Journal of Cleaner Production 181 (2018) 584–598

Contents lists available at ScienceDirect

Journal of Cleaner Production

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

Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions

Jun-qing Li [a, b, c, *], Hong-yan Sang [b, **], Yu-yan Han [b], Cun-gang Wang [b], Kai-zhou Gao [b]

[a] School of Information Science and Engineering, Shandong Normal University, 250014, PR China
[b] College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[c] State Key Laboratory of Synthetic Automation for Process Industries, Northeastern University, Shenyang 110819, PR China

ARTICLE INFO

Article history:
Received 15 June 2017
Received in revised form
31 January 2018
Accepted 1 February 2018
Available online 16 February 2018

Keywords:
Hybrid flow shop scheduling problem
Multi-objective optimization
Setup energy consumption
Energy-aware

ABSTRACT

This paper proposes an energy-aware multi-objective optimization algorithm (EA-MOA) for solving the hybrid flow shop (HFS) scheduling problem with consideration of the setup energy consumptions. Two objectives, namely, the minimization of the makespan and the energy consumptions, are considered simultaneously. In the proposed algorithm, first, each solution is represented by two vectors: the machine assignment priority vector and the scheduling vector. Second, four types of decoding approaches are investigated to consider both objectives. Third, two efficient crossover operators, namely, Single-point Pareto-based crossover (SPBC) and Two-point Pareto-based crossover (TPBC) are developed to utilize the parent solutions from the Pareto archive set. Then, considering the problem structure, eight neighborhood structures and an adaptive neighborhood selection method are designed. In addition, a right-shifting procedure is utilized to decrease the processing duration for all machines, thereby improving the energy consumption objective of the given solution. Furthermore, several deep-exploitation and deep-exploration strategies are developed to balance the global and local search abilities. Finally, the proposed algorithm is tested on sets of well-known benchmark instances. Through the analysis of the experimental results, the highly effective proposed EA-MOA algorithm is compared with several efficient algorithms from the literature.

1. Introduction

The HFS scheduling problem is one generalization of the classical flow shop scheduling problem (FSSP), which has been verified to be a Non-deterministic Polynomial-time hard (NP-hard) problem (Ruiz and Vázquez Rodríguez, 2010; Ribas et al., 2010). In an HFS problem, two types of tasks should be considered simultaneously: assigning machines for each job and scheduling each job on each assigned machine. Therefore, the HFS problem is harder than the classical FSSP due to the additional consideration of parallel device selection for each job. Many published papers have discussed solving the HFS problem with many different algorithms. We can classify these algorithms by the number of stages in the considered problems. There are three types of problems: two-stage, three-stage, and m-stage. The two-stage problem is the HFS problem with two consecutive stages, while the m-stage problem has a series of m stages. Gupta (1988) studied the HFS problem with two stages where there is only one device in the second stage. Lin and Liao (2003) investigated the same problem with setup time and dedicated machines. Riane et al. (1998) developed an efficient heuristic for minimizing the makespan in a three-stage HFS problem. Carlier and Neron (2000) proposed an exact algorithm for solving the multi-processor flow shop. The benchmark problems that they generated were used in many studies as test problems.

The HFS with m stages is closer to the production reality. Therefore, it has been the focus of more research. Exact algorithms were first applied to solve the m-stage HFS problem, such as the Lagrange method (Chang and Liao, 1994) and the B&B algorithm

* Corresponding author. School of information science and engineering, Shandong Normal University, 250014, PR China.
** Corresponding author.
E-mail addresses: lijunqing@lcu-cs.com (J.-q. Li), sanghongyan@lcu-cs.com (H.-y. Sang).

https://doi.org/10.1016/j.jclepro.2018.02.004
0959-6526/

(Portmann et al., 1998). However, exact algorithms have limited ability to solve HFS problems with large scales. During recent years, heuristic and meta-heuristic algorithms have been developed to solve HFS problems, including genetic algorithm (GA) (Oguz and Ercan, 2005; Engin et al., 2011), artificial bee colony (ABC) (Li and Pan, 2015; Li et al., 2016a,b; Pan, 2016), iterated greedy (IG) (Ying et al., 2014), cuckoo search algorithm (CSA) (Marichelvam et al., 2014a,b), parallel tabu search algorithm (PTSA) (Bożejko et al., 2013), particle swarm optimization (PSO) (Liao et al., 2012; Chou, 2013), estimation of distribution algorithm (EDA) (Wang et al., 2015), bi-layer optimization approach (BLO) (Jiang et al., 2015), artificial immune system (AIS) (Chung and Liao, 2013), local search method (Lei and Guo, 2016), ant colony optimization (ACO) (Qin et al., 2015), bat algorithm (Marichelvam et al., 2013), and fruit-fly optimization algorithm (FOA) (Li et al., 2016a,b). Very recently, some hybrid meta-heuristics have also been designed to solve HFS problems, such as a hybrid of GA and TS (Sukkerd and Wuttipornpun, 2016), a hybrid of ABC algorithm and several heuristics (Pan et al., 2014), a hybrid of ABC and TS (Li and Pan, 2015), a combination of GA and imperialist competitive algorithms (ICA) (Moradinasab et al., 2013), and a hybrid of variable neighborhood search (VNS) algorithms (Li et al., 2014a,b). Some meta-heuristics have better global search abilities, while others have better local search abilities. Therefore, well-designed hybrid algorithms can always obtain better performances than single algorithms. However, most of the current literature about HFS problems has not considered machine differences in terms of power consumption capabilities.

