Computers & Operations Research 37 (2010) 509 – 520

Contents lists available at ScienceDirect

Computers & Operations Research

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

A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems

Ling Wang[a], Quan-Ke Pan[b,*], P.N. Suganthan[c], Wen-Hong Wang[b], Ya-Min Wang[b]

[a]*Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China*
[b]*College of Computer Science, Liaocheng University, Liaocheng 252059, PR China*
[c]*School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Ave., 639798 Singapore, Singapore*

ARTICLE INFO

Available online 24 December 2008

Keywords:
Blocking flow shop
Differential evolution
Discrete differential evolution
Hybrid algorithm
Local search
Speed-up

ABSTRACT

This paper proposes a novel hybrid discrete differential evolution (HDDE) algorithm for solving blocking flow shop scheduling problems to minimize the maximum completion time (i.e. makespan). Firstly, in the algorithm, the individuals are represented as discrete job permutations, and new mutation and crossover operators are developed for this representation, so that the algorithm can directly work in the discrete domain. Secondly, a local search algorithm based on insert neighborhood structure is embedded in the algorithm to balance the exploration and exploitation by enhancing the local searching ability. In addition, a speed-up method to evaluate insert neighborhood is developed to improve the efficiency of the whole algorithm. Computational simulations and comparisons based on the well-known benchmark instances of Taillard [Benchmarks for basic scheduling problems. European Journal of Operational Research 1993;64:278–285], by treating them as blocking flow shop problem instances with makespan criterion, are provided. It is shown that the proposed HDDE algorithm not only generates better results than the existing tabu search (TS) and TS with multi-moves (TS + M) approaches proposed by Grabowski and Pempera [The permutation flow shop problem with blocking. A tabu search approach 2007;35:302–311], but also outperforms the hybrid differential evolution (HDE) algorithm developed by Qian et al. [An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. Computers and operations research 2009;36(1):209–233] in terms of solution quality, robustness and search efficiency. Ultimately, 112 out of 120 best known solutions provided by Grabowski and Pempera [The permutation flow shop problem with blocking. A tabu search approach 2007;35:302–311] and Ronconi [A branch-and-bound algorithm to minimize the makespan in a flowshop problem with blocking. Annals of Operations Research 2005;138(1):53–65] are further improved by the proposed HDDE algorithm.

1. Introduction

Flow shop scheduling problem is one of the most popular machine scheduling problems with extensive engineering relevance, representing nearly a quarter of manufacturing systems, assembly lines, and information service facilities in use nowadays [1–8]. This paper considers flow shop scheduling problems with blocking constraint. In blocking flow shops, there are no buffers between machines and hence intermediate queues of jobs waiting in the production system for their next operations are not allowed. Therefore, when needed, a job having completed processing on a machine has to remain on this machine and block itself until next machine is available for processing. The blocking flow shop scheduling problems have important applications in the production environment where processed jobs are sometimes kept in the machines because of the lack of intermediate storage [9], or stock is not allowed in some stages of the manufacturing process because of technological requirements [10]. On the other hand, it has been proved that the blocking flow shop scheduling problem to minimize makespan with three machines is NP-hard in the strong sense [11]. Therefore, it is of significance both in theory and in engineering applications to develop effective and efficient novel solution procedures to solve such problems.

However, blocking flow shop scheduling problems have not captured enough research so far [2,12]. Among the research, McCormich et al. [13] developed a constructive heuristic, known as profile fitting (PF), for solving sequencing problems in an assembly line with blocking to minimize cycle time. In their heuristic, the authors created a partial sequence by adding an unscheduled job to obtain the minimum sum of idle times and blocking times on machines. A more comprehensive approach is presented by Leisten [14] for dealing

* Corresponding author. Tel.: +86 635 8238360; fax: +86 635 8238809.
E-mail address: qkpan@lcu.edu.cn (Q.-K. Pan).

0305-0548/$ - see front matter © 2009 Published by Elsevier Ltd.
doi:10.1016/j.cor.2008.12.004

with permutation and non-permutation flow shops with finite and unlimited buffers to maximize the use of buffers and to minimize the machine blocking. However, the author concluded that the heuristic did not produce better solutions than the Nawaz–Enscore–Ham (NEH) [16] heuristic. Ronconi [9] proposed three constructive heuristics, called minmax (MM), combination of MM and NEH (MME), and combination of PF and NEH (PFE), respectively, for blocking flow shop problems with makespan criterion. It was demonstrated by the authors that the MME and PFE heuristics outperformed the NEH algorithm in problems with up to 500 jobs and 20 machines. Based on the connection between no-wait flow shop scheduling problems and flow shop scheduling problems with blocking, Abadi et al. [17] proposed a heuristic for minimizing cycle time in blocking flow shops. Recently, Ronconi and Henriques [12] studied the minimization of the total tardiness in flow shops with blocking scheduling and presented some constructive heuristics. By computational tests, the authors showed that their new approaches were promising for the problems considered.

With the development of computer technology, a few meta-heuristics have been used to solve blocking flow shop scheduling problems. Caraffa [18] developed a genetic algorithmic approach for solving large size restricted slowdown flow shop problems in which blocking flow shop problems were special cases. Ronconi [4] proposed a heuristic algorithm based on branch-and-bound method by using the new lower bounds which exploited the blocking nature and were better than those presented in their earlier paper [15]. By applying some properties of the problems associated with the blocks of jobs as well as using multi-moves to accelerate the convergence to more promising areas of the solutions space, recently, Grabowski and Pempera [2] developed two tabu search (TS) and TS with multi-move (TS+M) approaches. Computational results demonstrated that the TS and TS + M algorithms outperformed both the genetic algorithm [18] and Ronconi's method [4].

