Arc-flow model and valid inequalities for the job sequencing and tool switching problem with non-identical parallel machines

Arthur Kramer, Khadija Hadj Salem, Alexis Robbes

Arc-flow model and valid inequalities for the job sequencing and tool switching problem with non-identical parallel machines

Arthur Kramer[1 [0000−0002−1991−5046]], Khadija Hadj Salem[2 [0000−0003−3739−2899]], and Alexis Robbes[3 [0000−0003−1077−7211]]

[1] Mines Saint-Étienne, Univ Clermont Auvergne, CNRS , UMR 6158 LIMOS, Institut Henri Fayol, F-42023 Saint-Étienne, France
arthur.kramer@emse.fr

[2] L@bISEN, Usine du Futur, ISEN Yncréa Ouest, 33 Quater Avenue du Champ de Manœuvres, F-44470 Carquefou, France
khadija.hadj-salem@isen-ouest.yncrea.fr

[3] ONERA/DTIS, Université de Toulouse, 2 av. Edouard Belin, BP 74025 F-31055, Toulouse Cedex 4, France
alexis.robbes@onera.fr

Abstract. In this work, we study the job sequencing and tool switching problem with non-identical parallel machines that arises in flexible manufacturing systems where machines can process a variety of jobs depending on the loaded tools. We propose an arc-flow model and valid inequalities to solve the problem. Preliminary experiments on instances from the literature evaluate the effectiveness of the arc-flow as well as the impact of valid inequalities.

Keywords: Job sequencing and tool switching problem · Non-identical parallel machines · Arc-flow formulation · Valid inequalities.

1 Problem description

The *job sequencing and tool switching problem with non-identical parallel machines* (SSP-NPM) is a generalization of the classical *job sequencing and tool switching problem* (SPP), firstly defined in [4]. These problems arise in flexible manufacturing systems where flexible machines are available to process a variety of jobs with specific tool requirements (see [1] for a literature review on SSPs).

The SSP-NPM was first proposed by [3]. This problem considers a set of jobs $\mathcal{J}$ to be processed in a set of non-identical parallel machines $\mathcal{M}$. The set of tools required for processing the jobs are denoted by $\mathcal{T}$. Each job $j \in \mathcal{J}$ has tool requirements, represented by $\mathcal{T}_j \subseteq \mathcal{T}$, so that it can be processed in machine $k \in \mathcal{M}$ only if all tools $t \in \mathcal{T}_j$ are loaded in this machine during its processing. For each job a machine-dependent processing time p_{jm} is required. In turn, the machines exhibit distinct magazine capacities C_m, representing the maximum quantity of tools that can be loaded simultaneously. Then, each time a tool switch is performed, a machine-dependent switch time sw_m is incurred. Then, the SSP-NPM requires scheduling the jobs on the unrelated parallel machines with limited tool capacity, so that the makespan is minimized. The study on SSP-NPM is still in its early stages, especially regarding exact methods. In this direction, we highlight the contribution of [2] that is, to the best of our knowledge, the only work to propose *mixed integer linear programming* (MILP) models to the SSP-NPM. Thus, in this work, we propose a MILP arc-flow model (AF) and valid inequalities (VIs) to solve the makespan minimization SSP-NPM.

