Modular equipment optimization in the design of multi-product reconfigurable manufacturing systems

Abdelkrim R. Yelles-Chaouche[1,2], Evgeny Gurevsky[3],
Nadjib Brahimi[4] and Alexandre Dolgui[2]

[1] IRT Jules Verne, Bouguenais, France
[2] LS2N, IMT Atlantique, Nantes, France
[3] LS2N, Université de Nantes, France
[4] Rennes School of Business, France

Keywords: Reconfigurable manufacturing systems, modularity, ILP, multi-product.

1 Introduction

This paper deals with reconfigurable manufacturing systems (RMS), which are designed for handling multiple products. These latter are manufactured by a fixed number of machines, each has a limited number of emplacements, where modules can be plugged. A module is a physical unit able to perform sequentially a set of tasks. One of the main characteristics of RMS is achieved through the use of modules, thanks to their ability to be easily moved and removed from one machine to another (see Koren et al. (1999)). The modules are activated one by one within a machine. As a consequence, the load of a machine (which can not exceed a predefined cycle time) is calculated as the sum of all the processing times of the tasks assigned to its modules.

As concerns the products, they share the same set of tasks to be executed. This is due to the fact that they belong to the same family. Each product is associated to its precedence constraints. However, the latter and the processing time of the tasks may be different from one product to another.

In this study, an admissible *configuration* refers to the set of tasks of a particular product assigned to a number of modules, which are allocated to machine emplacements while meeting all the aforementioned constraints. In the case where several products are to be manufactured, admissible configurations need to be designed for each of them. Since the studied line is reconfigurable, it is therefore possible to switch from one product configuration to another one by adding, moving or removing the modules.

This context arises an important optimization problem, which consists in designing an admissible configuration for each product, such as the total number of different modules between all these configurations is minimized. To tackle this new problem, an integer linear programming (ILP) model is developed, which is presented in Section 2. The preliminary results are shown and analyzed in Section 3. Finally, conclusion and perspectives are addressed in Section 4.

2 Problem formulation

In this section, the ILP formulation of the studied optimization problem is given. The used notations and variables are introduced below.

Notations:

- V is the set of all tasks;
- W is the set of available machines;
- $m_{\max}$ is the maximum number of modules per machine;

- $r_{\max}$ is the maximum number of tasks per module;
- M is the set of all the modules that could be generated;
- $E = \{1, \ldots, |W| \cdot m_{\max}\}$ is the set of all module emplacements within a configuration;
- $E(k) = \{(k-1)m_{\max}+1, \ldots, km_{\max}\}$ is the set of module emplacements corresponding to the machine $k \in W$
- P is the set of products;
- C is the cycle time;
- $t_i^{(p)}$ is a processing time of the task $i \in V$ for the product $p \in P$;
- $G^{(p)} = (V, A^{(p)})$ is a directed acyclic graph representing the precedence constraints of the product $p \in P$. Here, $A^{(p)}$ is the set of arcs for $G^{(p)}$, where an arc $(i,j) \in A^{(p)}$ means that the task j has to be assigned either to the same module as the task i, or to succeeding ones.

Variables:

- x_{im} is equal to 1 if the task $i \in V$ is assigned to the module $m \in M$, 0 otherwise.
- $y_{me}^{(p)}$ is equal to 1 if the module $m \in M$ is allocated in the emplacement $e \in E$ of the configuration corresponding to the product $p \in P$, 0 otherwise.
- $z_{ime}^{(p)}$ is equal to 1 if the task $i \in V$ is assigned to the module $m \in M$, which is allocated in the emplacement $e \in E$ of the configuration corresponding to the product $p \in P$, 0 otherwise.
- s_m is equal to 1 if the module $m \in M$ is not empty, 0 otherwise.