In recent years, multi-objective optimization algorithms have been considered and studied in many fields (Deb et al., 2002; Deb and Jain, 2014; Zhang and Li, 2007; Marichelvam et al., 2014a,b; Wang and Liu, 2014; Huang et al., 2015; Tran and Ng, 2013; Shahvari and Logendran, 2016; Pan et al., 2011). Several multi-objective optimization algorithms have been proposed, such as NSGA-II (Deb et al., 2002), NSGA-III (Deb and Jain, 2014), and MOEA/D (Zhang and Li, 2007). Most of the published multi-objective algorithms have been investigated to solve continuous optimization problems. There is less literature on solving multi-objective HFS problems. Marichelvam et al. (2014a,b) proposed a discrete firefly algorithm to solve the HFS problem considering two objectives, i.e., makespan and the mean flow time. Wang and Liu (2014) investigated the HFS problem with minimization of the unavailability of the first stage machine and the makespan. Huang et al. (2015) developed a subgroup PSO approach for solving multi-objective two-stage HFS problems. Tran and Ng (2013) presented a hybrid water flow algorithm for this problem considering the minimization of the makespan and the total tardiness. Shahvari and Logendran (2016) presented a TS-based algorithm for the minimization of two objectives simultaneously, i.e., the weighted sum of the total weighted completion time and the total weighted tardiness. It can be concluded from the above analysis that there is less literature in which the multi-objective features in HFS problems are considered, especially with the consideration of the energy efficiency characteristics.

Nowadays, energy efficient algorithms are being investigated by increasing numbers of researchers (Gahm et al., 2016; Che et al., 2016). Zhang et al. (2014) utilized a time-indexed integer programming formulation to minimize the electricity cost and the carbon footprint under time-of-use tariffs in flow shop environments. For the permutation flow shop problems, Ding et al. (2016a,b) designed a multi-objective NEH algorithm (MONEH), where NEH is short for Nawaz et al. (1983), and a modified multi-objective iterated greedy (MMOIG) algorithm to minimize the total energy consumption and the makespan. For parallel machine scheduling problems, Ding et al. (2016a,b) proposed a time-interval-based mixed integer linear programming formulation to minimize the total electricity cost. Zhang and Chiong (2016) investigated an enhanced local search for minimizing the total weighted tardiness and the total energy consumption in job shop horizons. Luo et al. (2013) developed a hybrid algorithm based on the ant colony optimization method to solve the HFS problems considering the electric power cost (EPC) in the presence of time-of-use (TOU) electricity prices. Dai et al. (2013) presented a genetic simulated annealing algorithm for making a significant trade-off between the makespan and the total energy consumption in flexible flow shop horizons. For the same problem, Tang et al. (2016) utilized an improved particle swarm optimization method. Lu et al. (2017) considered two objectives namely the makespan and the energy consumption in permutation flow shop scheduling problem. There is less literature on minimization of both the makespan and the energy consumption in HFS problems, and there is no published literature in which the setup energy consumption is considered.

In realistic HFS environments, some stages contain multiple devices with different processing capabilities. In addition, each machine usually contains two states, i.e., the working state and the standby state. In each state, the machine will consume different volumes of energy. Furthermore, the setup energy consumption should be considered because it is significant in practice. The main reason for considering the setup energy consumption is that, setup energy consumption may occur when the setup operation is performed to clear the previous job from the certain container, for example, some types of iron in a torpedo. Different pairs of jobs may require different energy consumptions for the setup or clearing procedure. Therefore, in this study, we consider energy efficiency in HFS problems and minimize the energy consumptions and makespan. The rest of this paper is organized as follows: Section 2 gives the problem description. Then, the proposed algorithm is presented in Section 3. Section 4 reports the experimental results and compares them with those of other algorithms in the literature to evaluate the performance of the proposed algorithm. Finally, the last section presents the conclusions of our work.

2. Problem description

2.1. Notations and constraints

In an HFS problem, there are n tasks to be processed on m devices in a predefined order. All tasks and devices are available at time zero. Pre-emption is not allowed, that is, no task can be interrupted before the completion of its current operation. Setup times and setup energy consumptions are considered. Problem data are deterministic and known in advance. There are unlimited intermediate buffers between successive stages. The objective of an HFS problem is to schedule each task on each device such that the makespan and energy consumptions are minimized. The notations that are used in this paper are summarized below:

- Indices

i	index of jobs, $i = 1,2 \ldots,n.$
k	index of machines, $k = 1,2, \ldots,m.$
j	index of stages, $j = 1,2, \ldots, s.$