European Journal of Operational Research 247 (2015) 421–440

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility

E. G. Birgin [a,*], J. E. Ferreira [b], D. P. Ronconi [b]

[a] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil

[b] Department of Production Engineering, Polytechnic School, University of São Paulo, Av. Prof. Almeida Prado, 128, Cidade Universitária, 05508-900, São Paulo SP, Brazil

ARTICLE INFO

Article history:
Received 8 October 2014
Accepted 10 June 2015
Available online 25 June 2015

Keywords:
Scheduling
Flexible job shop
Makespan
List scheduling
Beam search

ABSTRACT

An extended version of the flexible job shop problem is tackled in this work. The considered extension to the classical flexible job shop problem allows the precedences between the operations to be given by an arbitrary directed acyclic graph instead of a linear order. Therefore, the problem consists of allocating the operations to the machines and sequencing them in compliance with the given precedences. The goal in the present work is the minimization of the makespan. A list scheduling algorithm is introduced and its natural extension to a beam search method is proposed. Numerical experiments assess the efficiency of the proposed approaches.

1. Introduction

The classical job shop (JS) problem consists of scheduling n jobs on an environment with m machines. Each job is composed by several operations with a linear precedence structure and has a predetermined route through the machines. The flexible job shop scheduling (FJS) problem is a generalization of the JS problem in which there may be several machines, not necessarily identical, capable of processing each operation. The processing time of each operation on each machine is known and no preemption is allowed. The objective is to decide on which machine each operation will be processed, and in what order the operations will be processed on each machine so that a certain criterion is optimized.

This paper considers the extended version of the FJS problem that allows the precedences between the operations to be given by an arbitrary directed acyclic graph instead of a linear order. Therefore, the problem consists of allocating the operations to the machines and sequencing them in compliance with all given precedences. An example of a job with this general type of precedences is presented in Fig. 1. This problem appears in practical and industrial environments, such as the printing industry (Zeng, Jackson, Lin, Gustafson, Hoarau, & Mitchell, 2010), where assembling and disassembling operations are part of the production process. Printing processes can be divided into three major tasks: prepress steps, printing, and postpress steps (Printers National Environmental Assistance Center, 2015). Prepress steps include composition and typesetting, graphic arts photography, image assembly, color separation, and image carrier preparation. Printing can be performed by six separate and distinct processes: lithography, letterpress, flexography, gravure, screen printing, and plate-less technologies. Postpress operations consist of four major processes: cutting, folding, assembling, and binding. There are many additional lesser postpress finishing processes such as varnishing, perforating, drilling, etc. In-line finishing may also be considered as a final step of the postpress operations. These three major steps of the printing process have an obvious precedence constraint. However, within each major step, there are operations with no precedence constraint among them. Pages of a book are divided into signatures (bunch of 8, 16, or 32 individual pages) that can be printed, cut, and folded in separate. The book cover is also an element that can be prepared in separate. Then, all printed and non-printed elements need to be gathered in order to continue the process. See Fig. 2. It is easy to see that this arbitrary-precedences issue of the printing process may be found in most of the practical industrial applications, making the considered problem a problem with a potential wide range of applications. The scheduling performance measure considered in the present work is the makespan minimization.

The flexibility of representing the precedences between the operations of a job with an arbitrary directed acyclic graph instead of a linear order is known as sequencing flexibility, while routing flexibility refers to the possibility of an operation to be performed by a

* Corresponding author. Tel.: +55 11 3091 6135.
E-mail addresses: egbirgin@ime.usp.br (E.G. Birgin), jeferre@ime.usp.br (J.E. Ferreira), dronconi@usp.br (D.P. Ronconi).

http://dx.doi.org/10.1016/j.ejor.2015.06.023
0377-2217/

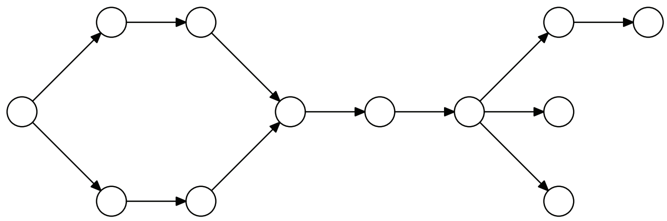

Fig. 1. Example of the precedence constraints of a single job of an instance of the extended version of FJS problem, in which precedences between operations are given by an arbitrary directed acyclic graph.

subset of machines instead of a single machine (this is the flexibility that transform a JSP into an FJSP). Other types of flexibility exist, like producing the same manufacturing feature with alternative operations or sequences of operations, known as processing flexibility. The effects of sequencing flexibility on the performance of dispatching rules used to schedule operations in manufacturing systems was analyzed in Lin and Solberg (1991); Rachamadugu, Nandkeolyar, and Schriber (1993) (see also the references therein). In Sabuncuoglu and Karabuk (1998), a flexible manufacturing system with finite buffer capacities and that considers automated guided vehicles is tackled. Different performance criteria are considered (mean flow time, mean tardiness, and makespan) and an *ad hoc* filtered beam search method is developed. The results of the method are analyzed in order to investigate the effects in the performance of the manufacturing system of incorporating different types of flexibilities. A more recent study can be found in Joseph and Sridharan (2011).

