Advances in Engineering Software 77 (2014) 35–47

Contents lists available at ScienceDirect

Advances in Engineering Software

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

An effective hybrid teaching–learning-based optimization algorithm for permutation flow shop scheduling problem

Zhanpeng Xie, Chaoyong Zhang *, Xinyu Shao, Wenwen Lin, Haiping Zhu

State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science & Technology (HUST), Wuhan, Hubei 430074, PR China

ARTICLE INFO

Article history:
Received 19 March 2014
Received in revised form 27 June 2014
Accepted 13 July 2014
Available online 24 August 2014

Keywords:
Permutation flow shop scheduling problem
Teaching-learning-based optimization algorithm
Variable neighborhood search
Simulated annealing
Makespan
Maximum lateness

ABSTRACT

Permutation flow shop scheduling (PFSP) is among the most studied scheduling settings. In this paper, a hybrid Teaching–Learning-Based Optimization algorithm (HTLBO), which combines a novel teaching–learning-based optimization algorithm for solution evolution and a variable neighborhood search (VNS) for fast solution improvement, is proposed for PFSP to determine the job sequence with minimization of makespan criterion and minimization of maximum lateness criterion, respectively. To convert the individual to the job permutation, a largest order value (LOV) rule is utilized. Furthermore, a simulated annealing (SA) is adopted as the local search method of VNS after the shaking procedure. Experimental comparisons over public PFSP test instances with other competitive algorithms show the effectiveness of the proposed algorithm. For the DMU problems, 19 new upper bounds are obtained for the instances with makespan criterion and 88 new upper bounds are obtained for the instances with maximum lateness criterion.

1. Introduction

In permutation flow shop scheduling problems, n jobs $N = N_1, N_2, \ldots, N_n$ have to be processed on a set of m machines $M = M_1, M_2, \ldots, M_m$ sequentially. Therefore, each job consists of a set of m operations $O_j = \{O_{j1}, \ldots, O_{jm}\}$. Each operation has a given processing time denoted by $P_{i,j}$ ($i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$). At any time, each machine can process at most one job and each job can be processed by at most one machine. Once the processing of a job on a machine has started, it must be completed without interruption. The sequence in which the jobs to be processed are identical for each machine. Thus there is $n!$ possible processing sequences for the problem. The minimum completion time, which is known as makespan or $C_{\max}$, is the most commonly studied objective of PFSP [1]. Recently, PFSP with other objectives such as those involving due dates have drawn significant attention [2–4]. Demirkol et al. [5] presented extensive sets of randomly generated test problems for the problems of minimizing makespan ($C_{\max}$) and maximum lateness ($L_{\max}$) in flow shops, generally referred to DMU problems. PFSP with the makespan criterion can be denoted as $F_m|prmu|C_{max}$ and has been proved NP-complete [6]. PFSP with the criterion of maximum lateness can be denoted as $F_m|prmu|L_{\max}$, where $L_j = \max\{C_j - d_j, 0\}$ is the lateness of job j, being d_j its due date and C_j its completion time at the last machine of the shop. Lenstra et al. [7] proved that the two-machine flow shop with maximum lateness is NP-complete.

Approaches for PFSP can be divided into three categories: exact algorithms, heuristics and meta-heuristics. Exact algorithms, such as branch-and-bound method, dynamic programming and mathematical programming, have been successfully applied in solving small instances [8–10]. However, they could not obtain promising results in a reasonable time for medium or large instances. As for the heuristics, a feasible solution is generally built based on some constructive operations with a fast process, while the solution is quite not satisfactory [11]. More recently, the meta-heuristic algorithms, such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), have been given special emphasis for they could provide high-quality solutions with reasonable computing times. In recent decade, an increasing number of research papers focusing on meta-heuristics for PFSP have been published.

Teaching–Learning-Based Optimization (TLBO) proposed by Rao et al. [12] is a novel efficient optimization method. The main idea behind TLBO is the simulation of a classical school learning process. The advantages of TLBO algorithm such as ease of implementation, immediately accessible for practical applications, speed to get the solutions and robustness are shown in the literature [12,13]. TLBO seems to be a rising star from amongst a number of meta-heuristics with relatively competitive performances. Empirical tests show that TLBO could outperforms the other

* Corresponding author. Tel.: +86 13429882087.
E-mail address: zcyhust@mail.hust.edu.cn (C. Zhang).

http://dx.doi.org/10.1016/j.advengsoft.2014.07.006
0965-9978/

well-known meta-heuristics regarding constrained benchmark functions, constrained mechanical design, and continuous non-linear numerical optimization problems [13]. However, applications of TLBO for discrete combinatorial optimization problems are still limited.

In this paper, a novel hybrid Teaching–Learning-Based Optimization algorithm (HTLBO) is proposed for PFSP to optimize two objectives: the makespan and maximum lateness of jobs. The paper is organized as follows: Section 2 presents the literature review about PFSP problem; Section 3 provides the description of the PFSP; Section 4 describes implementation details of the HTLBO for PFSP; Section 5 shows the computational results and comparisons with other competitive algorithms; and Section 6 concludes the paper.

