European Journal of Operational Research 240 (2015) 338–354

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Scheduling a hybrid assembly-differentiation flowshop to minimize total flow time

Fuli Xiong [a,b], Keyi Xing [b,*], Feng Wang [b]

[a] School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, Shanxi 710055, China
[b] State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an, Shanxi 710049, China

ARTICLE INFO

Article history:
Received 19 March 2014
Accepted 1 July 2014
Available online 9 July 2014

Keywords:
Scheduling
Hybrid assembly-differentiation flowshop
Hybrid meta-heuristics
Total flow time

ABSTRACT

This study considers a hybrid assembly-differentiation flowshop scheduling problem (HADFSP), in which there are three production stages, including components manufacturing, assembly, and differentiation. All the components of a job are processed on different machines at the first stage. Subsequently, they are assembled together on a common single machine at the second stage. At the third stage, each job of a particular type is processed on a dedicated machine. The objective is to find a job schedule to minimize total flow time (TFT). At first, a mixed integer programming (MIP) model is formulated and then some properties of the optimal solution are presented. Since the NP-hardness of the problem, two fast heuristics (SPT-based heuristic and NEH-based heuristic) and three hybrid meta-heuristics (HGA-VNS, HDDE-VNS and HEDA-VNS) are developed for solving medium- and large-size problems. In order to evaluate the performances of the proposed algorithms, a lower bound for the HADFSP with TFT criteria (HADFSP-TFT) is established. The MIP model and the proposed algorithms are compared on randomly generated problems. Computational results show the effectiveness of the MIP model and the proposed algorithms. The computational analysis indicates that, in average, the HDDE-VNS performs better and more robustly than the other two meta-heuristics, whereas the NEH heuristic consume little time and could reach reasonable solutions.

1. Introduction

In a two-stage assembly flowshop, there is a finite set J of n jobs to be processed. Each job requires m different components which are processed on m parallel machines at the first stage. At the second stage, a single assembly machine is used to assemble the components which are produced at the first stage. It has widely industrial applications such as fire engine assembly plant (Lee, Cheng, & Lin, 1993), distributed database systems (Allahverdi & Al-Anzi, 2006a, 2006b) and personal computer manufacturing (Potts, Sevastjanov, Strusevich, Wassenhove, & Zwaneveld, 1995).

The two-stage assembly flowshop scheduling problem (TSAFSP) with makespan criteria has been proved to be strongly NP-hard by Lee et al. (1993) and Potts et al. (1995), respectively. Some recent efforts are made to schedule jobs with different objectives in the two stage assembly production environment. Lee et al. (1993), Potts et al. (1995), Koulamas and Kyparisis (2001), and Allahverdi and Al-Anzi (2006a, 2006b) considered the TSAFSP to minimize makespan. Tozkapan, Kirca, and Chung (2003) studied the problem with objective of minimizing the total weighted flow time. Allahverdi and Al-Anzi (2006a, 2006b) addressed the problem to minimize lateness. Al-Anzi and Allahverdi (2007) also studied the TSAFSP with lateness criteria considering setup times. Mozdgir, Fatemi Ghomi, Jolai, and Navaei (2013) discussed the problem to optimize makespan and mean flow time simultaneously.

However, all the above research neglected the differentiation operations, which arise from various industrial applications (Herrmann & Lee, 1992; Riane, Artiba, & Elmaghraby, 2002; Cheng, Lin, & Tian, 2009; Lin & Hwang, 2011; Liu, Fang, & Lin, 2012; Wang & Liu, 2013). For example, in a PC manufacturing plant, components are assembled on a common stage station, no matter which kind of computers is to be processed. After the assembly stage, the computers will be transported to the differentiation stage where different types of computers are packaged on several package machines. Another example is car manufacturing (Wang & Liu, 2013), when a car model is finished assembly, it will be sent to the paint shop where several parallel paint machines are installed for painting different colors. In the aspect of differentiation flowshop scheduling problem (DFSP), Herrmann and Lee (1992) show that the DFSP is NP-hard even if there are only two

