Engineering Applications of Artificial Intelligence 76 (2018) 96–107

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Discrete differential evolution algorithm for distributed blocking flowshop scheduling with makespan criterion

Guanghui Zhang [a,b], Keyi Xing [a,*], Feng Cao [a]

[a] State Key Laboratory for Manufacturing Systems Engineering, and Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, 710049, PR China
[b] School of Sciences, Guizhou Institute of Technology, Guiyang, 550003, PR China

ARTICLE INFO

Keywords:
Scheduling
Distributed blocking flowshop
Makespan
Mathematical model
Discrete differential evolution algorithm

ABSTRACT

This paper deals with a distributed blocking flowshop scheduling problem, which tries to solve the blocking flowshop scheduling in distributed manufacturing environment. The optimization objective is to find a suitable schedule, consisting of assigning jobs to at least two factories and sequencing the jobs assigned to each factory, to make the maximum completion time or makespan minimization. Two different mathematical models are proposed, and in view of the NP-hardness of the problem, a novel hybrid discrete differential evolution (DDE) algorithm is established. First, the problem solution is represented as several job permutations, each of which denotes the partial schedule at a certain factory. Second, four widely applied heuristics are generalized to the distributed environment for providing better initial solutions. Third, both operators of mutation and crossover are redesigned to perform the DDE directly based on the discrete permutations, and a biased section operator is used to increase the diversity of the searching information. Meanwhile, an effective local search based on distributed characteristics and an elitist retain strategy are integrated into the DDE framework to stress both local exploitation and global exploration. Taking into account the time cost, an effective speed-up technique is designed to enhance the algorithmic efficiency. In the experimental section, the parameters of DDE are calibrated by the Taguchi method. Experimental results derived from a wealth of test instances have demonstrated the algorithmic effectiveness, which further concludes that the proposed DDE algorithm is a suitable alternative approach for solving the problem under consideration.

1. Introduction

Production scheduling addresses the assignment of resources, typically machines, to tasks or jobs over time for optimizing a certain objective. The type of a production scheduling problem is determined by several factors: the layout of machines, the flow of jobs on the machines as well as some other production constraints (Pinedo, 2012). Among different types of scheduling problems, the flowshop scheduling is one of the most extensively studied scheduling problems with a strong engineering application. Its production prototypes can be found in a rather wide range of industries (Garey et al., 1976; Gupta and Stafford Jr, 2006; Li et al., 2015; Nawaz et al., 1983; Reza Hejazi and Saghafian, 2005; Ruiz and Maroto, 2005), such as various manufacturing systems, assembly lines and information service facilities. When there is no storage capacity between any two adjoining machines, the classical flowshop scheduling evolves into the blocking flowshop scheduling problem (Grabowski and Pempera, 2000; Hall and Sriskandarajsh, 1996). In this situation, a job, having completed on a machine, has to be blocked on current machine until next machine becomes idle. The applications on such a scheduling problem abound in the production environments, where the buffers amongst machines either do not deploy due to insufficient investment or have to be prohibited to utilize since some special technological requirements.

For the scheduling problems mentioned above, it is assumed that the processing of all jobs is conducted in the same factory, that is to say, in a single production center. Nevertheless, the real-life problems, concurrent or mass production emerged recently, have necessitated the patterns of distributed or multi-factory production. This machine environment is able to allocate the total tasks among some independent production units, which enables enterprises to harvest a lot of potential benefits, including higher product quality, better corporate reputation, lower production costs and manufacturing period (Behnamian and Fatemi Ghomi, 2016). At present, several different types of distributed manufacturing systems have been proposed and surveyed. Among them,

* Correspondence to: The Systems Engineering Institute, Xi'an Jiaotong University, No.28, Xianning West Road, Xi'an City, Shaanxi, 710049, PR China
E-mail addresses: zhangguanghui@stu.xjtu.edu.cn (G.H. Zhang), kyxing@mail.xjtu.edu.cn (K.Y. Xing), caofeng@stu.xjtu.edu.cn (F. Cao).

https://doi.org/10.1016/j.engappai.2018.09.005
Received 2 July 2017; Received in revised form 27 May 2018; Accepted 8 September 2018

there are distributed flowshop system (Naderi and Ruiz, 2010), distributed jobshop system (Hsu et al., 2016; Naderi and Azab, 2014), distributed flexible manufacturing system (Chan et al., 2006b,a), and distributed two-stage assembly system (Xiong et al., 2014).

