Swarm and Evolutionary Computation 32 (2017) 121–131

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem

Jin Deng, Ling Wang*

Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 12 November 2015
Received in revised form
15 May 2016
Accepted 7 June 2016
Available online 8 June 2016

Keywords:
Distributed permutation flow-shop scheduling
Makespan
Total tardiness
Multi-objective optimization
Competitive memetic algorithm

ABSTRACT

In this paper, a competitive memetic algorithm (CMA) is proposed to solve the multi-objective distributed permutation flow-shop scheduling problem (MODPFSP) with the makespan and total tardiness criteria. Two populations corresponding to two different objectives are employed in the CMA. Some objective-specific operators are designed for each population, and a special interaction mechanism between two populations is designed. Moreover, a competition mechanism is proposed to adaptively adjust the selection rates of the operators, and some knowledge-based local search operators are developed to enhance the exploitation ability of the CMA. In addition, the influence of the parameters on the performance of the CMA is investigated by using the Taguchi method of design-of-experiment. Finally, extensive computational tests and comparisons are carried out to demonstrate the effectiveness of the CMA in solving the MODPFSP.

1. Introduction

The permutation flow-shop scheduling is an attractive research problem [1–5] with wide applications in practice [6–8]. For the classical permutation flow-shop scheduling problem (PFSP), all jobs are to be processed in a single factory. With the development of the cooperative production of multi-plants, the distributed scheduling is helpful to the resource utilization, production efficiency and profit.

Under the distributed environment, Naderi and Ruiz [9] extended the PFSP to the distributed PFSP (DPFSP). Compared to the PFSP, the literature about the DPFSP is very scant. As for the heuristic methods, Naderi and Ruiz [9] developed two factory assignment rules and a variable neighborhood descent (VND) algorithm. Gao and Chen [10] proposed a modified NEH [11] algorithm. Based on the solutions generated by heuristics, iterative search algorithms were used to further improve the solutions. Lin et al. [12] presented a modified iterative greedy (IG) algorithm with a new acceptance criterion and a variable number of removed elements. Fernandez-Viagas and Framinan [13] proposed a bounded search IG algorithm which utilized the lower bound to reduce the search space. To obtain higher quality solutions, some researchers focused on designing the meta-heuristics for solving the DPFSP. Liu and Gao [14] presented a discrete electromagnetism-like mechanism algorithm and a variable neighborhood search (VNS) to execute local search. Gao et al. [15] proposed a knowledge-based genetic algorithm (GA) which utilized the information of the elite individuals to guide the search process. Wang and Wang [16] proposed a earliest completion factory (ECF) rule as the decoding method, and developed an estimation of distribution algorithm (EDA) with multiple local search strategies. Xu et al. [17] presented a hybrid immune algorithm for the DPFSP.

In the above studies, it considered only one scheduling objective (i.e., makespan). However, in a practical situation, decision-makers may care about multiple objectives simultaneously. For example, the tardiness criterion is very important in production systems, because penalty costs will incur if jobs cannot be completed before the due dates. For the PFSP with both the makespan and tardiness criteria, much work has been done. To minimize the makespan of the PFSP where the maximum tardiness is limited by a given upper bound, Framinan and Leisten [18] developed a heuristic combining the proposed property and NEH. Fernandez-Viagas and Framinan [19] proposed a constructive heuristic based on a bounded insertion and a Tabu search, and then presented a non-population-based algorithm to improve the heuristic solution by means of a bounded relative local search. To obtain the Pareto set of the PFSP with makespan and maximum tardiness criteria, Qian et al. [20] proposed a multi-objective different evolution combining with a VNS based local search. To minimize makespan and total tardiness of the PFSP, Guo et al. [21] presented a discrete

* Corresponding author. Present/permanent address: Department of Automation, Tsinghua University, Beijing 100084, China.
E-mail address: wangling@mail.tsinghua.edu.cn (L. Wang).

http://dx.doi.org/10.1016/j.swevo.2016.06.002
2210-6502/

particle swarm optimization (DPSO) algorithm.

In this paper, the multi-objective DPFSP (MODPFSP) with the makespan and total tardiness criteria will be studied. Since the PFSP with makespan or total tardiness minimization is NP-hard [22], the MODPFSP is also NP-hard. As the problem scale increases, it is impractical to solve the problem with the exact methods. Moreover, in the DPFSP two sub-problems need to be solved, that is, assigning factory for each job and determining the job sequence in each factory. Because the two sub-problems are coupled and cannot be solved sequentially, the methods for the multi-objective PFSP may not be effective for the MODPFSP. Therefore, it is significant to propose effective and efficient algorithms for solving the MODPFSP.

