European Journal of Operational Research 207 (2010) 1086–1095

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

Shifting representation search for hybrid flexible flowline problems

Thijs Urlings [a,*], Rubén Ruiz [a], Thomas Stützle [b]

[a] Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain
[b] IRIDIA, CoDE, Université Libre de Bruxelles (ULB), Brussels, Belgium

ARTICLE INFO

Article history:
Received 30 July 2009
Accepted 24 May 2010
Available online 1 June 2010

Keywords:
Hybrid flexible flowline
Realistic scheduling
Precedence constraints
Setup times
Time lags
Local search

ABSTRACT

This paper considers the hybrid flexible flowline scheduling problem with a set of additional restrictions and generalizations that are common in practice. These include precedence constraints, sequence dependent setup times, time lags, machine eligibility and release times. There are many potential solution representations for this problem, ranging from simple and compact, to more complex and complete. Typically, when choosing the degree of detail of the solution representation, a tradeoff can be found between efficiency of the algorithm and the size of the search space. Several adaptations of existing methods are introduced (memetic algorithm, iterated local search, iterated greedy), as well as a novel algorithm called shifting representation search (SRS). This new method starts with an iterated greedy algorithm applied to a permutation version of the problem and at a given time, switches to an iterated local search on the full search space. As far as we know, this shift of the solution representation is new in the scheduling literature. Experimental results and statistical tests clearly prove the superiority of SRS compared with classical and existing methods.

1. Introduction

Many studies have demonstrated the need for research addressing real-world production scheduling problems. In fact, a large part of the scheduling literature considers basic scheduling problems like the single machine, the parallel machine, or the permutation flowshop problem. Unfortunately, they represent only a simplified version of most production environments. Medium size and large factories typically consist of several production stages, with parallel machines in at least one stage. In the literature, this setting is called a hybrid flexible flowline (HFFL). In practice, sequence dependent setup times are a very common additional restriction. Of the more than 300 papers considered in the review on scheduling with setup times or costs by Allahverdi et al. (2008), about 80 papers treat the single machine problem, 70 parallel machines, 80 flowshops with a single machine in each stage and only 18 hybrid flexible flowline problems. Linn and Zhang (1999) stress in their review on HFFL scheduling the gap between theory and practice, even for the HFFL problems, which can be considered as a common machine setting in practice. Problem restrictions that are very common in industry are usually considered, if at all, in separate papers. Doing so, the scheduling problem is not treated in its full complexity, and interactions between constraints are not revealed. For more relevant references supporting this criticism, the reader is referred to Ruiz et al. (2008).

The purpose of this study is to provide methods that obtain close-to-optimal solutions for a realistic HFFL scheduling problem, within a time frame that is acceptable in industrial environments. The existence of two problem features, namely the assignment of tasks to machines and the order in which each machine processes those tasks, make the considered problem more difficult to solve than standard flowshop problems. Note that the hybrid flexible flowline problem is more difficult than its permutation version, where the permutation of jobs cannot change from one stage to the other. While in the permutation problem the combination of a single job permutation and a set of machine assignments suffices to represent a solution completely, in the unrestricted hybrid flexible flowline problem, for each stage a job permutation is required.

By applying algorithm engineering approaches, we have adapted existing algorithms such as a memetic, an iterated local search and an iterated greedy algorithm to the specific problem, analysed their results and developed several improvements. The insights obtained during the design and analysis process have led to a novel algorithm, consisting of two phases with a distinct solution representation in each phase.

In order to tackle the difficulty of multiple problem features, in the first phase, the new algorithm explores only the job order and uses a single machine assignment rule to handle the task assignment. In a second phase, the algorithm searches the full solution space. Thanks to this strategy, it yields better results than all previously tested methods. To the best of our knowledge, there have

* Corresponding author. Tel.: +34 963 879 952; fax: +34 963 877 239.
E-mail addresses: thijs@iti.upv.es (T. Urlings), rruiz@eio.upv.es (R. Ruiz), stuetzle@ulb.ac.be (T. Stützle).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.05.041

been no attempts to cope with these two problem features in this way. The rest of this paper is structured as follows: Section 2 gives a short literature review on realistic scheduling, hybrid flexible flowline problems and local search algorithms. In Section 3 we introduce the problem we consider, with all its constraints and generalizations. The earlier proposed algorithms are described in Section 4, whereas Section 5 describes the development of the new metaheuristics. Section 7 contains the experimental results and in Section 8 the conclusions and some ideas for future research are given.