In this paper, we have tackled a new distributed blocking flowshop scheduling problem (DBFSP) with makespan criterion, which focused on solving the blocking flowshop scheduling problem in a distributed environment. To the authors' knowledge, at present there is only one journal paper (Ying and Lin, 2017) having been published for such a problem. In addition, it can be proved that it is a NP-hard problem since the single factory case with more than two machines is strongly NP-complete (Hall and Sriskandarajsh, 1996). Therefore, it is unrealistic to use the traditional mathematical methods to solve the DBFSP, especially for solving the large-scale problems. In order to obtain the solutions with both quality and efficiency, this paper establishes a novel and effective discrete differential evolution (DDE) algorithm.

Other components of this paper are arranged as follows. Section 2 systemically reviews the relevant literature. Section 3 devotes to illustrate the DBFSP. The details of DDE algorithm are presented in Section 4. Numerical experiments and statistical analyses are provided in Section 5. Finally, conclusions and future research topics are showed in Section 6.

2. Literature review

For several decades now, the typical blocking flowshop scheduling problem has been extensively focused on, and a myriad of solution methods, mainly approximation approaches such as the heuristics (McCormick et al., 1989; Pan and Wang, 2012; Ronconi, 2004; Ronconi and Armentano, 2001; Ronconi and Henriques, 2009) and the meta-heuristics (Caraffa et al., 2001; Grabowski and Pempera, 2007; Liang et al., 2010; Ribas et al., 2011; Tasgetiren et al., 2017; Wang et al., 2010; Wang and Tang, 2012), have been proposed to solve such a problem. For more detailed investigation, interested readers could further refer to the relevant literature.

Compared with the single factory case, the scheduling regarding the distributed blocking flowshops is a more complicated problem. A schedule of it has to first assign jobs to the factories distributed, and then determine the sequence of the jobs assigned at each blocking-flowshop factory. For such a novel problem, the literature published so far is very limited, whereas the literature on other type of the distributed scheduling problems is relatively rich. For instance, Behnamian (2014) addressed a distributed parallel factory system by adopting a decomposition based hybrid VNS-TS algorithm. De Giovanni and Pezzella (2010) used an improved genetic algorithm (GA) to figure out the distributed flexible jobshop scheduling problem. Jia et al. (2003) proposed a GA and further, Jia et al. (2007) combined the Gantt chart and GA for handling the distributed jobshop scheduling problem. Chan et al. (2005) applied an adaptive GA with dominant genes (GADG) for scheduling the medium- and large-scale distributed jobshops. Later, Chan et al. (2006b, a) separately applied GADG to deal with the distributed flexible manufacturing system and a variant of it with the machine maintenance. Xiong et al. (2014) presented three hybrid metaheuristics for a distributed two-stage assembly flowshop scheduling problem.

Naderi and Ruiz (2010) for the first time addressed the distributed permutation flowshop scheduling problem (DPFSP) with infinite buffer capacity, and presented six mathematical models and fourteen heuristic methods. Starting with this pioneering work, different methods have been surveyed, such as the genetic algorithm (Gao and Chen, 2011), tabu search (Gao et al., 2013), estimation of distributed algorithm (Wang et al., 2013), immune algorithm (Xu et al., 2014), scatter search algorithm (Naderi and Ruiz, 2014), iterated greedy algorithm (Fernandez-Viagas and Framinan, 2015; Lin et al., 2013).

