Computers & Industrial Engineering 80 (2015) 125–131

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Variable neighborhood search for two-agent flow shop scheduling problem ☆

Deming Lei *

School of Automation, Wuhan University of Technology, China

ARTICLE INFO

Article history:
Received 7 January 2014
Received in revised form 8 September 2014
Accepted 27 November 2014
Available online 4 December 2014

Keywords:
Flow shop scheduling problem
Agents
Variable neighborhood search
Feasibility model

ABSTRACT

Multi-agent scheduling in flow shop environment is seldom considered. In this paper flow shop scheduling problem with two agents is studied and its feasibility model is considered, in which the goal is to minimize the makespan of the first agent and the total tardiness of the second agent simultaneously under the given upper bounds. A simple variable neighborhood search (VNS) algorithm is proposed, in which a learning neighborhood structure is constructed to produce new solutions and a new principle is applied to decide if the current solution can be replaced with the new one. VNS is tested on a number of instances and the computational results show the promising advantage of VNS when compared to other algorithms of the problem.

1. Introduction

As the extended version of the classical scheduling problem, the scheduling problems with multiple agents have some different features: each agent has his/her own set of jobs and wishes to minimize an objective function that depends on the completion time of his/her jobs; all agents compete on the use of common processing resources. These features lead to a new goal, which is to find a schedule that minimizes a combination of the agents' objective functions or satisfies each agent's requirement for his/her own objective function.

The scheduling problems with multiple agents have attracted much attention in the past decade since the pioneering works of Baker and Smith (2003) and Agnetic, Mirchandani, Pacciarelli, and Pacifici (2004). A large body of literature has discussed the problem in single machine, parallel machines and flow shop environments. Cheng, Ng, and Yuan (2006) study the NP-compete feature of multi-agent single-machine scheduling, in which each agent's objective is to minimize the total weighted number of tardy jobs. Cheng, Ng, and Yuan (2008) discuss the feasibility model and the minimality model of multi-agent scheduling on a single machine. Wu, Huang, and Lee (2011) present a BB and two heuristic algorithms for two-agent single machine scheduling with learning considerations. Cheng, Cheng, Wu, Hsu, and Wu (2011) and Cheng, Chung, Liao, and Lee (2013) develop branch-and-bound (BB) and simulated annealing (SA) algorithm for two-agent single-machine scheduling with learning considerations and two-agent single-machine scheduling with release times, respectively. Lee, Chung, and Hu (2012) propose three genetic algorithms (GA) for two-agent single-machine 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 single-machine scheduling problems with assignable due dates and provide polynomial-time algorithms. Liu, Yi, Zhou, and Hua (2013) discuss the optimal properties of two-agent single machine scheduling with sum-of-processing-times- based deterioration and present some polynomial time algorithms to solve the problem. Wu et al. (2013) propose a BB algorithm and a tabu search (TS) for two-agent single-machine scheduling with deterioration jobs. Feng, Fan, Li, and Shang (2014) give a 2-approximation algorithm and a fully polynomial-time approximation scheme for two-agent scheduling with rejection on a single machine.

With respect to multi-agent parallel machines scheduling, 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. Elvikis, Hamacher, and T'kindt (2011) discuss the problem with two conflicting objectives and provide polynomial time algorithms for solving the problem. 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 (2013) present the fully polynomial time

☆ This manuscript was processed by Area Editor T.C. Edwin Cheng.

* Tel.: +86 15327311013.
E-mail address: deminglei11@163.com

http://dx.doi.org/10.1016/j.cie.2014.11.024
0360-8352/

approximation schemes for two-agent scheduling problems on identical parallel machines.

Some studies have considered multi-agent scheduling in flow shop environments. Lee, Chen, Chen, and Wu (2011) study a two-machine flow shop scheduling problem (FSSP) with two agents where the objective is to minimize the total completion time of first agent with no tardy jobs for the second agent. Lee, Chen, and Wu (2010) consider two-machine FSSP with two agents and develop a BB algorithm and a SA to minimize the total tardiness of the 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 FSSP with two agents. Mor and Mosheiov (2014) propose some polynomial time solution algorithms for proportionate FSSP with two agents.

The previous studies on multi-agent production scheduling have the following features:

(1) Most of papers have been presented for multi-agent scheduling on a single machine. Multi-agent scheduling is not considered fully in many-machines environments. To the best of our knowledge, FSSP with more than two machines and multiple agents is not considered.
(2) The polynomial-time algorithm and mathematical programming methods are main approaches to solve the problem. Only few papers apply some meta-heuristics such as GA, TS and SA.
(3) Most of literature involves multiple objectives; moreover, in most cases, the goal is to optimize an objective of jobs from the first agent given that the objective of jobs from the second agent does not exceed an upper bound. In general, all agents should have the same chance to compete for the common resources and objectives of them should be dealt with equally and optimized simultaneously. It is not fair to only minimize the objective of the first agent.