2 Mathematical formulation and valid inequalities

Let us define the acyclic directed multigraph $G = (N, A)$ and $\mathcal{H} = \{0, 1, \ldots, H\}$ as the set of time instants. The set N is composed of vertices (k, p), for each $k \in \mathcal{M}$ and $p \in \mathcal{H}$ and the set of arcs A contains all arcs representing job processing ($\mathcal{A}$), tool loading ($\mathcal{O}$), setup times ($\mathcal{S}$) and loss arcs ($\mathcal{L}$), such that $A = \mathcal{A} \cup \mathcal{O} \cup \mathcal{S} \cup \mathcal{L}$. The set $\mathcal{A} = \{(j, k, p, q) : j \in \mathcal{J}; k \in \mathcal{M}; p \in \mathcal{H}; q = p + p_j^k \in \mathcal{H}\}$ contains job arcs from node (k, p) to node (k, q), representing the processing of job j from time p to time q on machine k. Similarly, the set $\mathcal{O} = \{(i, k, p, q) : i \in \mathcal{T}; k \in \mathcal{M}; p \in \mathcal{H}; q \in \mathcal{H} | q > p\}$, contains the tool arcs from node (k, p) to node (k, q) and is used to define the presence of tool i on a slot of machine k during the time interval $[p, q)$. The set $\mathcal{S} = \{(k, p, q) : k \in \mathcal{M}; p \in \mathcal{H}; q = p + l \cdot sw_k | l \in \{1, \ldots, C_k\}$ and $q \in \mathcal{H}\}$ contains the arcs from node (k, p) to node (k, q) that represent setup operations. Finally, the set $\mathcal{L} = \{(k, p) : k \in \mathcal{M}, p \in \mathcal{H}\}$ contains the loss arcs, i.e., arcs connecting node (k, p) to sink node (k, H). Thus, our proposed AF model uses a continuous variable ($C_{\max}$) to represent the makespan and four sets of binary variables to represent job processing, tool loading, setup (tool switching), and loss arcs. Variable x_{jpq}^k assumes value 1 if job arc $(j, k, p, q) \in \mathcal{A}$ is taken, 0 otherwise; variable y_{ipq}^k that is associated with tool loading and assumes value 1 if tool arc $(i, k, p, q) \in \mathcal{O}$ is taken, 0 otherwise; variable s_{pq}^k that is associated with a tool switching operation and assumes value 1 if setup arc $(k, p, q) \in \mathcal{S}$ is taken, 0 otherwise; finally, variable l_p^k takes value 1 if loss arc $(k, p) \in \mathcal{L}$ is taken, 0 otherwise. Then, the proposed AF for the SSP-NPM is as follows:

$$\min \quad C_{\max} \tag{1}$$

subject to:

$$\sum_{(j,k,p,q) \in \mathcal{A}} x_{jpq}^k = 1 \quad j \in \mathcal{J} \tag{2}$$

$$\left(\sum_{(j,k,q,r) \in \mathcal{A}} x_{jqr}^k + \sum_{(k,q,r) \in \mathcal{S}} s_{qr}^k + l_q^k \right) - \left(\sum_{(j,k,p,q) \in \mathcal{A}} x_{jpq}^k + \sum_{(k,p,q) \in \mathcal{S}} s_{pq}^k \right) = \begin{cases} 1, & \text{if } q = 0 \\ -1 + \sum\limits_{(k,p) \in \mathcal{L}} l_p^k, & \text{if } q = H \\ 0, & \text{otherwise} \end{cases} \quad \substack{k \in \mathcal{M} \\ q \in \mathcal{H}} \tag{3}$$

$$\left(\sum_{(i,k,q,r) \in \mathcal{O}} y_{iqr}^k + \sum_{(k,q,r) \in \mathcal{S}} \frac{(r-q)}{sw_k} s_{qr}^k \right) - \left(\sum_{(i,k,p,q) \in \mathcal{O}} y_{ipq}^k + \sum_{(k,p,q) \in \mathcal{S}} \frac{(q-p)}{sw_k} s_{pq}^k \right) = \begin{cases} C_k, & \text{if } q = 0 \\ -C_k, & \text{if } q = H \\ 0, & \text{otherwise} \end{cases} \quad \substack{k \in \mathcal{M} \\ q \in \mathcal{H}} \tag{4}$$

$$\sum_{\substack{(j,k,p,r) \in \mathcal{J} \\ p \le q < r}} x_{jpr}^k - \sum_{\substack{(t,k,p,r) \in \mathcal{O} \\ p \le q < r}} y_{ipr}^k \le 0 \quad \substack{k \in \mathcal{M} \\ j \in \mathcal{J} \\ i \in \mathcal{T}_j \\ q \in \mathcal{H}} \tag{5}$$

$$\sum_{(i,k,q,r) \in \mathcal{O}} y_{iqr}^k - \sum_{(k,p,q) \in \mathcal{S}} \frac{(q-p)}{sw_k} s_{pq}^k = 0 \quad \substack{k \in \mathcal{M} \\ q \in \mathcal{H}/\{0,H\}} \tag{6}$$

$$C_{\max} \ge \sum_{(j,k,p,q) \in \mathcal{A}} q \cdot x_{jpq}^k \quad \forall j \in \mathcal{J} \tag{7}$$

