Minimizing the sum of makespan on multi-agent single-machine scheduling with release dates

Xinyue Wang[1,2], Feng Chu[1], Tao Ren[2], Danyu Bai[3]

[1] Laboratory IBISC, Univ-Évry, Université Paris-Saclay, Paris, France
[2] Software College, Northeastern University, Shenyang, China
[3] School of Maritime Economics & Management, Dalian Maritime University, Dalian, China
chinaxywang@163.com, feng.chu@univ-evry.fr, chinarentao@163.com, mikebdy@163.com

Keywords: *Multi-agent scheduling, Release date, Branch and bound, Artificial bee colony algorithm, Deep Reinforcement learning*

1 Introduction

Recently, multi-agent scheduling (MAS) problems [1] with multiple agents competing on the usage of a common resource have received increasing attention in various fields, such as industrial production, transportation, and telecommunication, etc. Different from traditional scheduling models which mainly focus on the optimal allocation of processor resources to achieve an optimal indicator, MAS models consider that multiple agents may be competing for a shared processing resource and the objective is to globally achieve the satisfaction of each agent. Given the NP-hardness of the MAS problems, different types of algorithms are suggested, including a branch and bound (B&B) algorithm, a hybrid discrete artificial bee colony (HDABC) algorithm, and a deep reinforcement learning (DRL) method.

2 Problem description

This research investigates the following MAS versions. Firstly, considering that each agent has a release date and the jobs maintained by the same agent arrive simultaneously. The objective is to minimize the sum of makespans belonging to several agents individually. This version is denoted as problem P_1. The weighted version (denoted as problem P_2) supposes that each agent has a weight and each job has its own release date to minimize the sum of the weighted makespans of all agents. The periodic maintenance version (denoted as problem P_3) assumes that the processor is required to be maintained at regular intervals and each job has its release date. The objective is to minimize the sum of makespans of all agents. In reality, problems P_2 and P_3 can be seen as the extensions of problem P_1.

3 Solution method

For problem P_1, a B&B algorithm is developed to obtain optimal solutions. To enhance the performance of the B&B algorithm, a release date-based pruning rule and a preemption-based lower bound are designed to narrow the computing space. Firstly, for obvious reasons, if a job which has arrived is artificially postponed until the completion of a later release job, in this case, an optimal solution cannot be obtained. Based on this phenomenon, the release date-based pruning rule is proposed.

Secondly, the preemption version of problem P_1 can be optimally solved in polynomial time by the preemptive available dominate agent first (PADAF) rule, which provides insights for calculating the lower bound.

For problem P_2, a HDABC algorithm is designed to achieve satisfactory solutions. In the HDABC algorithm, the heuristic-based initialization guarantees the population diversity and avoids the local optima, the variable neighborhood search-simulated annealing (VNS-SA) strategy explores the neighborhood of a population subset, the agent block-based insertion combines the features of the proposed problem and performs targeted perturbations on the solution.

For problem P_3, a DRL method that combined the pointer network and the policy gradient is introduced to obtain high-quality solutions in dynamic situation. In the DRL, five features are selected as the input to the network, including processing time, release date, agent, weight, and maintenance cycle. In addition, Adam is used to optimize the training.

4 Experiment results

The B&B algorithm is compared with CPLEX to highlight the superiority of the forme. The B&B algorithm solve all instances within 30s, of which 86.7% (16 out of 120) instances are solved within 1s. However, CPLEX cannot obtain the optimal solution of any instance within 1800s.

Three state-of-the-art metaheuristics are introduced to compare with the HDABC algorithm, namely, DABC [2], DDE [3], and PSO [4]. The results show that the HDABC algorithm achieves the smallest objective values than others for all instances. Furthermore, the component analysis verifies the efficiency of the proposed strategies.

A total of 90 instances were executed with the proposed DRL, DABC [2], DDE [3], and PSO [4]. The DRL achieves the minimum value of almost all instances, revealing its effectiveness. In addition, with the problem scale enlarging, the CPU times of the metaheuristics increase significantly, while DRL can always solve the problem within 1s.

5 Conclusions et perspectives

This research proposes an MAS model with release dates to minimize the sum of makespans belonging to several agents individually. Both the weighted and periodic maintenance versions are investigated in this article. The main contribution of the article is to develop different types of algorithms. A series of comparative experiments and statistical comparisons confirm the efficiency of the algorithms.

References

[1] Agnetis A, Mirchandani P B, and Pacifici P A. Scheduling Problems with two Competing Agents. Operations Research, 52 (2): 229–242, 2004.

[2] Bai D, Diabat A, Wang X, Yang D, Fu Y, Zhang Z, Wu C. Competitive bi-agent flowshop scheduling to minimise the weighted combination of makespans, International Journal of Production Research, 5: 1-22, 2021.

[3] Zhao F, Zhao L, Wang L, Song H. An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion. Expert Systems with Applications, 160, 2020.

[4] Ren T, Zhao P, Zhang D, Liu B, Yuan H, Bai D. Permutation flow-shop scheduling problem to optimize a quadratic objective function. Engineering Optimization, 49: 1589-1603, 2017.