* Corresponding author.
E-mail addresses: xiongfl1974@163.com (F. Xiong), kyxing@sei.xjtu.edu.cn (K. Xing), wangf@mail.xjtu.edu.cn (F. Wang).

http://dx.doi.org/10.1016/j.ejor.2014.07.004
0377-2217/

dedicated machines at stage two. Riane et al. (2002) studied the DFSP in which there are two dedicated machines in the second stage. Its objective is to minimize the makespan. They proved that the problem is strongly NP-complete and presented three heuristics and a dynamic programming algorithm. Cheng et al. (2009) considered the DFSP to minimize the weighted sum of machine completion times (WMT). They proved that the problem is strongly NP-hard and gave an $O(n^3)$ polynomial time algorithm to solve the special case where the sequences of jobs per type are fixed. Also, they developed an approximation algorithm with a tight performance ratio for the general case. Lin and Hwang (2011) presented a dynamic programming algorithm for the DFSP with fixed sequences per job type. Liu et al. (2012) proposed a branch and bound algorithm for DFSP to minimize the makespan. Computational results show that their algorithm can substantially reduce the computing efforts to find optimal solutions. Wang and Liu (2013) considered the same problem with makespan criteria. They proposed a heuristic method based on branch and bound algorithm and given some lower bounds, upper bounds and dominance properties. Experimental results showed the effectiveness of their algorithm. Although there are some research works on the DFSP as above, the process of components manufacturing for assembly operation is neglected. In this situation, comprehensively considering the DFSP and the TSAFSP, we address a novel three-stage production scheduling problem which we called the hybrid assembly-differentiation flowshop scheduling problem (HADFSP). At the first stage, m different components required by a job are processed on m different machines in parallel. When all of the components are completed, a common single machine at the second stage assembles the components together into a model. Subsequently, the model is processed on one of several dedicated machines at the last stage. The first two stages (stage 1 and stage 2) and the last two stages (stage 2 and stage 3) can be regarded as a two-stage assembly flowshop (TSAF) and a differentiation flowshop, respectively. Our objective is to find an optimal or near-optimal permutation schedule for the HADFSP to minimize the total flow time (HADFSP-TFT). To the best of our knowledge, there are no published papers for dealing with this problem.

Obviously, the HADFSP-TFT is strongly NP-hard since its special cases (TSAFSP (Allahverdi & Al-Anzi, 2009), DFSP (Cheng et al., 2009), and three machine flowshop scheduling problem (Pinedo, 2002) are all strongly NP-hard). It is unlikely to find a polynomial time algorithm to obtain the optimal solution for the HADFSP-TFT. Recently, many meta-heuristics have been developed to provide optimal or near-optimal solutions for NP-hard problems in a reasonable execution time. The meta-heuristics include simulated annealing algorithm (SA, Eglese, 1990), genetic algorithm (GA, Holland, 1975), differential evolution (DE, (Storn & Price, 1997)), estimation of distribution algorithm (EDA, Muhlenbein & Paass, 1996), ant colony optimization (ACO, Dorigo & Gambardella, 1997), iterated greedy algorithm (IGA, Ruiz & Stutzle, 2007, 2008), variable neighborhood search (VNS, Mladenovic & Hansen, 1997) and tabu search (TS, Glover, 1996), etc. As an NP-hard problem, it is difficult to know the optimal solution of HADFSP-TFT and hence, to verify the effectiveness and preference of a meta-heuristic algorithm against others meta-heuristics if there are no other meta-heuristics for comparison. Since a skilled combination of different meta-heuristics can improve the performances of many combinatorial problems (Allahverdi & Aydilek, 2014; Chen, Pan, & Lin, 2008; Figielska, 2014; Li, Ong, & Nee, 2004; Murata, Ishibuchi, & Tanaka, 1996; Naderi & Ruiz, 2014; Tseng & Lin, 2009; Wang, Wang, Xu, Zhou, & Liu, 2012), we proposed three hybrid meta-heuristics (HGA-VNS, HEDA-VNS and HDDE-VNS) and compared them with each other for this novel scheduling problem.