Constraints (2) ensure that all jobs are processed. Constraints (3) and (4) are flow conservation constraints on jobs and tools, respectively. Constraints (5) impose that a job can be processed on a machine only if all of its required tools are loaded in this machine during its processing. Constraints (6) impose a setup time for each switch operation and Constraints (7) define the makespan. The domain of variables is not shown in the model but is defined in the description of the model: $x^k_{jpq}, (j,k,p,q) \in \mathcal{A}$; $y^k_{ipq}, (i,k,p,q) \in \mathcal{O}$; $s^k_{pq}, (k,p,q) \in \mathcal{S}$; and $l^k_p, (k,p) \in \mathcal{L}$ are binary variables.

To improve the performance of the AF model, we propose some VIs that are presented in the following:

$$\sum_{(k,q,r)\in\mathcal{S}} s^k_{qr} + \sum_{(k,p,q)\in\mathcal{S}} s^k_{pq} \leq 1 \qquad \substack{k\in M \\ q\in\mathcal{H}/\{0,H\}} \tag{8}$$

$$\sum_{(j,k,q,r)\in\mathcal{A}} x^k_{jqr} - \sum_{(k,q,r)\in\mathcal{S}} s^k_{qr} \geq 1 \qquad k \in \mathcal{M} \tag{9}$$

$$C_{\max} \geq \sum_{(k,p)\in\mathcal{L}} p \cdot l^k_p \text{ and } \sum_{(k,p)\in\mathcal{L}} l^k_p = 1 \qquad k \in \mathcal{M} \tag{10}$$

$$\sum_{(k,q,r)\in\mathcal{S}} \frac{(r-q)}{sw_k} s^k_{qr} - \sum_{(i,k,p,q)\in\mathcal{O}} y^k_{ipq} = 0 \qquad \substack{k\in\mathcal{M} \\ q\in\mathcal{H}/\{0,H\}} \tag{11}$$

VI (8) forbid consecutive setup arcs on a machine; (9) impose that the number of setup arcs on a machine must be smaller than the number of job arcs; (10) redefine the $C_{\max}$; and (11) state that after a setup time it must be a tool arc.

3 Preliminary results and future works

We tested our proposed AF model (1)–(7) and the VI (8)–(11) on a set of benchmark instances used in [2]. We run our experiments on a computer with an Intel Core i7-1185G7 3.00GHz processor and 32 GB of RAM. We used Gurobi 9.5.2 as MILP solver and imposed a time limit of 10 minutes on each run. The preliminary results show that consideration of the VIs improves the AF's performance, solving instances with up to 10 jobs and 10 tools to optimality.

As further research, we intend to propose new MILP models, VIs, and decomposition methods as exact approaches to solve the SSP-NPM.

References

1. Calmels, D.: The job sequencing and tool switching problem: state-of-the-art literature review, classification, and trends. Int. J. Prod. Res. **57**(15-16), 5005–5025 (2019)
2. Calmels, D.: A comparison of different mathematical models for the job sequencing and tool switching problem with non-identical parallel machines. Int. J. Oper. Res. **45**(4), 419–441 (2022)
3. Calmels, D., Rajendran, C., Ziegler, H.: Heuristics for solving the job sequencing and tool switching problem with non-identical parallel machines. In: Fortz, B., Labbé, M. (eds.) Operations Research Proceedings 2018. pp. 459–465. Springer, Cham (2019)
4. Tang, C., Denardo, E.: Models arising from a flexible manufacturing machine, part I: minimization of the number of tool switches. Oper. Res. **36**(5), 767–777 (1988)