The main contribution of this study is to consider FSSP with two agents and develop an effective variable neighborhood search (VNS) to minimize simultaneously objectives of two agents under the given upper bounds. VNS has the following features: A learning neighborhood structure is applied to produce new solutions; a new replacement principle is used to decide if the current solution can be replaced with the new one; a simple method is applied to update the non-dominated set. VNS is finally applied to FSSP with two agents and computational results are analyzed and summarized.

The remainder of the paper is organized as follows. Problem under study are described in Section 2. The proposed algorithm for the problem is shown in Section 3. Numerical test experiments on VNS are reported in Section 4 and the conclusions are summarized in the final section and some topics of the future research are provided.

2. Problem description

FSSP is quite common in practice, especially in the process industry. In general, it is assumed that all jobs of FSSP come from the same agent; however, more than one agent provides processing tasks for the same manufacturer in the real-world situation.

FSSP with two agents is composed of n jobs $J_1, J_2, \cdots, J_n$ and m machines $M_1, M_2, \cdots, M_m$. All jobs follow the same route, i.e. they are processed first on machine M_1, and then machine M_2, and so on. Each machine can only process one job at a time. Each job belongs to either the first agent AG_1 or the second agent AG_2. S_1 and S_2 indicate the set of jobs of AG_1 and AG_2 respectively. p_{ij} indicates the processing time of its jth operation o_{ij} of job J_i. C_i^l is the completion time of job J_i from AG_l, $l = 1, 2$. For job J_i of AG_2, its due date is denoted as d_i^2.

The goal of the problem is to decide an appropriate processing sequence of all jobs to minimize the following two objectives simultaneously.

$$\text{Minimize} \quad f_1 = \max_{i \in S_1}\{C_i^1\} \tag{1}$$

$$\text{Minimize} \quad f_2 = \sum_{i \in S_2} \max\{C_i^2 - d_i^2, 0\} \tag{2}$$

Obviously, a schedule of the problem is composed of jobs from two agents and these jobs affect each other. We cannot schedule jobs of each agent independently and should treat each job equally, as a result, the considered problem can be regarded as the traditional FSSP when it is solved using meta-heuristics; moreover, the makespan of the first agent and the total tardiness of the second agent must conflict each other because of the competition between agents for processing resources. These characteristics show that the problem is a multi-objective one in essence. Table 1 shows an illustrative example of the problem, where "_" indicates no due date is considered for jobs of AG_1.

With respect to the problem, the following two models (Cheng et al., 2006, 2008) can be used.

(1) Feasibility model: objectives should satisfies $f_i \leqslant Q_i$, $i = 1, 2, \cdots, T$
(2) Minimality model: the sum of all objectives $\sum_{i=1}^{T} f_i$ should be minimized.

where Q_i is an upper bound of f_i and T is the number of agents, in this study, $T = 2$.

In most of literature on two-agent scheduling, the task is to find a schedule that minimizes the objective of first agent given that the objective of the second agent does not exceed an upper bound. This method has some defects, for example, only the objective of the first agent is minimized and the optimization results are beneficial to the first agent. When two agents should be treated fairly, it is reasonable to use the feasibility model. When the objectives of two agents are simultaneously optimized under the given upper bounds, even if we should give priority to the first agent, many feasible solutions meeting $f_i \leqslant Q_i$ can be obtained, from which the win–win schedule for two agents can be chosen.

Generally, for the problem with minimization of f_1 and f_2, the optimal result is not a single solution but a set of solutions; moreover, the optimal set cannot be obtained without comparing all solutions produced by the method. When solutions in a set are compared each other, take x and y as an example, if $f_i(x) \leqslant f_i(y)$ for $\forall i \in \{1, 2\}$ and $f_i(x) < f_i(y)$ for $\exists i \in \{1, 2\}$, then x dominates y; if

Table 1
An illustrative example.

Job	Operation	p_{ij}	M_k	AG_l	d_i^l
J_1	o_{11}	5	M_1	AG_2	30
	o_{12}	5	M_2		
	o_{13}	7	M_3		
J_2	o_{21}	10	M_1	AG_1	_
	o_{22}	8	M_2		
	o_{23}	2	M_3		
J_3	o_{31}	6	M_1	AG_2	24
	o_{32}	8	M_2		
	o_{33}	9	M_3		
J_4	o_{41}	3	M_1	AG_1	_
	o_{42}	5	M_2		
	o_{43}	4	M_3		