European Journal of Operational Research 000 (2016) 1–14

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

An algorithmic framework for tool switching problems with multiple objectives ☆

Martina Furrer, Torsten Mütze*

Institute of Theoretical Computer Science, ETH Zurich, Switzerland

ARTICLE INFO

Article history:
Received 29 March 2016
Accepted 17 November 2016
Available online xxx

Keywords:
Flexible manufacturing system
Tool switching
Branch-and-bound

ABSTRACT

The tool switching problem is a classical and extensively studied problem in flexible manufacturing systems. The standard example is a CNC machine with a limited number of tool slots to which tools for drilling and milling have to be assigned, with the goal of minimizing the number of necessary tool switches and/or the number of machine stops over time. In this work we present a branch-and-bound based algorithmic framework for a very general and versatile formulation of this problem (involving arbitrary setup and processing times) that allows addressing both of these objectives simultaneously (or only of them), and that improves over several known approaches from the literature. We demonstrate the usefulness of our algorithm by rigorous theoretical analysis and by experiments with both large real-world and random instances.

1. Introduction

The main contribution of this article is an algorithmic framework that improves over several known results for a family of optimization problems on flexible manufacturing systems that involve multiple objectives. Such problems have received considerable attention mainly due to their relevance in numerous industrial applications (Adjiashvili, Bosio, & Zemmer, 2015; Al-Fawzan & Al-Sultan, 2003; Amaya, Cotta, & Fernández, 2008; Bard, 1988; Catanzaro, Gouveia, & Labbé, 2015; Chaves, Lorena, Senne, & Resende, 2016; Crama, 1997; Crama & van de Klundert, 1999; Crama, Kolen, Oerlemans, & Spieksma, 1994; Ghiani, Grieco, & Guerriero, 2010; Gray, Seidmann, & Stecke, 1993; Hertz, Laporte, Mittaz, & Stecke, 1998; Keung, Ip, & Lee, 2001; Laarhoven & Zijm, 1993; Laporte, Salazar-González, & Semet, 2004; Mütze, 2014; Privault & Finke, 1995; 2000; Song & Hwang, 2002; Stecke, 1983; Tang & Denardo, 1988a; 1988b; Tzur & Altman, 2004). Specifically, we are considering the so-called tool switching problem on flexible manufacturing systems with arbitrary setup and production times (with different objectives), which is a subproblem of the more general job sequencing and tool switching problem. Here we do not consider this more general problem, but we discuss the relation between both problems below. Before addressing these applications and references in detail, let us begin by stating the problem formally.

1.1. Problem statement

A flexible manufacturing system has m *slots*, each of which can hold one of n available *tools* at a time. We are given a sequence of ℓ *jobs* $J_1, \dots, J_\ell$, where each job $J_i \subseteq [n] := \{1, \dots, n\}$ consists of all tools that need to be mounted to the slots to process the job. The jobs are processed in the given order, i.e., job J_i is processed in the ith step. Denoting the tools of job J_i by $\alpha_1, \dots, \alpha_{|J_i|}$, this is achieved by assigning $\alpha_1, \dots, \alpha_{|J_i|}$ to $|J_i|$ of the slots, and the remaining $m - |J_i|$ slots remain unused in this step. Such an assignment of tools to slots for each of the jobs is a *plan*, described by an $\ell \times m$ matrix. To evaluate the quality of a plan, we consider specific configurations in the plan, so-called tool switches and stops (see Fig. 1). A *tool switch* occurs whenever a tool α is assigned to a slot in some step i_α and a different tool β is assigned to the same slot in a later step i_β, and no tool is assigned to this slot in between. Tool switches are a purely combinatorial property of the plan. To further evaluate the relevance of tool switches, we take into account setup times of the tools and processing times of the jobs. Specifically, each tool $\alpha \in [n]$ has an integer *setup time* $s(\alpha)$ representing the number of consecutive time units it takes to mount tool α to a slot before it can be used, and each job J_i has an integer *processing time* p_i representing the number of

☆ The real-world data used for our experiments was provided by our industrial partner Supercomputing Systems AG, Zurich, Switzerland. See http://www.scs.ch/optimizer for a game-like demonstration of the problem and the industrial applications discussed in this article.

* Corresponding author.
E-mail addresses: martfur@student.ethz.ch (M. Furrer), torsten.muetze@inf.ethz.ch (T. Mütze).

http://dx.doi.org/10.1016/j.ejor.2016.11.034
0377-2217/

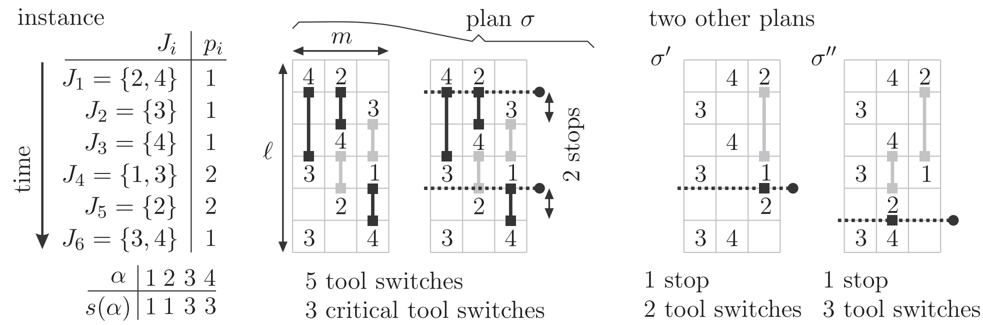

Fig. 1. Problem instance (left) with $\ell = 6$ jobs, $m = 3$ slots and $n = 4$ tools, and three different plans (middle and right). We indicate tool switches in a plan by vertical lines, where critical tool switches are drawn in black and non-critical tool switches in grey. Moreover, we indicate stops by horizontal dashed lines. Resolving the critical tool switches in σ requires two stops, the first one either after J_1 or J_2, and the second one either after J_4 or J_5. The plans σ' and σ'' have the minimal number of stops, σ' also minimizes tool switches (σ minimizes neither stops nor tool switches).

