Applied Soft Computing 61 (2017) 122–128

Contents lists available at ScienceDirect

Applied Soft Computing

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

Hybrid flow shop scheduling with assembly operations and key objectives: A novel neighborhood search

Deming Lei [a,b,*], Youlian Zheng [c]

[a] School of Automation, Wuhan University of Technology, Wuhan 430070, China
[b] State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[c] Faculty of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

ARTICLE INFO

Article history:
Received 15 June 2016
Received in revised form 28 April 2017
Accepted 29 July 2017

Keywords:
Hybrid flow shop scheduling
Neighborhood search
Assembly operation
Key objective

ABSTRACT

This paper addresses hybrid flow shop scheduling problem (HFSP) with assembly operations, in which parts of each product are produced in a hybrid flow shop and then assembled at an assembly stage. The goal is to minimize total tardiness, maximum tardiness and makespan simultaneously. Tardiness objectives are regarded as key ones because of their relative importance and this situation is seldom considered. A simple strategy is applied to handle the optimization with key objectives. A novel neighborhood search with global exchange (NSG) is proposed, in which a part-based coding method is adopted and global exchange is cooperated with neighborhood search to produce high quality solution. Extensive experiments are conducted and the results show that the strategy on key objectives is reasonable and effective and NSG is a very competitive method for the considered HFSP.

1. Introduction

Hybrid flow shops extensively exist in many traditional industries including paper, pharmaceutical, textile, chemical industries and many modern industries such as semi-conductor wafer fabrication. In hybrid flow shop, at least one stage has more than one parallel machine. The redundancy of machines results in some improvements such as introducing flexibility, increasing capacities and avoiding bottle- neck. Hybrid flow shop scheduling problem (HFSP) is an extension of flow shop scheduling and NP- hard one. It has been considered fully in the past fifty years and a number of results on HFSP with various conditions such as multiple objectives and assembly are obtained.

Regarding multi-objective HFSP, Ruiz and Vázquez-Rodríguez [1] pointed out that it is necessary to focus on HFSP with multiple objectives. Jungwattanakit et al. [2] proposed some heuristics and a genetic algorithm (GA) for HFSP with unrelated machines, setup time and dual criteria. Behnamian et al. [3] provided a three-phase hybrid method for minimizing makespan and the sum of the earliness and tardiness of jobs. The study by Naderi et al. [4] aimed to solve HFSP with sequence-dependent setup times, transportation times and two objectives using an improved simulated annealing (SA). Rashidi et al. [5] proposed an improved hybrid parallel GA. Cho et al. [6] reported a parallel GA with four different versions of local search strategies to solve reentrant HFSP with the minimization of total tardiness and makespan. A multi-phase GA is proposed by Karimi et al. [7] for bi-objective hybrid flexible flowshop group scheduling problem. Tran and Ng [8] applied a hybrid water flow algorithm for HFSP with limited buffers and multiple objectives. Mousavi et al. [9] presented a bi-objective local search algorithm with three phases. Bozorgirad and Logendran [10] applied four efficient methods based on tabu search (TS) for group scheduling problem in hybrid flow shop where parallel machines in one or more stages are unrelated. Wang and Liu [11] proposed a multi-objective tabu search (MOTS) for a two-stage HFSP with preventive maintenance. Su et al. [12] used a distributed coevolutionary algorithm to solve a bi-objective HFSP. Lei [13] reported a two-phase neighborhood search algorithm for bi-objective HFSP. Lei and Guo [14] introduced a shuffled frog leaping algorithm for HFSP with two agents and the sum of two objectives. Bozorgirad and Logendran [15] compared local search with population-based algorithms on HFSP with realistic characteristics and the linear combination of two objectives. Karimi and Davoupour [16] provided a colonial competitive algorithm for bi-objective HFSP.

HFSP with assembly operations also has attract some attention, in which all parts of each product are first processed in hybrid flow shop and then are assembled into the final product at the assembly stage. This problem is common in real-world situation and exists

* Corresponding author at: Luoshi Road 122, School of Automation, Wuhan University of Technology, Wuhan 430070, China.
E-mail address: deminglei11@163.com (D. Lei).

http://dx.doi.org/10.1016/j.asoc.2017.07.058
1568-4946/

in many industries including furniture production [17].Yokoyama [18] discussed lower bound and the method getting lower bound. Fattahi et al. [19] provided a hierarchical branch and bound algorithm for HFSP with setup time and assembly operations. Nikzad et al. [17] proposed a hybrid imperialist competitive algorithm to solve two-stage assembly-type HFSP. Komaki et al. [20] provided some methods based on artificial immune system for two-stage HFSP with assembly operations.

It can be seen from the previous works on multi-objective HFSP that two or three objectives are frequently optimized but their relative importance is seldom considered. In fact, it is essential to handle the relative importance of objectives in many cases. For example, in make-to-order (MTO) production systems, on-time delivery is the main focus of the manufacturers and all their efforts are to complete processing before due date, so the minimization of tardiness objectives is more vital than that of make- span and tardiness objectives should be optimized as key ones to meet on-time delivery requirement of customers better. On the other hand, the literature on HFSP with assembly operations is mainly about single objective problem. The unavoidable conflicting among objectives makes it be necessary to solve multi-objective HFSP with assembly operations.