The remainder of this paper is organized as follows. Section 2 describes and formulates the HADFSP-TFT under consideration. Subsequently, some properties of the optimal solutions are given. Two constructive heuristics (a SPT-based heuristic and a NEH-based heuristic), three hybrid meta-heuristics (HGA-VNS, HEDA-VNS and HDDE-VNS) and a lower bound for the problem are presented in Section 3. Section 4 shows computational results for the problem. Section 5 concludes the paper and gives suggestions for future research.

2. Problem definition and formulation

The problem studied here can be defined as follows. There are n jobs to be processed. They can be divided into g disjoint sets of jobs $N_1 = \{J_1, J_2, \ldots, J_{n_1}\}, N_2 = \{J_{n_1+1}, J_{n_1+2}, \ldots, J_{n_1+n_2}\}, \ldots,$ and $N_g = \{J_{n_1+n_2+\cdots+n_{g-1}+1}, J_{n_1+n_2+\cdots+n_{g-1}+2}, \ldots, J_{n_1+n_2+\cdots+n_{g-1}+n_g}\}$. Each set consists of a specific type of jobs and N_h contains n_h type h jobs. Let $N = N_1 \cup N_2 \cup \cdots \cup N_g$ be the set of all jobs. All jobs must be processed in three stages. Each job J_j has m components to be processed on m different machines in parallel at the first stage. When all of these m components are completed, they are assembled into Job J_j on the same machine at the second stage. At the third stage, there are g dedicated machines $M_{3,1}, M_{3,2}, \ldots, M_{3,g}$. If a job J_j belongs to type h, then it will be processed on machine $M_{3,h}$.

In this paper, we only discuss permutation schedules. That is, job components are to be processed on each machine at the first stage in the same order. Under the same order, jobs are assembled at the second stage while the job sequence on the dedicated machine at the third stage can be determined by the first-come, first-served (FCFS) rule.

Several assumptions are done as follows.

- All machines are available at time zero.
- Each machine can process at most one job at a time.
- Each job can be processed on at most one machine at a time.
- Setup times and transportation times are neglected.
- The processing time on both machines in three stages are known constants.
- Job processing cannot be preempted before it is finished.
- There are unlimited buffers between the machines of the stage one and two and the stage two and three.

Fig. 1 shows a simple HADFSP example with $m = 3$ and $g = 2$. At the first stage, three different components required by a job are processed on $M_{1,1}$, $M_{1,2}$, and $M_{1,3}$, respectively. At the second stage, all of three components are assembled into a job on the assembly machine. At the last stage, there are two dedicated machines $M_{3,1}$ and $M_{3,2}$, each of which processes a special type job.

Suppose that there are $n = 4$ jobs $J_1, J_2, J_3,$ and J_4 to be processed. Jobs J_1 and J_2 belong to type 1 whereas Jobs J_3 and J_4 belong to type 2. Table 1 lists the processing time of jobs in all three stages. For a given a permutation schedule $S = (J_1, J_4, J_2, J_3)$, its Gantt chart is shown in Fig. 2.

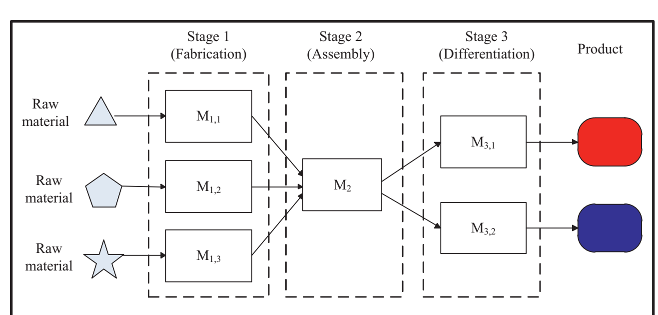

Fig. 1. An example of the HADFSP ($m = 3$, $g = 2$).