The extended FJS problem considered in the present work is NP-hard, since it has the JS problem (that is known to be NP-hard Garey, Johnson, & Sethi, 1976) as a particular case. Due to its complexity, the number of publications concerned with the exact solution of the FJS problem is very small. Fattahi, Mehrabad, and Jolai (Fattahi, Mehrabad, & Jolai, 2007) proposed a mixed integer linear programming (MILP) model for the FJS problem and used it to solve small and medium-sized instances with a commercial software. A more concise MILP model, that modifies an earlier one presented in Manne (1960) in order to incorporate routing flexibility, was introduced in Özgüven, Özbakır, and Yavuz (2010). More recently, a new MILP model for the extended version of FJS considered in the present work was presented in Birgin, Feofiloff, Fernandes, de Melo, Oshiro, and Ronconi (2014). This model was analyzed using instances from the literature and instances inspired by the printing industry. According to the numerical experiments, the software CPLEX produced better results with the new model than with the one presented in Özgüven et al. (2010).

Several works from the literature proposed heuristic methods to address the makespan minimization in the classical FJS problem. Brandimarte (Brandimarte, 1993), one of the pioneers of this approach, applied dispatching rules to assign each operation of each job to a machine and, in a second phase, employed a tabu search heuristic to define the sequence of the operations on each machine. This kind of strategy is known as hierarchical approach. Tabu search (TS) based heuristics to solve this problem, but in an integrated way (i.e. considering simultaneously the assignment and schedule of the operations), were also developed in Dauzère-Pérès and Paulli (1997); Mastrolilli and Gambardella (2000). A recent literature also includes genetic algorithms (GA) to deal with the FJS problem in an integrated approach. The learnable genetic architecture (LEGA), proposed in Ho, Tai, and Lai (2006), provides an integration between evolution and learning methodologies within a random search framework. In this context, the learning module is used to influence the diversity and quality of offsprings. On the other hand, a traditional GA with improved components selected from the literature and a new mutation assignment operator named intelligent mutation was introduced in Pezzella, Morganti, and Ciaschetti (2008). According to the presented computational tests the proposed GA outperformed other known GAs from the literature and obtained solutions comparable with the ones

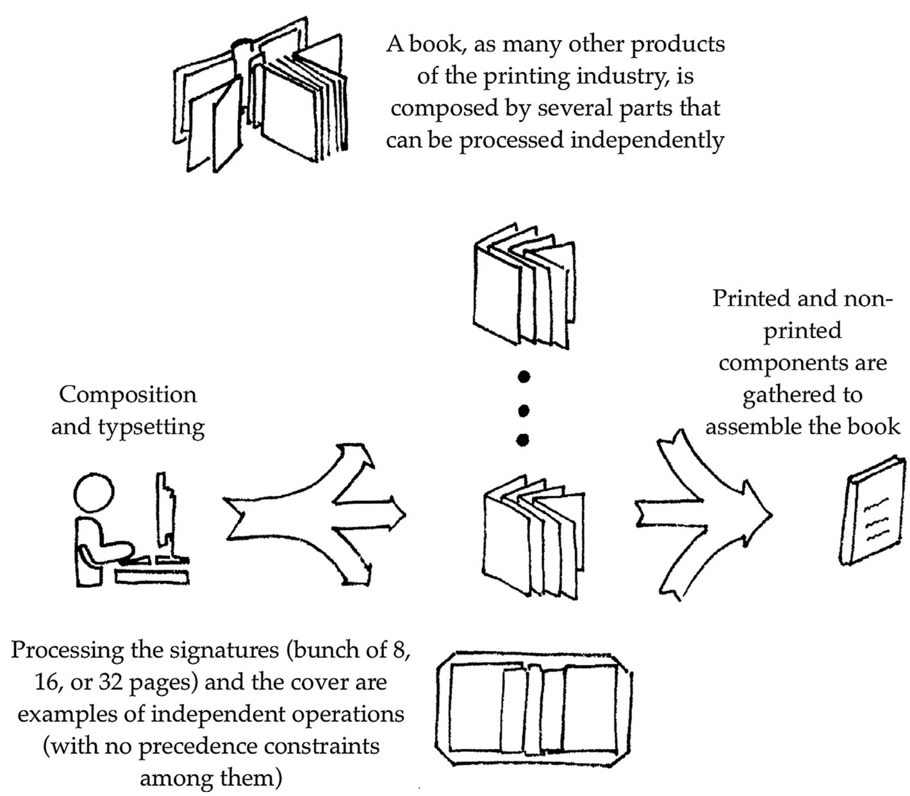

Fig. 2. Illustrative scheme including operations with no precedence constraints among them (the different signatures and the cover) in the printing-industry task of producing a book.