Computers & Operations Research 40 (2013) 1064–1075

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

# A genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem

Shijin Wang*, Ming Liu*

*Department of Management Science and Engineering, School of Economics & Management, Tongji University, Shanghai 200092, China*

ARTICLE INFO

Available online 30 October 2012

*Keywords:*
Two-stage hybrid flow shop
No-wait
Genetic algorithm
Heuristics

ABSTRACT

Considering the practical application and the computational complexity of the two-stage no-wait hybrid flow shop scheduling problem, this paper proposes a genetic algorithm (GA). Based on the description of the problem and its properties, some constructive heuristics are first proposed to obtain the upper bound. Then the implementation details of the proposed GA are illustrated, in which the results of heuristics are employed into the initial population. Next, a preliminary computational test with factorial design is conducted to tune the key parameters of four versions of the proposed genetic algorithms resulting from combinations of different crossover and mutation operators. With the tuned parameters, the performance of the proposed genetic algorithms is evaluated in terms of the mean percentage deviation of the solution with respect to the lower bound value, through an extensive computational experiment. The results with different problem configurations demonstrate the effectiveness and efficiency of the proposed genetic algorithm and also demonstrate that the GA performs relatively better when the LOX (two-point linear order crossover) operator and the swap mutation operator are used.



## 1. Introduction

Hybrid flow shop (HFS) scheduling problems combine the properties of flow-shop scheduling problems and parallel machine scheduling problems [1]. It differs from regular flow shop in that at least one stage has more than one machine (with characteristics of identical, uniform or unrelated), which introduces the additional flexibility to the production process. The hybrid flow shop scheduling problems are quite common in practice and are the adequate model for several industrial settings, including electronics, paper, and textile [2]. Due to its important theoretical and practical interest, the HFS scheduling problem has attracted considerable attention. Various problems and variants in two-stage, three-stage or multiple-stage HFS environment are studied. Linn and Zhang [3] provided earlier surveys of this topic, and Ruiz and Vazquez-Rodriguez [2] and Ribas et al. [4] gave the recent comprehensive reviews.

A special but common case of HFS is the two-stage hybrid flow shop, which is shown to be NP-hard even in the case where there are only two machines in one stage and there is only a single machine in the other stage [5]. Hoogeveen et al. [6] showed that the preemptive case is also NP-hard. Due to its theoretical complexity, various methods are designed to solve the problem, including heuristic or approximation methods, and exact methods like branch and bound (B&B). Guinet et al. [7] developed a sequence-first, allocate-second heuristic approach for the two-stage hybrid flow shop problem with the objective of minimizing makespan, in which each stage has multiple parallel identical machines. Gupta and Tunc [8, 9] studied the case in which the first stage contains a single machine, and the second stage contains several identical machines, with the objective of minimizing makespan and minimizing total number of tardy jobs, respectively. Several heuristic methods are proposed in each paper. Gupta et al. [10] studied the reverse case of [8] with respect to makespan minimization. A B&B method has been designed with lower bounds to obtain optimal solution for small scale problem up to 20 jobs and several constructive heuristics are developed for large scale problem up to 250 jobs. Soewandi and Elmaghraby [11] studied a two-stage hybrid flow shop scheduling problem with uniform machines to minimize makespan. Three heuristic procedures have been proposed with comparison to lower bounds, and the worst-case performance bound has been derived for one heuristic. Lee and Kim [12] proposed a B&B algorithm for the two-stage hybrid flow shop setting same as in [9] but with the objective of minimizing total tardiness of jobs. Haouari et al. [13] developed a B&B method to minimize makespan in a two-stage identical parallel machine hybrid flow shop

* Corresponding authors. Tel./fax: +86 21 6598 2806.
*E-mail addresses:* shijinwang0223@yahoo.com.cn (S. Wang), mingliu@tongji.edu.cn (M. Liu).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.10.015

problem. Fei et al. [14] treated the daily operating theater scheduling problem into a two-stage hybrid flow shop scheduling problem and used a hybrid genetic algorithm to solve it. Hmida et al. [15] proposed a climbing depth-bounded discrepancy search (CDDS) method to solve the two-stage hybrid flow shop scheduling problem in which each stage has multiple parallel identical machines, with the objective of makespan minimization.

Besides the scheduling problems in two-stage hybrid flow shop setting mentioned above, the current trend in the development of two-stage hybrid flow shop has been that of increasing the complexity and practical relevance of the models. Multi-process task scheduling [16], availability constraints [17,18], setup time [19,20], reentrant [20,21], no-wait [19,22,23] are typical constraints that are included to extend the standard two-stage HFS into more complicated but more practical problems. This paper falls into the two-stage hybrid flow shop setting with no-wait job processing with respect to makespan minimization.

In no-wait environment, every job must be processed from start to finish without any interruption either on or between machines [24]. This constraint was motivated by many types of manufacturing environments: (1) production process with physical space limitation is such a typical problem due to limited warehouse space available for finished goods, space-efficient way to reduce the holding costs, and requirements of quality of service to customers [25]. (2) Scheduling goods with high decay rate. This is the case of the production of steel, concrete wares, chemical products or food [26]. For example, in food processing industry, the canning operation must immediately follow cooking to ensure freshness [24].