$$\min \sum_{m \in M} s_m \tag{1}$$

$$1 \leq \sum_{m \in M} x_{im} \leq |P|, \quad \forall i \in V \tag{2}$$

$$\sum_{e \in E} y_{me}^{(p)} \leq 1, \quad \forall m \in M, \quad \forall p \in P \tag{3}$$

$$\sum_{m \in M} y_{me}^{(p)} \leq 1, \quad \forall e \in E, \quad \forall p \in P \tag{4}$$

$$\sum_{m \in M} \sum_{e \in E} z_{ime}^{(p)} = 1, \quad \forall i \in V, \quad \forall p \in P \tag{5}$$

$$x_{im} \leq s_m, \quad \forall i \in V, \quad \forall m \in M \tag{6}$$

$$x_{im} + y_{me}^{(p)} \leq z_{ime}^{(p)} + 1, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{7}$$

$$z_{ime}^{(p)} \leq x_{im}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{8}$$

$$z_{ime}^{(p)} \leq y_{me}^{(p)}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{9}$$

$$\sum_{m \in M} \sum_{e \in E} e \cdot z_{ime}^{(p)} \leq \sum_{m \in M} \sum_{e \in E} e \cdot z_{jme}^{(p)}, \quad \forall (i,j) \in A^{(p)}, \quad \forall p \in P \tag{10}$$

$$\sum_{e \in E(k)} \sum_{m \in M} \sum_{i \in V} t_i^{(p)} \cdot z_{ime}^{(p)} \leq C, \quad \forall k \in W, \quad \forall p \in P \tag{11}$$

$$\sum_{i \in V} x_{im} \leq r_{\max}, \quad \forall m \in M \tag{12}$$

$$\sum_{i \in V} t_i^{(p)} \cdot x_{im} \leq C, \quad \forall m \in M, \quad \forall p \in P \tag{13}$$

$$z_{ime}^{(p)} = 0, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \notin \bigcup_{k \in Q_i^{(p)}} E(k), \quad \forall p \in P \tag{14}$$

$$\left\lceil \frac{|V|}{r_{\max}} \right\rceil \leq \sum_{m \in M} s_m \leq |V| \tag{15}$$

$$s_{m+1} \leq s_m, \quad \forall m \in M \setminus \{|M|\} \tag{16}$$

$$x_{im} \in \{0,1\}, \quad \forall i \in V, \quad \forall e \in E$$

$$y_{me}^{(p)} \in \{0,1\}, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P$$

$$z_{ime}^{(p)} \in \{0,1\}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P$$

$$s_m \in \{0,1\}, \quad \forall m \in M$$

Objective function (1) minimizes the total number of non-empty modules. Constraints (2) state that any task should be assigned to at least one, but at most $|P|$ modules. Constraints (3) express that any module can be assigned to at most one emplacement within a configuration. Whereas (4) state that one emplacement could be occupied by no many than one module. Constraints (5) is used so that all the required tasks are performed in each configuration. Constraints (6) state that a module is not empty if at least one task is assigned to it. Constraints (7), (8) and (9) ensure that the assignment of the task i to the module m forces the allocation of this latter to an emplacement within at least one configuration. This helps the model to consider and allocate only the modules that are not empty. The precedence constraints in each configuration are expressed by inequalities (10). Constraints (11) provide that the cycle time for each machine in any configuration is not exceeded. Similarly, constraints (13) ensure that the sum of the processing time of the tasks assigned to the module do not exceed the cycle time. The maximum number of tasks per module is checked by constraints (12). Constraints (14) induce that the task i can only be allocated to a restricted set of workstations, denoted by the interval $Q_i^{(p)}$, where

$$Q_i^{(p)} = \left[\left\lceil \frac{t_i^{(p)} + \sum\limits_{j \in \mathcal{P}_i^{(p)}} t_j^{(p)}}{C} \right\rceil, |W| + 1 - \left\lceil \frac{t_i^{(p)} + \sum\limits_{j \in \mathcal{S}_i^{(p)}} t_j^{(p)}}{C} \right\rceil \right].$$