The differential evolution (DE) algorithm was first introduced by Storn and Price [19] to optimize complex continuous nonlinear functions. As a population-based evolutionary algorithm, the DE uses simple mutation and crossover operators to generate new candidate solutions, and applies one-to-one competition scheme to greedily decide whether the new candidate or its parent will survive in the next generation. Due to its simplicity, ease of implementation, fast convergence, and robustness, the DE algorithm has gained much attention and a wide range of successful applications such as digital PID controller design [20], feed-forward neural networks training [21,35], digital filter design [22] and earthquake hypocenter location [23]. However, because of its continuous nature, applications of the DE algorithm on scheduling problems are still considerably limited [24–31]. Therefore, in this paper, we propose a hybrid discrete DE (HDDE) algorithm for solving blocking flow shop scheduling problems with makespan criterion. In the proposed DDE algorithm, individuals are represented as discrete job permutations, and novel job-permutation-based mutation and crossover operators are employed to generate new candidate solutions, and an effective insert-neighborhood-based local search is embedded to enhance exploitation. Furthermore, a speed-up method for insert neighborhood structure is developed to reduce computational time requirements. Simulation results and comparisons demonstrate the effectiveness of the proposed HDDE algorithm in solving blocking flow shop scheduling problems with makespan criterion.

This paper is organized as follows. In Section 2, the blocking flow shop scheduling problem is stated and formulated. In Section 3, the speed-up method for the insert neighborhood structure is proposed. In Section 4, the discrete DE (DDE) algorithm is proposed in detail. Section 5 presents a HDDE algorithm after explaining an effective local search. The computational results and comparisons are provided in Section 6. Finally, we conclude the paper with some concluding remarks in Section 7.

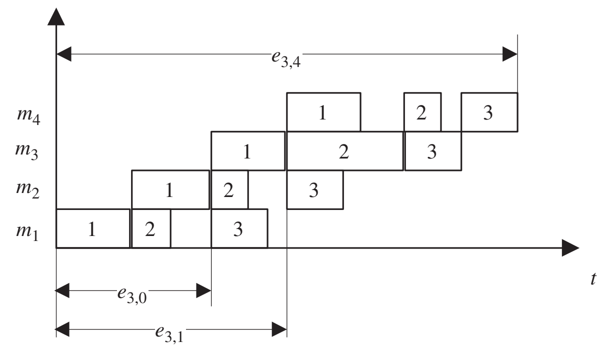

Fig. 1. Computation of $e_{j,k}$.

2. The blocking flow shop scheduling problem

The blocking flow shop scheduling problem can be defined as follows. There are n jobs from the set $J=\{1,2,\ldots,n\}$ and m machines from the set $M=\{1,2,\ldots,m\}$. Each job $j\in J$ will be sequentially processed on machine $1,2,\ldots,m$. Operation $o_{j,k}$ corresponds to the processing of job $j\in J$ on machine k $(k=1,2,\ldots,m)$ during an uninterrupted processing time $p_{j,k}$, where its setup time is included into the processing time. At any time, each machine can process at most one job and each job can be processed on at most one machine. The sequence in which the jobs are to be processed is the same for each machine. Since the flow shop has no intermediate buffers, a job cannot leave a machine until its next machine downstream is free. In other words, the job has to be blocked on its machine if its next machine is not free. The aim is then to find a sequence for processing all jobs on all machines so that its maximum completion time (i.e. makespan) is minimized.

2.1. Mathematical model

Let a job permutation $\pi=\{\pi_1,\pi_2,\ldots,\pi_n\}$ represent the schedule of jobs to be processed, and $e_{j,k}$ be the departure time of operation $o_{j,k}$ (illustrated in Fig. 1). According to the literature [9], $e_{j,k}$ can be calculated as follows

$$e_{1,0}=0, \quad (1)$$

$$e_{1,k}=e_{1,k-1}+p_{1,k}, \quad k=1,\ldots,m-1, \quad (2)$$

$$e_{j,0}=e_{j-1,1}, \quad j=2,\ldots,n, \quad (3)$$

$$e_{j,k}=\max\{e_{j,k-1}+p_{j,k},\ e_{j-1,k+1}\}, \quad j=2,\ldots,n,\ \ k=1,\ldots,m-1, \quad (4)$$

$$e_{j,m}=e_{j,m-1}+p_{j,m}, \quad j=1,\ldots,n, \quad (5)$$

where $e_{j,0}$, $j=1,\ldots,n$, denotes the starting time of job π_j on the first machine. In the above recursion, the departure times of the first job on every machine are calculated first, then the second job, and so on until the last job. Then the makespan of the job permutation $\pi=\{\pi_1,\pi_2,\ldots,\pi_n\}$ is given by $C_{\max}(\pi)=e_{n,m}$, and its computational complexity is $O(mn)$.

Let $f_{j,k}$ be the tail, i.e., duration between the latest loading time of operation $o_{j,k}$ $(k=m,\ldots,1)$ and the end of the operations, and $f_{j,m+1}$ be the duration between the latest completion time of operation $o_{j,m}$ and the end of the operations (illustrated in Fig. 2). To follow the blocking constraints, we can obtain the recursion below

$$f_{n,m+1}=0, \quad (6)$$