Computers & Industrial Engineering 126 (2018) 435–450

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

# Scheduling in a two-stage flowshop with parallel unrelated machines at each stage and shared resources

Ewa Figielska

*Warsaw School of Computer Science, Marka Edelmana 17, 00-169 Warsaw, Poland*

ARTICLE INFO

*Keywords:*
Scheduling
Flowshop
Resource constraints
Heuristic
Column generation

ABSTRACT

The paper deals with the problem of preemptive scheduling in the two-stage flowhop with parallel unrelated machines at both the stages and renewable resources shared among the stages. A novel heuristic is proposed which minimizes the flowshop makespan taking into account both the stages simultaneously. This heuristic introduces a new concept of the sets of jobs which are allowed to be processed at stage 1 and stage 2 in successive time intervals. The definition of these sets is based on a priority rule. The constraints resulting from this definition are included into the optimization problem solved by a column generation (CG) algorithm. The CG algorithm creates the schedule which is composed of partial schedules assigning jobs to machines for simultaneous processing at the first and second stages during some periods of time, so that the resource constraints are satisfied at any moment. We developed four heuristic algorithms which start either from a straightforward initial solution or from a solution provided by a linear programming based procedure, and use a simulated annealing or tabu search procedure to create partial schedules in each successive iteration. The results of the extensive computational experiment indicate that these algorithms provide good quality solutions with reasonable computational effort, even for the difficult problems with strong resource constraints.

## 1. Introduction

The flowshops with parallel machines (FSPM) have received considerable attention from researchers since they were first considered in the literature in 1970s. The surveys on FSPM can be found in Linn and Zhang (1999), Kis and Pesch (2005), Ribas, Leisten, and Framiñan (2010), Ruiz and Vazquez-Rodriguez (2010). The majority of papers on FSMP deal with machines that are identical. However, a flowshop containing unrelated machines, where the processing time of a job depends on the machine and the job, is a more general case and therefore more likely to be found in practice. For example, papers (Lei & Zheng, 2017; Rashidi, 2010; Ruiz & Maroto, 2006) addressed flowshop scheduling problems with unrelated parallel machines and solved them by means of various metaheuristic algorithms. In the system with unrelated machines, jobs choose machines on which their processing times are short, which in turn can improve the makespan. Scheduling problems in FSMP without preemptions are strongly NP-hard in most cases (Gupta, 1988).

Allowing job preemptions can considerably improve the schedule makespan (Błażewicz, Kedad-Sidhoum, Monna, Mounié, & Trystram, 2015; Ebadi & Moslehi, 2012; McNaughton, 1959) but does not simplify the FSMP problem itself. Hoogeveen, Lenstra, and Veltman (1996) have shown that the makespan minimization problem of preemptive scheduling in the two-stage flowshop with two identical parallel machines at one stage and one machine at another is NP-hard in the strong sense. Preemptive scheduling problems arise in the automated manufacturing systems with negligible setups (Azizoğlu, 2003), in the mass production of a large number of products which can be processed in parts, or when an article is produced in a great amount. For example in the textile industry, at the weaving stage (which is usually followed by the dyeing stage), the processing of the article to be woven on one of the parallel looms may be interrupted and resumed on the same or the other loom (Serafini, 1996). Though the benefits of allowing job preemptions can be significant, there are few studies that considered preemptive scheduling in the context of FSPM. For example, in Djellab and Djellab (2002) a preemptive FSPM with precedence relations between jobs was addressed and a class of precedence constraints for which the problem is polynomially solved was proposed. Jiang, Dong, and Ji (2013) considered a preemptive scheduling problem on two parallel machines with a single server. In this problem, a server can be considered as a single machine of the first stage in a two-stage flowshop and parallel machines form the second stage at which job preemptions are allowed. The paper (Chen, Lu, & Yuan, 2015) dealt with a two-stage scheduling problem in which jobs are first processed preemptively on identical machines at a manufacturing facility and then are delivered to their customers by one vehicle which can take one job at each

*E-mail address:* efigielska@poczta.wwsi.edu.pl.

https://doi.org/10.1016/j.cie.2018.09.038
Received 14 November 2017; Received in revised form 4 September 2018; Accepted 20 September 2018
Available online 24 September 2018
0360-8352/ 

shipment. The preemptive scheduling problems in a flowshop with additional renewable resources were addressed in papers by Figielska (2008, 2009, 2010, 2014).

Scheduling problems with additional renewable resources, i.e. resources whose usage is constrained at every moment, have been widely investigated for the one stage systems (e.g. Afzalirad & Rezaeian, 2016; Błażewicz, Cellary, Słowiński, & Węglarz, 1987; Edis & Ozkarahan, 2012; Özpeynirci, Gökgür, & Hnich, 2016; Różycki & Węglarz, 2012; Słowiński, 1980). For the case where resource requirements are of 0-1 type, the polynomial methods have been provided by Słowiński (1981) and extended by de Werra (1988). In the case of arbitrary job resource requirements, the problem of preemptive scheduling on unrelated parallel machines in one stage system can be formulated as a non-linear problem and solved by means of a column generation (CG) algorithm decomposing the original problem into two, alternatively solved, sub-problems: a linear programming (LP) problem and 0–1 integer programming problem (Figielska 2009; Toczyłowski, 1989).

