Information Sciences 316 (2015) 487–502

Contents lists available at ScienceDirect

Information Sciences

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

Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm ☆

Jun-qing Li [a,b], Quan-ke Pan [c,*]

[a] College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[b] State Key Laboratory of Synthetic Automation for Process Industries, Northeastern University, ShenYang 110819, PR China
[c] State Key Lab of Digital Manufacturing Equipment & Technology, Huazhong University of Science & Technology, Wuhan 430074, PR China

ARTICLE INFO

Article history:
Received 19 November 2013
Received in revised form 18 September 2014
Accepted 3 October 2014
Available online 13 October 2014

Keywords:
Hybrid flow shop scheduling problem
Artificial bee colony
Tabu search
Limited buffer

ABSTRACT

This paper presents a novel hybrid algorithm (TABC) that combines the artificial bee colony (ABC) and tabu search (TS) to solve the hybrid flow shop (HFS) scheduling problem with limited buffers. The objective is to minimize the maximum completion time. Unlike the original ABC algorithm, in TABC, each food source is represented by a string of job numbers. A novel decoding method is embedded to tackle the limited buffer constraints in the schedules generated. Four neighborhood structures are embedded to balance the exploitation and exploration abilities of the algorithm. A TS-based self-adaptive neighborhood strategy is adopted to impart to the TABC algorithm a learning ability for producing neighboring solutions in different promising regions. Furthermore, a well-designed TS-based local search is developed to enhance the search ability of the employed bees and onlookers. Moreover, the effect of parameter setting is investigated by using the Taguchi method of design of experiment (DOE) to determine the suitable values for key parameters. The proposed TABC algorithm is tested on sets of instances with large scales that are generated based on realistic production. Through a detailed analysis of the experimental results, the highly effective and efficient performance of the proposed TABC algorithm is contrasted with the performance of several algorithms reported in the literature.

1. Introduction

In recent years, the classical flow shop scheduling problem (FSSP) has been proven to play an important role in modern manufacturing and production systems. The hybrid flow shop (HFS) scheduling problem is one branch of the FSSP and has been verified to be an NP-hard problem [11]. Recent and comprehensive reviews on the HFS can be found in [33,36]. Many heuristics for solving the HFS have been developed. Of these heuristic or meta-heuristic algorithms, genetic algorithms (GAs) are the most popular. Portmann et al. introduced an enhanced version of the B&B algorithm crossed with a GA [32]. Jin et al. solved a scheduling problem involving a real printed circuit board manufacturing system using a GA method [13]. Oguz and Ercan developed a GA approach for the HFS with multiprocessor tasks [28]. Ruiz and Maroto developed a GA for the HFS with

☆ This research is partially supported by National Science Foundation of China 61174187, 51435009, 61374187, and 61104179, Program for New Century Excellent Talents in University (NCET-13-0106), Specialized Research Fund for the Doctoral Program of Higher Education (20130042110035), Science Foundation of Liaoning Province in China (2013020016), Basic Scientific Research Foundation of Northeast University under Grant N110208001 and N130508001, Starting Foundation of Northeast University under Grant 29321006, and IAPI Fundamental Research Funds (2013ZCX02).

* Corresponding author.
E-mail addresses: lijunqing.cn@gmail.com (J.-q. Li), panquanke@mail.neu.edu.cn (Q.-k. Pan).

http://dx.doi.org/10.1016/j.ins.2014.10.009
0020-0255/© 2014 Elsevier Inc. All rights reserved.

sequence-dependent setup times and machine eligibility [34]. Engin et al. developed an efficient GA for the HFS with multiprocessor tasks [6]. Other strategies have also been introduced for solving the HFS. Lin and Liao solved the two-stage HFS problem with setup time and dedicated machines [23]. Babayan and He presented an agent-based approach for a three-stage HFS with identical parallel machines [2]. Ying and Lin solved the problem by using ant colony optimization (ACO) [48]. Zandieh et al. applied artificial immune systems (AIS) for the HFS with sequence-dependent setup times [49]. Janiak et al. applied three constructive algorithms and three meta-heuristics based on the tabu search (TS) and simulated annealing (SA) algorithms [12]. More recently, Kahraman et al. investigated a parallel greedy algorithm for solving the multistage HFS [14]. Liao et al. proposed an approach using particle swarm optimization (PSO) and bottleneck heuristics for the problem [22]. The realistic HFS is a much more complex generalization of the traditional HFS. Ruiz et al. solved realistic HFS problems with skipped stages, sequence-dependent setup times, machine lags, release dates, machine eligibility and precedence relationships [35]. Zandieh and Gholami solved the HFS with stochastic machine breakdown by using an immune algorithm (IA) [50]. Dugardin et al. focused on the multi-objective resolution of a re-entrant HFS [5]. Recently, Xuan and Li investigated the batch decomposition strategy with a mixed backward and forward dynamic programming algorithm [46]. Pan et al. proposed an efficient artificial bee colony (ABC) algorithm for solving the steelmaking problem by considering the assignment of different penalty coefficients for three objectives, i.e., the average sojourn time and the earliness/tardiness penalty [31]. Indeed, the literature reveals that the HFS has become increasingly important in realistic production systems.