consecutive time units it takes to process the job. We call a switch from tool α to β as above *critical*, if the sum of all processing times p_i with $i_\alpha < i < i_\beta$ is strictly smaller than the setup time $s(\beta)$. Every such critical tool switch requires a *stop* of the manufacturing system at some point in between step i_α and i_β to gain additional time for the setup. We say that such a stop *resolves* the critical tool switch (e.g., the critical switch from tool 4 to 3 on slot 1 in σ can be resolved by a stop either after J_1, J_2 or J_3). A single stop resolves every critical tool switch for which the setup can be performed during this stop (e.g., a stop after J_1 in σ resolves two critical tool switches, and a stop after J_4 resolves one critical tool switch). In particular, there can be fewer stops than critical tool switches, and careful placement of stops is crucial, even for a fixed plan and therefore fixed critical tool switches.

Our objective is to find a plan that minimizes the number of stops and the number of tool switches, where stops are given higher priority than tool switches. We refer to this as *lex-minimization of stops and tool switches*. The order of jobs is fixed and not part of the optimization. We will show that this objective reflects several practical needs and that, furthermore, many other interesting objectives (e.g., minimizing makespan, or minimizing only tool switches or only stops), are subsumed as special cases by choosing the model parameters appropriately.

1.2. Applications

Flexible manufacturing systems as described above have been studied extensively (both Gray et al., 1993 and Crama, 1997 list well over 100 references), and the classical example is a CNC machine that has a tool magazine with a limited number of slots holding various tools for drilling and milling. Minimizing the number of tool switches (Crama et al., 1994; Privault & Finke, 1995; Tang & Denardo, 1988a) or the number of stops (Tang & Denardo, 1988b), sometimes also referred to as *switching instants*, are certainly the most natural objectives in this context, but others have also been studied (see Privault & Finke, 2000; Song & Hwang, 2002; Stecke, 1983). An important feature of these machines is that tool slots in the magazine cannot be accessed individually while the machine is busy, so setups can only be performed during stops – we refer to this scenario as *single magazine* setting. It means that every tool switch is critical, which we can achieve in our model by setting all setup times to ∞ (i.e., to a large enough finite number).

In contrast to that, mailroom insert planning as discussed in Adjiashvili et al. (2015) is an application of our model where finite setup times are the norm. Here, a line of feeders (=slots) inserts various sets (=jobs) of different advertising brochures (=tools) into a newspaper. Each feeder can be accessed individually to exchange a batch of brochures for another one (=a tool switch) without stops if the feeder is unused sufficiently long (=a non-critical tool switch). The instance in Fig. 1 is of this type: e.g., the switch from tool 2 to 1 in slot 3 of the plan σ' does not require a stop, which is typical for such an application, but untypical for the single magazine setting of CNC machines. At this point let us briefly mention three more concrete applications of this type: printed circuit board assembly, chemical processing and pharmaceutical packaging (for details, see Laarhoven & Zijm, 1993; Mütze, 2014; Tzur & Altman, 2004). In all these examples the goal is to minimize costs incurred by tool switches and by stops that interrupt the production.

In our model the ordering of jobs is fixed, but there are variations of the previously mentioned scenarios where ordering the jobs is also part of the decision process, so the goal is to determine an optimal ordering of the jobs *and* an optimal assignment of tools to slots. Unfortunately, all reasonable objectives, in particular minimizing tool switches or stops, are NP-hard in this more general setting already for $m = 2$ slots (Adjiashvili et al., 2015; Crama et al., 1994). In view of this, finding an optimal assignment of tools to slots becomes the more important once a (heuristically computed) job ordering has been determined in the first phase. Put differently, solving the assignment problem optimally allows us to consider the ordering problem separately. In fact, researchers have proposed numerous IP formulations and heuristic/approximative algorithms to tackle the ordering problem (Al-Fawzan & Al-Sultan, 2003; Amaya et al., 2008; Bard, 1988; Catanzaro et al., 2015; Chaves et al., 2016; Crama & van de Klundert, 1999; Crama et al., 1994; Ghiani et al., 2010; Hertz et al., 1998; Laporte et al., 2004; Privault & Finke, 1995; Tang & Denardo, 1988a; 1988b) (see Ermatinger, 2014 for an extensive comparison between these approaches).

Note also that if setup operations can only be performed during stops (as in the single magazine setting of CNC machines), or if the time required for stopping and restarting the manufacturing system is larger than the setup times, minimizing stops is equivalent to minimizing the production makespan. This is the case, e.g., for the above-mentioned mailroom insert planning problem and its relatives. For these scenarios, the lex-minimization of stops and tool switches is therefore the objective of main interest from a practical point of view (cf. Adjiashvili et al., 2015 and our real-world data presented in Section 6; see also the remarks at the end of this article).

1.3. Outline of this article

In Section 2 we present the main results of this work. In Section 3 we introduce the notation needed for the subsequent formal treatment. In Sections 4 and 5 we describe the two main phases of our algorithm. In Section 6 we report on our computational experiments. In Section 7 we indicate some directions for further research.