Computers & Operations Research 36 (2009) 2498–2511

Contents lists available at ScienceDirect

Computers & Operations Research

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

A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems

Quan-Ke Pan[a], Ling Wang[b,*], Bin Qian[b]

[a]College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[b]Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Available online 29 October 2008

Keywords:
Differential evolution
Discrete differential evolution algorithm
No-wait flow shop
Multi-objective
Insert neighborhood
Speed-up

ABSTRACT

This paper presents a novel discrete differential evolution (DDE) algorithm for solving the no-wait flow shop scheduling problems with makespan and maximum tardiness criteria. First, the individuals in the DDE algorithm are represented as discrete job permutations, and new mutation and crossover operators are developed based on this representation. Second, an elaborate one-to-one selection operator is designed by taking into account the domination status of a trial individual with its counterpart target individual as well as an archive set of the non-dominated solutions found so far. Third, a simple but effective local search algorithm is developed to incorporate into the DDE algorithm to stress the balance between global exploration and local exploitation. In addition, to improve the efficiency of the scheduling algorithm, several speed-up methods are devised to evaluate a job permutation and its whole insert neighborhood as well as to decide the domination status of a solution with the archive set. Computational simulation results based on the well-known benchmarks and statistical performance comparisons are provided. It is shown that the proposed DDE algorithm is superior to a recently published hybrid differential evolution (HDE) algorithm [Qian B, Wang L, Huang DX, Wang WL, Wang X. An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. Computers & Operations Research 2009;36(1):209–33] and the well-known multi-objective genetic local search algorithm (IMMOGLS2) [Ishibuchi H, Yoshida I, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation 2003;7(2):204–23] in terms of searching quality, diversity level, robustness and efficiency. Moreover, the effectiveness of incorporating the local search into the DDE algorithm is also investigated.

1. Introduction

No-wait flow shop scheduling problem is a typical scheduling problem with strong engineering background, which has important applications in different industries including chemical processing, food processing, concrete ware production and pharmaceutical processing [1–6]. A comprehensive survey paper on the research and applications of the no-wait flow shop scheduling problem can be found in Ref. [4].

In a no-wait flow shop scheduling problem, the processing of each job has to be continuous, i.e., once a job is started on the first machine, it must be processed through all machines without any interruption. Thus, when needed, the start of a job on the first machine must be delayed in order to satisfy the no-wait constraint. Given the processing time of each job on each machine, the no-wait flow shop scheduling problem is to find a schedule or a set of schedules so that one or multiple criteria are optimized. For the no-wait flow shop scheduling problems with single objective, it is strongly NP-hard when the number of machines is more than two [7]. Therefore, in the past decades, efforts have been dedicated to obtain high-quality solutions in generally acceptable time and memory requirements by heuristic optimization techniques. These solutions techniques can be found in Refs. [8–12]. To attain a better solution quality, meta-heuristics have grown quickly with the development of computer technology, e.g., genetic algorithm (GA) [13], simulated annealing (SA) algorithm [6,14], hybrid GA and SA algorithm [15], variable neighborhood search (VNS) algorithm [15], descending search algorithm [16], tabu search (TS) algorithm [16], iterated greedy (IG) algorithm [17], hybrid particle swarm optimization (PSO) algorithm [18] and discrete PSO algorithm [19,20].

* Corresponding author. Tel.: +86 1062783125272; fax: +86 1062786911.
E-mail address: wangling@mail.tsinghua.edu.cn (L. Wang).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.10.008

However, most real-life no-wait flow shop scheduling problems naturally involve the optimization of multiple conflicting objectives. Therefore, it is more important to develop multiple objective scheduling technologies and approaches for the no-wait flow shop scheduling problems. Although meta-heuristics has been widely extended to the multi-objective scheduling problems [1–2,21–30] since the late 1980s, only a few papers were focused on scheduling problems with the no-wait requirement. To the best of our knowledge, Aliahverdi and Aldowaisan [31] were the first to develop a hybrid SA algorithm and a hybrid genetic heuristics for no-wait flow shop scheduling problems with makespan and maximum lateness criteria, where a weight sum of both criteria was addressed. Recently, Tavakkoli-Moghaddam et al. [32] presented a hybrid multi-objective immune algorithm to find Pareto-optimal solutions for no-wait flow shop scheduling problems with weighted mean completion time and weighted mean tardiness criteria. With regard to the complexity of the multi-objective flow shop scheduling problems, it was shown in [33] that considering more than one objective did not simplify the scheduling problem, and it was further proved in [34] that multi-objective scheduling problems were as complex as the corresponding single objective problems.