Differential evolution (DE) (Storn and Price, 1997) is a well-known and effective metaheuristic method for continuous optimization problems that iteratively performs three operators: mutation, crossover and selection. Due to its simple mechanism, easy implementation and fast convergence, DE algorithm has won more and more attention from both academia and the industry. In recent years, plenty of successful applications were achieved for optimizing continuous problems (Neri and Mininno, 2010; Slowik, 2011; Rakshit et al., 2013; Shen and Wang, 2017). Nevertheless, the continuous nature of the original DE determines that it cannot be directly used to solve the combinatorial optimization with discrete characteristics such as the scheduling problems.

At present, the application of DE algorithm to solve the scheduling problems is mainly through two schemes (Santucci et al., 2016). In most of DE algorithms for the scheduling problems, it needs to take a special transformation technique to encode the scheduling solutions as numerical vectors. The DEs evolve based on these numerical vectors, which are re-decoded as schedules only at the time of evaluation. This type of scheme has been used in DE algorithms reported in literature (Tasgetiren et al., 2006; Onwubolu and Davendra, 2006; Qian et al., 2009) and, more recently, in Liu et al. (2014) and Vincent and Ponnambalam (2013). Another scheme of the DE is referred to as the discrete DE or DDE, where the evolutionary operators are designed directly based on discrete permutation representation. It is able to avoid the transformation operation between real and integer solution representation. Currently, the work on DDE is very little and mainly used for solving the flowshop scheduling problems, like Tasgetiren et al. (2009), Pan et al. (2011), Deng and Gu (2012), Xiong et al. (2015) and Santucci et al. (2016). To the best of our knowledge, there is no research using the DDE to solve the distributed blocking flowshop scheduling problems. In order to solve the DBFSP with makespan minimization directly, this paper has established a novel and effective hybrid DDE algorithm based on the problem characteristics of both discreteness and distribution.

3. Illustration of DBFSP

3.1. Problem description

The blocking flowshop scheduling problem (BFSP) is first described as follows. There is a set $J = \{J_j|(j = 1, 2, \ldots, n)\}$ of n jobs to be processed on a set $M = \{M_i|(i = 1, 2, \ldots, m)\}$ of m machines disposed in a certain order. The processing of job J_j requires a set of m operations $O_j = \{O_{j,i}|(i = 1, 2, \ldots, m)\}$, where $O_{j,i}$ is executed on machine M_i with a processing time $P_{j,i}$. Without intermediate buffer between machines, job J_j, having completed operation $O_{j,i}$, has to remain on M_i until M_{i+1} becomes idle. Both machines and jobs are available at time zero. Each machine handles at most one job and each job is processed on at most one machine at a time. The processing of jobs follows the same order on all machines.

The DBFSP considered herein processes n jobs in J by using a set $F = \{F_k|(k = 1, 2, \ldots, f)\}$ of $f \geq 2$ factories, each of which possesses the same machine environment as the BFSP. Once assigning a job to one factory, the transfer to the other factories is prohibited. A complete schedule of the DBFSP involves two interrelated decisions: assigning jobs to factories and sequencing the jobs assigned at each factory. The final objective is to find a schedule for minimizing the criterion of maximum completion time or makespan.

3.2. Makespan calculation

Let $\pi_k = [J_{\pi_k(1)}, J_{\pi_k(2)}, \ldots, J_{\pi_k(n_k)}]$ denote the partial schedule of n_k jobs assigned to factory F_k. Then, a schedule of the DBFSP with n jobs and f factories can be denoted by a set of the partial schedules at all factories, i.e., $\pi = \{\pi_k|(k = 1, 2, \ldots, f)\}$. In order to calculate its makespan $C_{max}(\pi)$, two mathematical models are given by extending the methods in Ronconi (2004) and Wang et al. (2010) to distributed setting considered in this paper.