2. Literature review

In 1999, the two first reviews on hybrid flexible flowline problems appeared, one by Vignier et al. (1999) and one by Linn and Zhang (1999). The latter concludes that there exists a gap between theory and practice and that there is a need of future research in this direction. Quadt and Kuhn (2007) give a taxonomy for hybrid flexible flowline problems. More recent reviews are provided by Ruiz and Vázquez Rodríguez (2010) and Ribas et al. (2010). In the survey on exact methods for the hybrid flowshop, Kis and Pesch (2005) stress the progress of those methods, due to the development of new tight lower bounds. However, the addition of restrictions such as sequence dependent setup times make lower bounds such as the one recently proposed by Haouari and Hidri (2008) inapplicable for the problem considered here. Ruiz et al. (2008) presented some heuristics and a mixed integer programming model to obtain exact results for instances with a maximum of 15 jobs without the use of lower bounds for the highly constrained HFFL problem that is addressed in this paper. A memetic algorithm is introduced in Urlings and Ruiz (2007), in order to solve larger instances efficiently. In Urlings et al. (2010), genetic algorithms are presented for the same problem. These papers will be described in more detail throughout this article. Algorithms for less general HFFL problems can be found in, for example, Kurz and Askin (2004), Ruiz and Maroto (2006) and Yang et al. (2007), among others. More recently, Jungwattanakit et al. (2009) have considered a hybrid flowshop problem with unrelated machines in each stage and sequence dependent setup times. The authors compare a genetic algorithm, a tabu search and a simulated annealing approach.

The local search methods we present in this study have previously been used for other problems with great success. The design of iterated local search algorithms is described by den Besten et al. (2001). Ruiz and Stützle (2007) implement an Iterated Greedy algorithm to solve the regular flowshop and apply the same algorithm to the flowshop with sequence dependent setup times in Ruiz and Stützle (2008). In both cases, the Iterated Greedy algorithm outperforms the state of the art in a significant way. Memetic algorithms can be found in many works. Gonçalves et al. (2005) apply, within a genetic algorithm, a local search on the critical tasks in a jobshop problem. In Jenabi et al. (2007), a memetic algorithm and simulated annealing are implemented for the combination of a lot sizing and hybrid flexible flowshop problem. Unfortunately, the results are hard to compare, since the algorithms are not stopped after the same computation time.

Since the scheduling literature in general is more concentrated on less restricted problems, the number of strongly related papers is rather small.

3. Problem description

We consider a HFFL problem with many realistic constraints that can be defined as follows. The shop consists of an ordered set of production stages $M = \{1, \dots, m\}$, where each stage i contains a set of unrelated parallel machines $M_i = \{1, \dots, m_i\}$. A set of jobs $N = \{1, \dots, n\}$ is to be processed on this flowline. The processing time for job j on machine l at stage i is given by p_{ilj}. Each machine can only process one job at a time. Furthermore, we take into account a number of restrictions occurring in industrial environments. Jobs might skip stages; consequently a job j requires processing only at a subset $F_j \subseteq M$ of all stages. Machine eligibility reduces the set of machines able to process job j at stage $i \in F_j$ to $E_{ij} \subseteq M_i$. We consider also general precedence constraints, which enforce that a job j cannot be processed before all predecessors P_j have been finished completely. Machines are not necessarily available from time zero, but have individual release dates. These release dates prohibit a machine l within stage i to process any job before its release date rm_{il}. Between two consecutive jobs j and k on machine l at stage i, a setup time S_{iljk} needs to be taken into account. The time depends on the machine and on both jobs (hence called sequence dependent setup time). This setup can be either anticipatory or not, depending on the binary parameter A_{iljk}. In the case of an anticipatory setup, the setup can be performed directly when the previous job is completed at the current machine. In the contrary case, when setup is not anticipatory, setup can only be done when the job has arrived at the current machine. The recent review of Allahverdi et al. (2008) shows that setup times arise in many practical applications. The time lag for job j between stage i and the next visited stage, is given by lag_{ilj}, where l represents the machine job j is assigned to at stage i. This lag could be negative (overlap) or positive (waiting time).

Many objectives can be defined for a production environment. The objective of the HFFL we consider, is to have all jobs completed as early as possible i.e., to minimize the maximum completion time or makespan. In an economic sense, this can be seen as the goal to maximize efficiency. Other objectives that represent, for example, client satisfaction, such as tardiness, are not considered here. The makespan can formally be described as $C_{\max} = \max_{i \in M, j \in N} C_{ij}$, where C_{ij} denotes the completion time of job j at stage i.

This problem is $\mathcal{NP}$-hard, since many simplifications of it have been shown to be $\mathcal{NP}$-hard. The permutation flowshop, for example, is a special case and it is well known to be $\mathcal{NP}$-hard. Using the three-field notation, the problem can be denoted as follows:

$$\text{HFFLm}, \left(\left(RM^{(i)} \right)_{i=1}^{(m)} \right) / rm, lag, S_{ijk}, M_j, prec / C_{\max}$$

The problem has been studied before in Ruiz et al. (2008), in Urlings and Ruiz (2007) and in Urlings et al. (2010).

3.1. Problem applications

Many applications of the problem treated in this paper can be found in the production industry. A good example is the ceramic tile manufacturing sector. In this production process, several stages can be distinguished: raw material preparation, molding, drying, glazing, kiln firing and classification among others. Typically, factories have more than one kiln, where the processing speed depends on the characteristics of the kiln and of the tiles to be processed. The glazing stage is optional; some tiles are produced without glaze.

Some machines are specialized for certain types of tiles. This might result in higher speeds for those tiles, but it also leads to machine eligibility constraints. At the moment of scheduling, machines are usually processing earlier scheduled tasks. No jobs can be assigned to a machine before it finishes those pending tasks. Some more complex ceramic structures are composed of pieces that need to be finished before the total structure can be processed. The composition of those pieces cannot be started before all of them are finished, hence introducing precedence constraints.