Real-life systems that are encountered in chemical, food, cosmetics or textile industries usually consist of a number of production centers with parallel machines and are subjected to some resource constraints associated with limited availability of the resources such as skilled labor, tools, power (renewable resources). In the literature on flowshop scheduling there are a few studies that consider additional resource constraints. For example, Cheng, Lin, and Huang (2012) analyzed the process of tearing-down and reconstructing buildings as a two-machine flowshop in which a job starting on the first machine consumes some amount of a resource taken from the resource pool of limited availability and a different amount of the resource is returned by the job when it completes its processing at the second machine. Mehravaran and Logendran (2013) addressed the flowshop with single machines at the stages and the constrained labor resource, where a job for processing on a machine requires one labor with appropriate skill level. In Laribi, Yalaoui, Belkaid, and Sari (2016) a two-machine flowshop was considered in which non-renewable resources are consumed by jobs processed on the second machine. Waldherr and Knust (2017) considered a synchronous flow shop in which every job needs a single resource during its entire production process on all the machines. Figielska (2008, 2010) addressed the problems of preemptive scheduling in the flowshops with parallel unrelated machines and renewable resources, in which job resource requirements were assumed to be of 0–1 type. Papers Figielska (2009, 2014) considered two-stage flowshops with arbitrary resource requirements and parallel unrelated machines at one of the stages. In Figielska (2009) the algorithm was presented which first minimized the length of the schedule on the parallel machines of the first stage under the resource constraints and then the resulting partial schedules were sequenced so as to achieve as small as possible the idle time of a single machine at stage 2. In Figielska (2014), where shared resources were considered, first jobs were sequenced on a single machine at stage 1, and then the length of the schedule at stage 2 was minimized by solving a resource constrained problem with non-zero ready times of jobs and different resource availabilities throughout the scheduling period.

The examples of resources shared among the processing centers include the resources that are capable of migrating to processing centers as needed. Workers are often cross-trained to develop the skills required to perform different tasks associated with multiple processing centers (Daniels & Mazzola, 1994). Another example of shared resources can be equipment, personnel and information technology which is used in both of the two stages of hospital operations: to generate a medical record to track treatments, tests, drug dosages and costs in the first stage and to generate the second stage patient services (Chen, Du, Sherman, & Zhu, 2010).

In the present paper, we solve a general problem of preemptive scheduling in a two-stage flowshop with parallel unrelated machines at each of the stages and jobs requiring some amounts of additional renewable resources which are shared among the stages. When solving this problem, one must cope with difficulties connected with the flowshop scheduling as well as those imposed by the resource constraints, which must be satisfied in the whole system at every moment. In order to minimize the makepan effectively for such a system, a novel approach handling both the stages simultaneously is provided.

The remaining of the paper is organized as follows. The next section presents the framework of the heuristic. The detailed description of the problem is provided in Section 3. Section 4 presents an illustrative example. In Section 5, the details of the heuristic are provided. The results of a computational experiment are presented and analyzed in Section 6. Section 7 summarizes the paper.

## 2. Framework of the heuristic

In this paper, first we order the jobs according to a priority rule. The obtained sequence is denoted by P. Then, we divide the scheduling period into $n + 1$ ($n$ is the number of jobs) time intervals and impose the requirements that the job standing in position $k$ in sequence P, at stage 1 is allowed to be processed in intervals $1, ..., k$ and must be completed before the end of interval $k$, and at stage 2 it can be processed in intervals $k + 1, k + 2, ..., n + 1$. Therefore, in interval $k$, at stage 1 jobs whose positions in sequence P are greater than or equal to $k$ are allowed to be processed (the set of these jobs is denoted by $Q_1^k$), and at stage 2 jobs from positions smaller than $k$ are allowed to be processed (the set of these jobs is denoted by $Q_2^k$). Because of the possibility of preemptions, parts of one job can be processed in different time intervals and in one time interval parts of different jobs can be processed.

Jobs are processed on parallel machines of two stages, using resources whose availability is limited at any moment. It is convenient to present a feasible solution as a set of partial schedules. In a partial schedule, jobs (parts of jobs) are processed simultaneously on the machines of both the stages during some period of time so that the resource constraints are satisfied at any moment. The length of a time interval is equal to the sum of the durations of the partial schedules executed in this interval. The length of the whole schedule (i.e. the makespan in the flowshop) is equal to the sum of the durations of all the partial schedules.

Our aim is to solve the scheduling problem so that the makespan of the final schedule is as short as possible. So, we formulate the optimization problem, in which the objective is to minimize the sum of the durations of all the partial schedules, subject to the constraints ensuring the completion of all the jobs in the flowshop and the constraints on the availability of resources shared among the stages. In all the constraints in the optimization problem, the restrictions on the availability of jobs in successive time intervals are taken into account.

In order to minimize the makespan, we use a modified column generation (CG) algorithm. The CG algorithm divides the original problem into two problems: the master problem and the sub-problem. In the master problem, the durations of partial schedules are decision variables, the objective function is minimization of the sum of the durations of partial schedules, and the constraints ensure the completion of all the jobs in the flowshop. In the sub-problem, the assignments of jobs to machines in partial schedules are decision variables and constraints are imposed on the availability of the resources. The optimal solution to the dual problem to the master problem is used to formulate the objective function of the sub-problem. The master problem and the sub-problem are solved in each iteration of the CG algorithm. In the first iteration, the master problem is solved with the initial set of columns (a column determines the assignment of jobs to machines). In each iteration, a metaheuristic algorithm solves the sub-problems in the successive time intervals, creating a number of columns which, when included in the master problem, improve its solution. Then, the improved solution of the master problem again provides the dual information for the sub-problem, and so on, until no column improving the solution of the master problem can be found.

Four algorithms are developed, HSA1, HSA2, HTS1, HTS2, which