This paper develops a novel differential evolution (DE) algorithm for solving the no-wait flow shop scheduling problems with makespan and maximum tardiness criteria. The DE algorithm is one of the latest evolutionary optimization methods proposed by Storn and Price [35] for complex continuous non-linear functions. In the DE algorithm, simple mutation and crossover operators are used to generate new candidate solutions and a one-to-one competition scheme is applied to greedily determine the new target individuals for next generation. Due to its simplicity, easy implementation and quick convergence, the DE algorithm has gained much attention and a wide range of successful applications [36–41]. However, because of its continuous nature, the applications of the DE algorithm to production scheduling problems are still very limited [40,41]. Therefore, in this paper, a discrete DE (DDE) algorithm is presented. First, individuals in the DDE algorithm are represented as discrete job permutations, and the job-permutation-based mutation and crossover operators are developed to generate candidates. Second, an elaborate one-to-one selection scheme is designed by considering the domination status of a candidate with its counterpart target individual as well as an archive set of the non-dominated solutions found so far. Third, an effective local search algorithm is developed to merge into the DDE algorithm to balance global exploration and local exploitation. Furthermore, several speed-up methods are devised to evaluate a job permutation and its whole insert neighborhood as well as to decide domination status of a solution with the archive set so as to improve the efficiency of the scheduling algorithm. Simulated results and comparisons demonstrate the effectiveness of the proposed DDE algorithm for the no-wait flow shop scheduling problems with makespan and maximum tardiness criteria.

The rest of the paper is organized as follows. In Section 2, the no-wait flow shop scheduling problem with makespan and maximum tardiness criteria is stated and formulated. In Section 3, the DDE algorithm is proposed and described in detail. The computational results and comparisons are provided in Section. 4 Conclusions are presented in Section 5.

2. No-wait flow shop scheduling problem

2.1. No-wait flow shop scheduling problem

The no-wait flow shop scheduling problem can be described as follows: Each of n jobs from set $J = \{1, 2, \ldots, n\}$ will be sequenced through m machines ($k = 1, 2, \ldots, m$). Job $j \in J$ has a sequence of m operations $(o_{j1}, o_{j2}, \ldots, o_{jm})$ and a given due date $d(j)$. Operation o_{jk} corresponds to the processing of job j on machine k during an uninterrupted processing time $p(j,k)$. At any time, each machine can process at most one job and each job can be processed on at most one machine. To follow the no-wait restriction, the completion time of the operation o_{jk} must be equal to the earliest start time of the operation $o_{j,k+1}$ for $k=1, 2, \ldots, m-1$. In other words, there must be no waiting time between the processing of any consecutive operations of each of n jobs. The problem is, then, to find a schedule such that the processing order of jobs is the same on each machine and the given criteria are optimized.

In this paper, two objectives to be minimized are considered. The first one is the maximum completion time or makespan. Suppose that a job permutation $\pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ represents a schedule of jobs to be processed. Let $e(\pi_{j-1}, \pi_j)$ be the minimum delay on the first machine between the start of jobs π_{j-1} and π_j restricted by the no-wait constraint when the job π_j is directly processed after the job π_{j-1}. Then the completion time $C(\pi_j, m)$ of job π_j on machine m can be computed by the following formula [16]:

$$C(\pi_1, m) = \sum_{k=1}^{m} p(\pi_1, k) \tag{1}$$

$$C(\pi_j, m) = \sum_{i=2}^{j} e(\pi_{i-1}, \pi_i) + \sum_{k=1}^{m} p(\pi_j, k), \quad j = 2, 3, ..., n \tag{2}$$

And $e(\pi_{j-1}, \pi_j)$ can be computed as follows [16]:

$$e(\pi_{j-1}, \pi_j) = F_{j-1,j}(\pi_j, m) - \sum_{k=1}^{m} p(\pi_j, k) \quad \text{for } j = 2, .., n \tag{3}$$

where $F_{j-1,j}(\pi_j, m)$ denotes the makespan of jobs π_{j-1} and π_j in the 2-machine permutation flow shop scheduling problem. So, the makespan of the schedule $\pi = \{\pi_1, \pi_2, ..., \pi_n\}$ is given as follows:

$$C_{\max}(\pi) = C(\pi_n, m) \tag{4}$$

The second objective is the maximum tardiness, which is given as follows:

$$T_{\max}(\pi) = \max_{j=1}^{n}(\max(0, C(\pi_j, m) - d(\pi_j))) \tag{5}$$

As an example, consider a 3-job and 3-machine problem with a processing time matrix

$$(p(j,k))_{3\times 3} = \begin{bmatrix} 2 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{bmatrix}$$

and a due date matrix

$$(d(j))_{1\times 3} = \begin{bmatrix} 7 \\ 9 \\ 10 \end{bmatrix}$$

Suppose that a schedule is $\pi=\{1, 2, 3\}$, and its Gantt chart is drawn in Fig. 1. According to Eqs. (3)–(5), the minimum delay on the first machine, makespan and maximum tardiness are given as follows: $e(1,2) = 3$; $e(2,3) = 3$; $C_{\max}(\pi) = 12$ and $T_{\max}(\pi) = 2$.

2.2. Short cut to evaluate a job permutation

For the no-wait flow shop scheduling problem, since $\sum_{k=1}^{m} p(\pi_j, k)$ and $e(\pi_{j-1}, \pi_j)$ can be computed in advance and be used in the evaluation of a job permutation, the time complexity of Eq. (4) is $O(n)$