Given its importance and complexity, the two-stage no-wait hybrid flow shop scheduling problem has been studied since 1990s. Srishandarajah [27] addressed the issue of worst case bound for two-stage no-wait hybrid flow shop scheduling problem in which there is a single machine at stage one and multiple identical machines at stage two with the objective of makespan minimization and established a tight worst case bound of $r=3-1/m$ using an arbitrary sequence for the list scheduling algorithm, where $m$ is the number of identical parallel machines at the second stage. Hall and Sriskandarajah [24] conducted a survey of the no-wait and blocking scheduling in different shop configurations including deterministic flow shop, job shop and open shop based on papers published before 1993. And they suggested to use meta-heuristic methodologies (e.g., simulated annealing and tabu search) to search good quality solutions to such problems. Cheng et al. [28] studied the computational complexities of some two and three-stage no-wait flow shop scheduling problems to minimize makespan. However, in one stage of all problems they considered, all the jobs require a constant processing time. In this paper, we extended the two-stage no-wait problem without specific requirements of constant processing time on either stage. Liu et al. [29] proposed a greedy heuristic named least deviation (LD) algorithm for the same problem in [27] and the worst case performance of LD is also $3-1/m$. Xie et al. [22] extended the LD into minimum deviation algorithm (MDA) for two-stage no-wait hybrid flow shop scheduling with multiple identical machines at both stages with the objective of makespan minimization. Guirchoun et al. [30] considered a computer system as a two-stage hybrid flow shop with a single machine at the first stage (as a server) and $m$ parallel machines at the second stage with a no-wait constraint between the two stages. The objective is to minimize total completion time. A mathematical formulation has been proposed for the general case and two polynomial time algorithms have been proposed for two particular cases with $m=2$: the processing times at the second stage are all less than 1 or the processing times at the second stage are all strictly greater than 1, both with the processing times at the first stage are all 1. Carpov et al. [23] studied two versions (the classical and no-wait) of the two-stage hybrid flow shop problem in which there are precedence constraints and parallel machines in second stage and the objective is makespan minimization. An adaptive randomized list scheduling (ARLS) heuristic, together with two priority rules, has been proposed for solving both problem versions.

From the literature mentioned above, it can be found that most research in two-stage no-wait hybrid flow shop scheduling focus on the development of heuristics, or optimal solutions for special cases and small scale problem. Due to the NP-hardness of the problem, meta-heuristic (like genetic algorithm) is usually employed to obtain good solutions to large scale problem with reasonable computational time. However, to the best of our knowledge, there is no report on genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem. Hence, in this context, this paper proposes a genetic algorithm method for the two-stage no-wait hybrid flow shop in which there is one single machine at stage one and multiple parallel identical machines at stage two with the objective of makespan minimization.

The rest of the paper is organized as follows: Section 2 describes the problem formally and some properties of the problem are derived. In Section 3, some heuristic methods are introduced to get the upper bound of the problem. Then in Section 4, the specific implementation details of the proposed genetic algorithm are presented. Section 5 illustrates the computational experiments employed to test the performance of the genetic algorithm. After describing the data generation and the parameter tuning for genetic algorithms, we report the computational results in comparison with the lower bound of the problem. Finally, Section 6 concludes the paper and gives future research.

## 2. Problem description and some properties

We consider the no-wait scheduling problem to minimize makespan in two-stage hybrid flow shop environment. The problem can be described formally as follows.

Consider a set $J=\{1,2,\dots,n\}$ of $n$ jobs to be processed in a two-stage hybrid flow shop, where there is a single machine $A$ at stage one and $m(m\geq 2)$ identical parallel machines $B=\{B_1,B_2,\dots,B_m\}$ at stage two. Each job $J_i$ $(i=1,2,\dots,n)$ is to be processed first at stage one with processing time $a_i$ and then at stage two with processing time $b_i$. With zero intermediate storage in between stages, if machine $A$ finishes the processing of any given job, the job must be proceeded immediately on one of the machines at stage two. This phenomenon is referred to as no-wait. The objective is to find a schedule

$$\pi=\left\{\begin{matrix}[1],[2],\ldots,[i],\ldots,[n]\\ B[1],B[2],\ldots,B[i],\ldots,B[n]\end{matrix}\right\}$$

to minimize the makespan (i.e., maximum completion time), denoted by $C_{\max}$. Here, $[i]$ represents $i$th job in the schedule and $B[i]$ indicates on which machine at stage two this job is processed. We have the following assumptions for the problem as well:

(1) All machines and all jobs are available from time $t=0$.
(2) Each job can only be processed on one machine at a time, and each machine can process no more than one job at a time.
(3) Pre-emption of jobs is not allowed.
(4) Without loss of generality, the processing times on machines are integers.
(5) It is assumed that $m<n$. Otherwise, it is identical to viewing the problem in the delivery time model, i.e., single machine scheduling to minimize the time by which all jobs have been