Expert Systems With Applications 42 (2015) 9333–9339

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents

Deming Lei[a,*], Xiuping Guo[b]

[a] School of Automation, Wuhan University of Technology, Wuhan 430070, PR China
[b] School of Economic and Management, Southwest Jiaotong University, Chengdu 610031, PR China

ARTICLE INFO

Keywords:
Hybrid flow shop scheduling
Shuffled frog-leaping algorithm
Minimality model
Two agents

ABSTRACT

Two-agent scheduling problem has attracted much attention; however, two-agent hybrid flow shop scheduling problem (TAHFSP) is seldom studied. In this study, TAHFSP and its minimality model are considered and a novel shuffled frog-leaping algorithm (SFLA) is proposed is to minimize the sum of the objectives of two agents. The following new principles are applied in SFLA: a tournament selection based method is used to divide population, not all solutions of population are allocated into memeplexes, the best solution of each memeplex acts as the object of the search process and the search process within memeplex consists of the global searches on two sub-problems sequentially and multiple neighborhood search. We test the performance of SFLA using a number of instances and the experimental results show the notable advantage of our SFLA when compared to other algorithms of TAHFSP.

1. Introduction

Generally, multi-agent scheduling problems are composed of several groups of jobs and some necessary processing resources such as machines. Each group of jobs belong to an agent, all agents compete on the use of common processing resources and each of them wishes to minimize an objective function that depends on the completion time of his/her own set of jobs. The goal of multi-agent problem is either to find a schedule that minimizes a combination of the agents' objective functions or to find a schedule that satisfies each agent's requirement for his/her own objective function.

Since Baker and Smith (2003) and Agnetic, Mirchandani, Pacciarelli, and Pacifici (2004) first introduce the multi-agent scheduling problem, the problem has attracted much attention in the past decade and a large body of literature discusses the problem in single machine, parallel machines and flow shop environments. With respect to multi-agent single machine scheduling, Cheng, Ng, and Yuan (2008) discuss the feasibility model and the minimality model. Lee, Chung, and Hu (2012) propose three genetic algorithms (GA) for two-agent scheduling with release time. The objective is to minimize the total tardiness of jobs from the first agent given that the maximum tardiness of jobs from the second agent does not exceed an upper bound. Yin, Cheng, Cheng, Wu, and Wu (2012) consider several two-agent scheduling problems with assignable due dates and give polynomial-time algorithm. Liu, Yi, Zhou, and Gong (2013) discuss the optimal properties of two-agent scheduling with sum-of-processing-times-based deterioration and present some polynomial time algorithms. Wu et al. (2013) propose branch-and-bound (BB) algorithm and tabu search for two-agent problem with deterioration jobs.

Yin and his partners carry out some related works on two-agent problem in single machine (Wu et al. 2013; Yin et al. 2012; Wang et al., 2015). They provide complexity results for two-agent problems with due date determination, propose a BB algorithm and a marriage in honey bees optimization algorithm for the problem with ready times and present a GA for the problem with learning effects. Feng, Fan, Li, and Shang (2015) discuss NP-hard features of two-agent problem with rejection and present some pseudo-polynomial-time algorithms. Oron, Shabtay, & Steiner, 2015 also prove the NP-hard complexity and provide some pseudo-polynomial-time algorithms for the problem with equal processing times.

With respect to parallel machines multi-agent scheduling, Elvikis, Hamacher, and T'kindt (2011) discuss two-agent scheduling on uniform parallel machines with two conflicting objectives and provide polynomial time algorithms for solving the problem. Li and Yuan (2012) consider the constrained optimization problem with an unbounded parallel-batch machines and two agents and provide polynomial-time and pseudo-polynomial-time algorithm. Fan, Cheng, Li, and Feng (2013) study the NP-hard feature and polynomial solvability of bounded parallel-batching scheduling with two competing agents and different objectives. Zhao and Lu (2015) propose two approximation algorithms for two-agent parallel machines scheduling

[*] Corresponding author. Tel.: +86 153 273 11013.
E-mail addresses: deminglei11@163.com (D. Lei), guoxiuping0029@sina.com (X. Guo).

http://dx.doi.org/10.1016/j.eswa.2015.08.025
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

Some studies have considered multi-agent scheduling in flow shop environments. Lee, Chen, and Wu (2010) consider two-agent two-machine flow shop scheduling problem and develop a BB algorithm and simulated annealing (SA) to minimize the total tardiness of the first agent with no tardy jobs for the second agent. Lee, Chen, Cheng, and Wu (2011) study a two-machine flow shop problem with two agents where the objective is to minimize the total completion time of first agent with no tardy jobs for the second agent. Luo, Chen, and Zhang (2012) investigate the weighted-sum optimization model and the constrained optimization model and study approximation schemes for two-machine flow shop scheduling with two agents. Mor and Mosheiov (2014) study polynomial time solution algorithms for the problem on a proportionate flow shop with two agents. Shiau, Tsai, Lee, and Cheng (2015) present some GAs two-agent two-machine flow shop with learning effects. Lei (2015a, b) propose a variable neighborhood search (VNS) and a two-phase neighbor-hood search for two-agent scheduling in flow shop and hybrid flow shop respectively to simultaneously minimize objectives of two agents under the given upper bound.