In this study HFSP with assembly operations and key objectives is addressed, the goal of which is to minimize total tardiness, maximum tardiness and makespan under the condition that two tardiness objectives are chosen as key ones. A simple strategy is used to cope with key objectives by neglecting the improvement of makespan in one case. A new algorithm called NSG is constructed based on a part-based coding method. Global exchange and neighborhood structures are cooperated to obtain high quality solutions. NSG is finally tested on a number of instances and computational results show the effectiveness of the strategy on key objectives and the advantages of NSG.

The remainder of the paper is organized as follows. Problem is described in Section 2. NSG for the considered HFSP is shown in Section 3. Computational experiments are reported in Section 4 and the conclusions are summarized and some topics of the future research are provided in the final section.

2. Problem descriptions and discussions on key objectives

HFSP with assembly operations and key objectives is described as follows. There are n products P_1, P_2, $\cdots$, P_n. Each product is obtained by assembling a set of different parts in terms of bill-of-material. N_h is the set of the required parts of product P_h, $h = 1, 2, \cdots, n.w = \sum_{h=1}^{n} |N_h|$ indicates the total number of parts. The assembly time of P_h is denoted by A_h. C_h and d_h respectively represent the completion time of processing and assembly, and the due date of product P_h.

For each product, its parts are first processed in a hybrid flow shop, then they are assembled into product at an assembly stage.

In the hybrid flow shop, part is processed according to the same production flow: stage 1, stage 2, . . ., stage m. Each processing stage k has $|M_k| \geq 1$ unrelated parallel machines, $|M_k| > 1$ for at least one stage, where M_k is the set of all parallel machines of stage $k.p_{ijk}$ indicates the processing time of part i on unrelated machine j at stage k, which is equal to η_{ik}/v_{ijk}. Machine j at stage k can process part i at the relative speed v_{ijk}. η_{ik} is the standard processing time of part i at stage k. A part can skip any stages provided it is processed in at least one stage. When a part i skips stage k, the processing of part i is not done at stage k and p_{ijk} must be equal to 0.

At the assembly stage, there is only one assembly machine and parts are assembled into products on this machine. The assembly operation cannot be started for a product until the processing of its all parts are finished in the hybrid flow shop.

The following constraints are considered. All machines and parts are available from time zero. Each part can only be processed on one machine at a time and each machine cannot process more than one part at a time. Preemption is not allowed in processing and assembly and buffer size is not limited. The assembly machine can only assemble one product at a time et al.

The goal of the problem is to obtain an appropriate machine assignment and processing sequences of all parts and assemble parts into products to optimize the following three objectives.

$$Minimize f_1 = \max_{h=1,2,\cdots,n} \left\{ C_h \right\} \tag{1}$$

$$Minimize f_2 = \sum_{h=1}^{n} \max \left\{ C_h - d_h, 0 \right\} \tag{2}$$

$$Minimize f_3 = \max_{h=1,2,\cdots,n} \left\{ \max \left\{ C_h - d_h, 0 \right\} \right\} \tag{3}$$

where objective f_1 is makespan, f_2 and f_3 are total tardiness and maximum tardiness.

For the problem with f_1, f_2, f_3, the optimal result is not a single solution but a set of solutions; moreover, the optimal set cannot be obtained without comparing all solutions. When solutions in a set are compared each other, take x and y as an example, if $f_i(x) \leq f_i(y)$ for $\forall i \in \left\{1, 2, 3\right\}$, $f_i(x) < f_i(y)$ for $\exists i \in \left\{1, 2, 3\right\}$, then x dominates y; if a solution x cannot be dominated by any other solutions in the same set, x is non-dominated solution regarding the set. If a solution isn't dominated by other solutions in search space, the solution is Pareto optimal. Pareto front is composed of the objective vectors of all Pareto optimal solutions.

In the above definition of Pareto dominance, three objectives have the same importance. When a decision-maker or a MTO manufacturer think tardiness objectives should have higher priority or be more important than makespan, tardiness objectives are regarded as key ones and makespan is a non- key one to reflect the differences of objectives.

The weighted method and the lexicographical approach are often applied to deal with the problem with key objectives. In these methods, the higher priority of key objectives is reflected by lowering the importance of non-key ones or neglecting the improvement of a solution on non-key objectives. The weighted method always provides less weights for non-key ones [21].The lexicographical approach often compares solutions according to key objectives at first (Dhouib et al. [22]; Lei and Guo [23]) and a solution with better non-key objectives is not accepted because of the worse key objectives.

Obviously, the generated solutions of these methods often locate on very narrow region of Pareto front and it is required to decide weights of objectives. In this study, a simple strategy is used to deal with the higher priority of key objectives in the optimization procedure.

3. NSG for HFSP with assembly operations and key objectives

Neighborhood search algorithms including variable neighborhood search (VNS), SA and TS have been proposed and applied to solve various production scheduling problems [4,10,11,24]. In these methods, some effective strategies such as variable neighborhood mechanism are adopted to result in good local search ability; however, the intensification of global exploration ability is seldom studied. In this paper, the cooperation of neighborhood search and global search is considered to improve global exploration ability.

3.1. Part-based coding and decoding

For HFSP, machine assignment is seldom represented like scheduling sub-problem. Machines are often assigned to jobs