During recent years, memetic algorithms have been shown to be powerful in solving the complex optimization problems [23,24]. The memetic algorithm (MA) combines the evolutionary search with the problem-specific local search to balance the exploration and exploitation reasonably. For neural network design, Liu et al. [25] proposed a PSO-based MA by combining the PSO and several faster training methods. For multi-compartment vehicle routing problem, El-Fallahi et al. [26] proposed a MA with a post-optimization phase based on path relinking. For graph coloring problem, Lü and Hao [27] proposed a MA with an adaptive multi-parent crossover and a distance-and-quality based replacement criterion. To find the minimum energy broadcast tree in wireless ad hoc networks, Arivudainambi and Rekha [28] proposed a MA with a knowledge mutation and the r-shrink procedure [29]. To select services of a workflow, Ludwig [30] proposed a GA-based MA and a PSO-based MA, combining the PSO and GA with the Munkres algorithm [31]. Inspired by the successful applications of the MA, in this paper we propose a novel MA framework named competitive memetic algorithm (CMA) for the MODPFSP. The CMA consists of four key phases: initialization, competition, local intensification, and interaction. To solve the MODPFSP, firstly, a simple encoding scheme is proposed to represent the factory assignment and the job processing sequence in each factory. Secondly, two populations are initialized, focusing on different objectives. In each population, competition is carried out among multiple search operators specific to the corresponding objective, and some knowledge-based local search operators are developed to enhance the exploitation. Thirdly, to balance two objectives in each population, some individuals will be exchanged between the two populations in the interaction phase. Besides, the Pareto-based approach is employed to deal with the two conflicting objectives. Extensive computational results and comparisons show that the CMA is effective in solving the MODPFSP.

The rest of the paper is organized as follows: In Section 2, the MODPFSP is described. In Section 3, the framework of the CMA is introduced and then the CMA for solving the MODPFSP is presented in detail. In Section 4, some performance metrics are introduced and the influence of parameter setting is investigated. Computational results and comparisons are given in Section 5. Finally, the paper ends with some conclusions and future work in Section 6.

2. Multi-objective distributed permutation flow-shop scheduling problem

2.1. Basic concepts of multi-objective optimization

Generally, a multi-objective optimization problem can be described as follows:

$$\text{Minimize } y = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})\right) \quad (1)$$

where $x \in R^p$ is a p dimension decision vector, and $y \in R^q$ is the objective vector with q objectives.

Different from single objective optimization, multiple conflicting criteria need to be considered simultaneously in multi-objective optimization and cardinality of the optimal set is usually more than one. There are many approaches to handle the multiple objectives [32,33], including the weighted sum approach, goal programming, the ε-constraint method, and the Pareto optimum approach.

(1) Weighted sum approach. It converts the multi-objective optimization into a scalar optimization problem by adding all the objective functions together using different weighting coefficients as follows.

$$\text{Minimize } f'(x) = \sum_{r=1}^{q} \omega_r f_r(x) \quad (2)$$

$$\sum_{r=1}^{q} \omega_r = 1 \quad (3)$$

where $f'(x)$ is the weighted objective, and $\omega_r(r = 1, 2, \ldots, q)$ are the weighting coefficients.

(2) Goal programming. The expected value for each objective is incorporated into the problem as the additional constraint. Then, the problem is converted to minimize the absolute deviations between the objective values and the expected values as follows.

$$\text{Minimize } f'(x) = \sum_{r=1}^{q} \left|f_r(x) - E_r\right| \quad (4)$$

where E_r is the expected value of the objective.

(3) The ε-constraint method. It converts the multi-objective optimization into the following constraint optimization by minimizing one primary objective and considering the others as constraints. Each constraint is bounded by an allowable upper limit ε_r.

$$\text{Minimize } f_{r*}(x) \quad (5)$$

$$\text{subject to: } f_r(x) \le \varepsilon_r, \quad \text{for } r = 1, 2, \ldots, \quad q, r \ne r* \quad (6)$$

(4) Pareto optimum approach. Rather than finding a single solution, it aims at obtaining a set of Pareto optimal solutions so that decision maker can choose a final plan according to actual demand.

A solution a is considered to dominate solution b (denoted as $a \succ b$) if and only if: (a) $f_r(a) \le f_r(b) \quad \forall r \in \{1, 2 \ldots, q\}$, and (b) $f_r(a) < f_r(b) \; \exists r \in \{1, 2 \ldots, q\}$. A solution X^* is called Pareto optimal if it is not dominated by any other solution.

In this paper, the Pareto optimum approach is employed to solve the MODPFSP, and the following concepts will be used.

Optimal Pareto set: The solution set containing all optimal Pareto solutions is defined as the optimal Pareto set.

Non-dominated solution set/Archive set (AS): The solution set containing all non-dominated solutions that obtained by a certain algorithm is defined as non-dominated solution set/archive set.

Optimal Pareto front: The optimal Pareto front (in the objective space) is formed by the objective vectors corresponding to the solutions in the optimal Pareto set.

Non-dominated sorting: Non-dominated sorting is used to rank the solutions into several levels according to their dominance degrees. Solutions in the first frontier are the Pareto optimal solutions, and the second frontier contains the solutions that are dominated by the solutions in the first frontier only, and so on.