The majority of the literature makes the assumption that there are sufficient intermediate buffers between consecutive stages. However, in realistic scheduling problems, a finite intermediate buffer (limited buffer) always exists, which blocks and delays operation on the previous machine or the intermediate buffer. Several heuristic and meta-heuristic algorithms that solve the flow shop scheduling problem with limited buffers have been reported. Brucker et al. modeled the problem as a disjunctive graph and solved it using a TS algorithm [4]. Wardono and Fathi also used the TS algorithm for solving the multi-stage parallel machine problem with limited buffer capacities [45]. Wang et al. proposed a hybrid GA for a permutation flow shop scheduling problem with limited buffers [42]. Grabowski and Pempera introduced several problem-specific neighborhood structures for the problem [10]. Liu et al. developed a hybrid PSO for the same problem [25]. Furthermore, Pan et al. investigated a hybrid discrete differential evolution (DE) algorithm [30]. Lin and Ying designed a hybrid algorithm based on the features of artificial immune systems and the annealing process of simulated annealing algorithms [24].

The HFS with limited buffers (hereafter denoted the HFS-LB) is a typical scheduling problem with a strong industrial background that can be found to exist in many different industries, particularly the steel industry. However, fewer studies have been dedicated to most realistic HFS-LB problems than to the flow shop problem with limited buffers. Sawik presented a mixed-integer programming approach for makespan minimization in flexible flow lines with limited buffers [37]. Tang and Xuan designed a Lagrangian relaxation algorithm for real-time hybrid flow shop scheduling with finite intermediate buffers [40]. Wang and Tang solved the HFS with finite intermediate buffers by a hybrid algorithm combining the TS and scatter search algorithms [44]. It should be noted that the three studies discussed above considered HFS-LB problems with identical parallel machines. In a realistic industrial system, such as that associated with steelmaking casting production, there are unrelated machines with different processing abilities at each stage. Yaurima et al. solved the HFS with unrelated machines, sequence-dependent setup times, availability constraints, and limited buffers with an improved GA [47]. An HFS-LB with at most six stages was considered in [47]. However, with respect to real-world applications, more research should be conducted on large-scale HFS-LB scheduling problems. Therefore, in this study, we develop a novel hybrid algorithm combining the ABC and TS (TABC) algorithms to solve the large-scale HFS-LB scheduling problem with unrelated machines.

To the best of our knowledge, there are no previous studies that have applied the ABC algorithm to solve the HFS with limited buffers. The main features of the proposed TABC algorithm are as follows: (1) four neighborhood structures are embedded; (2) a TS-based self-adaptive neighborhood structure strategy is employed to balance the exploitation and exploration capabilities; (3) the TS-based local search heuristic is used by both employed bees and onlookers; and (4) a decoding strategy considering the limited buffer constraints is developed. The rest of this paper is organized as follows: Section 2 briefly describes the formulation of the problem. Then, the related algorithms are presented in Section 3, and Section 4 presents the two TS-based heuristics. The proposed TABC algorithm is discussed in detail in Section 5, and Section 6 presents the experimental results as well as a comparison between the proposed TABC algorithm and the best-performing algorithms reported in the literature to demonstrate the superiority of the former. Finally, Section 7 provides concluding remarks and discusses future research directions.

2. Problem descriptions

In an HFS with limited buffers, there are n jobs, m machines, s stages, and h buffer capacities between consecutive stages. Let $S = \{S_i\}_{1 \leqslant i \leqslant s}$ be the series of stages, $M = \{M_k\}_{1 \leqslant k \leqslant m}$ be the set of machines, $J = \{J_j\}_{1 \leqslant j \leqslant n}$ be the set of jobs, and $B = \{B_q\}_{1 \leqslant q \leqslant h}$ be the set of limited buffers. Let $p_{i,j,k}$ be the processing time of job j on machine k at stage i ($p_{i,j,k} \geqslant 0$), $s_{i,j}$ be the starting time of job j in stage i, and $c_{i,j}$ be the completion time of job j at stage i.

The main assumptions and constraints for the considered problem are as follows:

- In stage i, there are m_i unrelated parallel machines with different processing abilities, where $m_i \geqslant 1$.
- Between the stages i and $i + 1$, there are b_i buffer capacities, where $b_i \geqslant 0$.