Here, $\mathcal{P}_i^{(p)}$ (resp. $\mathcal{S}_i^{(p)}$) represents the set of all predecessors (resp. all successors) of the task i with respect to the precedence graph $G^{(p)}$ corresponding to the product p. Since the objective function consists at minimizing the number of modules, one can notice that the latter can not be greater than the number of tasks (in the case where each module has only one task assigned to it). This is used to improve the upper bound on the number of modules. The lower bound could also be calculated as the ratio of the number of tasks and the maximum number of tasks per module. Thus, upper and lower bounds are expressed in constraints (15). Finally, constraints (16) is used to avoid symmetric solutions, meaning that the module $m + 1$ can be filled, only if the module m is already not empty.

3 Computational results

The ILP model is tested on the basis of 224 instances of $|V| = 20$ provided by Otto et al. (2013). Two products ($|P| = 2$) are considered with $r_{\max} = 2$ and $r_{\max} = 3$. Additionally, for each instance, the resolution `CPU` time is limited to 600 seconds, $C = 1000$,

$$|W| = \max_{p \in P} \left\{ \left\lceil 1.4 \cdot \frac{\sum_{i \in V} t_i^{(p)}}{C} \right\rceil \right\},$$

and

$$m_{\max} = \max_{p \in P} \max \left\{ k \mid \sum_{i=1}^{k} t_{\pi_i}^{(p)} \leq C \right\},$$

where $(\pi_1, \pi_2, \ldots, \pi_{|V|})$ is a permutation of V with respect to the non-decreasing order of their processing times corresponding to the product $p \in P$.

The ILP model is solved using CPLEX 12.9, installed on an 1.90GHz Intel(R) Core(TM) i7-8650U computer with 32 GB RAM. The results are expressed in Table 1, where the first column represents the number of products. The second column displays the value of $r_{\max}$. The third column presents the total number of instances. The number of instances solved to optimality as well as their average `CPU` time are shown in the fourth and last columns, respectively. While the instances for which no optimal solution was found are expressed by their average `GAP` in the fifth column.

Table 1. Summary of computational results for $|P| = 2$.

$\lvert P\rvert$	$r_{\max}$	#`INST`	#`OPT`	Avg. `GAP`, (%)	Avg. `CPU`, (s.)
2	2	224	135	17.76	226.29
	3	224	129	21.88	236.80

We can clearly analyze from Table 1 that, for $|P| = 2$, 60% of the instances were optimally solved regarding $r_{\max} = 2$, versus 58% concerning the case where $r_{\max} = 3$. The instances, which were not optimally solved (89 and 95 instances for $r_{\max} = 2$ and $r_{\max} = 3$, respectively) within the maximum `CPU` time, provide a relatively low average `GAP`. This is due to constraints (15), which significantly reduce the searching space. More detailed results for $|V| = 50$ as well as $|P| = 3$ will be provided and analyzed during the presentation on the conference.

4 Conclusion

The proposed ILP model is a first attempt to address the studied problem. The obtained results are promising, but not satisfactory. Hence, for our future research, we are looking forward to develop specific reduction rules, valid inequalities and decomposition techniques for improving the computational results.

Acknowledgements

This work was financially supported by the IRT PERFORM program, managed by IRT Jules Verne (French Institute in Research and Technology in Advanced Manufacturing).

References

Koren, Y. and Heisel, U. and Jovane, F. and Moriwaki, T. and Pritschow, G. and Ulsoy, G. and Van Brussel, H., 2013, «Reconfigurable manufacturing systems». *In: Dashchenko A.I. (Ed.) Manufacturing Technologies for Machines of the Future.* Springer, Berlin, Heidelberg, pp. 627-665.

Otto, A. and Otto, C. and Scholl, A., 2013, «Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing». *European Journal of Operational Research, 228(1): 33-45.*