2. Literature review

The makespan criterion for flow shop scheduling is still a hot topic of research as shown in the recent review by Gupta and Stafford Jr. [14]. Tseng et al. [15] introduced that the exact algorithms are sensible to the number of machines. As a result, a wealth of literature on heuristic and meta-heuristic methods for the PFSP problem and makespan criterion was published. Heuristics for the makespan minimization problem have been proposed by Palmer [16], Campbell et al. [17], Dannenbring [18], Nawaz et al. [19], Taillard [20], Framinan and Leisten [21] and Framinan et al. [22], Li et al. [23], Laha and Chakraborty [24]. Among these existing heuristics, the Nawaz–Enscore–Ham (NEH) heuristic has been proved one of the most successful constructive heuristics which can obtain comparable results against some modern constructive methods according to the results of the computational evaluation given by Ruiz and Maroto [25]. The meta-heuristics include simulated annealing [26,27], tabu search [28–31], genetic algorithms [32,33], ant colony optimization [34], iterated local search [35], iterated greedy methods [36,37], particle swarm optimization [38–40], differential evolution algorithm [41] and so on. Based on these meta-heuristics, some hybrid algorithms were proposed, which had been demonstrated effective according to the computational results on some well-known benchmark problems. Chang et al. [42] proposed a hybrid genetic-immune algorithm, in which the regular genetic algorithm is applied in the first stage to rapidly evolve and when the processes are converged up to a predefined iteration then artificial immune system is introduced to hybridize GA in the second stage. Among hybrid algorithms, the most popular strategy is to hybridize a meta-heuristic with a local search method. Murata et al. [43] showed two hybrid algorithms: genetic local search and genetic simulated annealing. Nearchou [44] designed a hybrid simulated annealing algorithm for solving the flow shop scheduling problem. In this algorithm, an iterated hill climbing procedure is applied on the population of schedules during the annealing process. Pan et al. [41] introduced a new and novel referenced local search procedure hybridized with both discrete differential evolution algorithm and iterated greedy algorithm to further improve the solution quality. Ahmadizar [45] developed a new ant colony optimization algorithm for makespan minimization in permutation flow shops. In this algorithm, novel mechanism is employed in initialization the pheromone trails based on an initial sequence. Tzeng et al. [46] proposed a hybrid estimation of distribution algorithm (EDA) with ant colony system (ACS). Their algorithm, in each iteration, applies a new filter strategy and a local search method to update the local best solution and, based on the local best solution, generates pheromone trails using a new pheromone-generating rule and applies a solution construction method of ACS to generate members for the next iteration.

For the maximum lateness criterion, within our knowledge, only a few of researchers adopted this criterion as the performance measure of proposed algorithms. Some researchers studied on the two-machine flow shop scheduling with maximum lateness criterion [47]. Tasgetiren et al. [38] first introduced a particle swarm optimization for maximum lateness minimization in permutation flow shop scheduling problem based on the DMU benchmark problems. Since then, some novel meta-heuristics have been proposed to deal with the objective of maximum lateness. Zheng and Yamashiro [11] designed a new quantum differential evolutionary algorithm, this algorithm based on the basic quantum-inspired evolutionary algorithm (QEA). Li and Yin [48] suggested an opposition-based differential evolution algorithm to solve PFSP with the criteria of makespan and maximum lateness.

3. Description of the permutation flow shop problem

Schedule π is a permutation of the n jobs, which can be denoted as $\{\pi_1, \pi_2, \ldots, \pi_n\}$, in which $\pi_i \in \Omega$ is the ith ($i = 1, 2, \ldots, n$) job in π. Π is the set of all the permutations of the n jobs. Let $p_{\pi_i,j}$ represents the processing time of job π_i on machine j and $C(\pi_i, m)$ represents the completion time of job π_i on machine m. Then the completion time for the n-job, m-machine problem can be calculated as follows:

$$C(\pi_1, 1) = p_{\pi_1}, 1 \tag{1}$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + p_{\pi_i,1}, \; i = 2, \ldots, n \tag{2}$$

$$C(\pi_1, j) = C(\pi_1, j-1) + p_{\pi_1,j}, \; j = 2, \ldots, m \tag{3}$$

$$C(\pi_i, j) = \max(C(\pi_{i-1}, j), C(\pi_i, j-1)) + p_{\pi_i,j}, \; i = 2, \ldots, n, \; j = 2, \ldots, m \tag{4}$$

The makespan of a permutation π can be formally defined as the completion time π_n of the last job on the last machine m, so the makespan is defined as:

$$C_{\max}(\pi) = C(\pi_n, m) \tag{5}$$

The PFSP with the makespan criterion is to find the optimal permutation π^* in the set of all permutation:

$$C_{\max}(\pi^*) \le C(\pi_n, m), \; \forall \pi \in \Pi \tag{6}$$

As for the flow shop scheduling with the due date constraint, let $L(\pi_i)$ denoted the lateness of jobs π_i and can be defined as:

$$L(\pi_i) = C(\pi_i, m) - d(\pi_i) \tag{7}$$

Maximum lateness $L_{\max}(\pi)$ of a permutation can be defined as:

$$L_{\max}(\pi) = \max(C(\pi_i, m) - d(\pi_i)) \tag{8}$$

where $d(\pi_i)$ is the due date of jobs π_i. The optimal solution π^* should satisfy the following criterion:

$$L_{\max}(\pi^*) \le L_{\max}(\pi), \; \forall \pi \in \Pi \tag{9}$$

4. Hybrid teaching–learning-based optimization algorithm for PFSP

4.1. Brief introduction of TLBO

Teaching–Learning-Based Optimization (TLBO) is a population-based method inspired by the effect of the influence of a teacher on the output of learners in a class and has been applied to cluster data [49], design of planar steel frames [50], optimization of two stage thermoelectric cooler [51], job shop scheduling [52] and so on. Like other nature-inspired algorithms, TLBO uses a population