As stated above, the previous studies on multi-agent production scheduling have the following features: (1) Most of papers have been presented for multi-agent scheduling on single machine. Multi- agent flow shop scheduling also attract some attention; however, the existing researches mainly focus on two-agent two-machine flow shop scheduling and multi-agent scheduling is not fully considered in hybrid flow shop, which is common manufacturing environments. (2) The pseudo-polynomial-time algorithm and BB algorithms are often applied to solve the problem. Several meta-heuristics such as GA (Lee et al., 2012), VNS (Lei, 2015a) and SA (Lee et al., 2010) also have been used to deal with multi-agent scheduling; however, for multi-agent scheduling, the applications of meta-heuristics are not extensive and the advantages of meta-heuristics such as SFLA are not fully investigated.

SFLA is based on observing, imitating and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food. SFLA is originally developed by Eusuff, Lansey, and Pasha (2006). SFLA has been applied to production scheduling problems (Rahimi-Vahed, Dangchi, Rafiei, & Salimi, 2009;Pan, Wang, Gao, & Li, 2011; Li, Pan, & Xie, 2012; Fang & Wang, 2013); however, to the best of our knowledge, SFLA is not applied to multi-agent scheduling. The main benefits of SFLA are its fast convergence speed and effective algorithm structures containing local search and global information exchanges. The advantages of SFLA and the extensive applications of hybrid flow shop motivate us to consider two-agent hybrid flow shop scheduling problem (TAHFSP) and apply some new principles to design SFLA for the problem. The objective is to minimize the sum of the objectives of two agents.

The remainder of the paper is organized as follows. Problem under study is described in Section 2 and followed by the description on basic SFLA (BSFLA) in Section 3. The novel SFLA for the problem is described in Section 4. Numerical test experiments on SFLA are reported in Section 5 and the conclusions are summarized in the final section and some topics of the future research are provided.

2. Problem description

Hybrid flow shop scheduling problem (HFSP) is generally to optimize the processing of a set of n jobs in a series of m stages in terms of certain objectives. More than one machine exists at least one stage. HFSP is quite common in practice, especially in the process industry. In general, it is assumed that all jobs of HFSP come from the same agent; however, what frequently happens is that more than one agent provides processing tasks for the same manufacturer.

TAHFSP is composed of n jobs and m stages.n jobs $J_1, J_2, \ldots, J_n$ are processed according to the same production flow: stage 1, stage 2, … , stage m. In each stage k,$|M_k| \geq 1$ machines are in parallel and $|M_k| > 1$ for $\exists k \in \{1, 2, \ldots, m\}$, where M_k is the set of all parallel machines in stage k. Each machine can only process one job at a time.S_1 and S_2 indicate the set of jobs of the first agent and the second agent respectively, $J_i \in S_1$ or $J_i \in S_2$.p^l_{ihk} indicates the processing time of job $J_i \in S_l$ on machine h of stage k ,$l = 1, 2$, $h \in [1, |M_k|]$.C^l_i and d^l_i indicate the completion time and due date of job $J_i \in S_l$.

Table 1
An illustrative example.

Job	Operation	Processing time						S_l	d^l_i
		Stage1			Stage2		Stage3		
		M_1	M_2	M_3	M_4	M_5	M_6		
J_1	o_{11}	4	5	2	8	7	8	S_2	130
	o_{12}	6	3	5	12	11	7		
	o_{13}	4	6	5	9	4	1		
J_2	o_{21}	11	14	11	8	3	6	S_1	140
	o_{22}	8	7	4	5	10	9		
	o_{23}	12	23	22	56	40	19		
J_3	o_{31}	15	9	6	6	4	11	S_2	124
	o_{32}	14	34	31	2	3	1		
	o_{33}	23	31	28	7	8	9		
J_4	o_{41}	10	8	9	15	12	6	S_1	115
	o_{42}	7	8	7	22	18	10		
	o_{43}	12	15	6	1	1	2		

Processing constraints of HFSP can be directly adopted in TAHFSP, for example, each machine can only process one job at a time. The goal of TAHFSP is to obtain an appropriate processing sequence and machine assignment of all jobs to minimize the sum of tardiness of all jobs $J_i \in S_1 \cup S_2$.

$$\text{Minimize } T_{tot} = f_1 + f_2 \tag{1}$$

where T_{tot} indicates the total tardiness of all jobs, $f_1 = \sum_{J_i \in S_1} \max\{C^1_i - d^1_i, 0\}$ is the objective of the first agent, $f_2 = \sum_{J_i \in S_2} \max\{C^2_i - d^2_i, 0\}$ is the objective of the second agent.

Table 1 shows an illustrative example of TAHFSP.

There are two models for multi-agent scheduling.

(1) Feasibility model. In this model, objectives should satisfies $f_l \leq Q_l$,$l = 1, 2, \ldots, D$
(2) Minimality model. In this model, the sum of all objectives $\sum_{l=1}^{D} f_l$ should be minimized.

where f_l and Q_l indicate the objective function and upper bound of the lth agent, respectively, D is the number of agents, in this study, $D = 2$.

Feasibility model is first studied by Agnetic et al. (2004) and minimality model is first considered by Baker and Smith (2003). When feasibility model is considered, objectives of the multi-agent scheduling problem should be optimized simultaneously and the problem is a multi-objective one. When multiple agents have the same kind of objective such as total tardiness and maximum tardiness and the sum of all objectives is meaningful, the problem is simplified and easier to be solved than the problem with feasibility model. The minimality model is a good choice in this case. Cheng et al. (2008) and Yin et al., 2012 considered the minimality model, in this study, the minimality model is also adopted.

3. Basic shuffled frog-leaping algorithm

In SFLA, the solution x of the problem is defined as the position of a frog, there is a population of possible solutions defined by a set of virtual frogs and the population is decomposed into different groups named memeplexes. Each memeplex then performs a search process. After a defined number of iterations are done within each memeplex, memeplexes are combined into a new population and the population is decomposed again, this process is called shuffling one. The search