;
X denotes the set of agents, say A and B;
J^X denotes the job set of agent X;
J_j^X denotes the job code of job j of agent X;
p_j^X denotes the job processing time of job J_j^X of agent X;
r_j^X denotes the release date of job J_j^X of agent X;
w_j^X denotes the weight of job J_j^X of agent X;
d_j^X denotes the due date of job J_j^X of agent X;