Search space reduction in MILP approaches for the robust balancing of transfer lines

Aleksandr Pirogov[1,2], André Rossi[3], Evgeny Gurevsky[4], and Alexandre Dolgui[1]

[1] LS2N, IMT Atlantique, Nantes, France
[2] CENTRAL, Lancaster University, UK
[3] LAMSADE, Université Paris-Dauphine (PSL), France
[4] LS2N, Université de Nantes, France

Keywords: transfer line, balancing, robustness, MILP, model enhancements, assignment intervals.

1 Problem description

This abstract deals with the transfer line balancing problem which consists in distributing n non-preemptive production tasks among m linearly ordered machines linked by a conveyor belt such that the load of any machine does not exceed the fixed cycle time T, and precedence constraints are satisfied. At each machine, the corresponding tasks are allocated to blocks, where the tasks are executed simultaneously. Thus, the working time of the block is equal to the longest processing time among the tasks allocated to it. Up to $r_{\max}$ tasks can be attributed to the same block and at most $b_{\max}$ blocks may be arranged at one machine. The blocks of the same machine are activated sequentially. As a consequence, the load time of the machine is equal to the sum of the working time of its blocks. We denote by $U = \{1, \ldots, mb_{\max}\}$ the set of all possible blocks, and by $U(p) = \{(p-1)b_{\max}+1, \ldots, pb_{\max}\}$ the set of blocks for any machine $p \in W = \{1, \ldots, m\}$.

The nominal processing time of task j is t_j for any $j \in V = \{1, \ldots, n\}$. A given nonempty subset $\widetilde{V} \subseteq V$ of tasks is the set of *uncertain tasks*, *i.e.*, the set of tasks whose processing time may vary and can be larger than t_j. The tasks in $V \setminus \widetilde{V}$ are called *certain tasks* and its processing time remains deterministic.

The set of tasks allocated to blocks for a given number of machines and satisfying the mentioned above constraints forms a so-called *feasible line configuration*. For each such configuration, we study a specific robustness measure, named *stability radius*. It is calculated as the maximum increase of the nominal processing time that may affect any uncertain task without compromising the feasibility of the corresponding line configuration by breaking the cycle time constraint. The problem, denoted by P_1 and studied in this abstract, seeks naturally a line configuration, which possesses the maximal value of its stability radius.

The first mixed-integer linear programming (MILP) formulation of P_1 is due to Pirogov (2019). This MILP is given in the next section. In order to solve it more efficiently, we propose some improvements in that formulation, as well as tighter assignment intervals for the tasks. Indeed, because of the precedence constraints, each task j has an assignment interval $[l_j, u_j]$ of indices of blocks which are available to perform this task.

2 Initial MILP formulation

P_1 has originally been formulated as a MILP on the following decision variables: ρ_1 is the stability radius value to maximize; $x_{j,k}$ is a binary variable that is set to one if and only if the task j is allocated to the block k; y_k is equal to 1 if the block k is not empty and 0, otherwise; $\tau_k \geq 0$ determines the working time of the block k; $\Delta^{(p)}_{\min} \geq 0$ represents the minimal value of the save time among all the blocks arranged at the machine p. The save time is defined only for the blocks having uncertain tasks. It is calculated as the difference between the nominal processing time of the longest uncertain task allocated to it and its working time; a_p is a non-negative variable, which is positive if the machine p processes at least one uncertain task; z_k is set to 1 if an uncertain task is allocated to the block k and 0, otherwise. The central idea of the MILP formulation for P_1 consists in maximizing ρ_1, expressed as the minimum idle time over all the machines that process uncertain tasks.

$$\text{Maximise} \;\; \rho_1$$

$$\sum_{k\in U} x_{j,k} = 1 \qquad \forall j \in V \tag{1}$$

$$\sum_{j\in V} x_{j,k} \leq r_{\max} \qquad \forall k \in U \tag{2}$$

$$x_{j,k} \leq y_k \qquad \forall k \in U,\ \forall j \in V \tag{3}$$

$$y_k \leq \sum_{j\in V} x_{j,k} \qquad \forall k \in U \tag{4}$$

$$y_{k+1} \leq y_k \qquad \forall p \in W,\ \forall k \in U(p) \setminus \{pb_{\max}\} \tag{5}$$

$$t_j \cdot x_{j,k} \leq \tau_k \qquad \forall k \in U,\ \forall j \in V \tag{6}$$

$$\Delta^{(p)}_{\min} \leq T \cdot (2 - y_k - z_k) + \tau_k - t_j \cdot x_{j,k} \qquad \forall p \in W,\ \forall k \in U(p),\ \forall j \in \widetilde{V} \tag{7}$$

$$x_{j,k} \leq z_k \qquad \forall k \in U,\ \forall j \in \widetilde{V} \tag{8}$$

$$\sum_{k\in U(p)} \tau_k \leq T \qquad \forall p \in W \tag{9}$$

$$\sum_{q=k}^{|U|-1} x_{i,q} \leq \sum_{q=k+1}^{|U|} x_{j,q} \qquad \forall (i,j) \in A,\ \forall k \in U \setminus \{mb_{\max}\} \tag{10}$$

$$x_{j,k} \leq a_p \qquad \forall p \in W,\ \forall k \in U(p),\ \forall j \in \widetilde{V} \tag{11}$$

$$\rho_1 \leq T \cdot (2 - a_p) - \sum_{k\in U(p)} \tau_k + \Delta^{(p)}_{\min} \qquad \forall p \in W \tag{12}$$

$$\rho_1 \geq 0$$

$$\Delta^{(p)}_{\min} \geq 0,\ a_p \geq 0 \qquad \forall p \in W$$

$$x_{j,k} \in \{0,1\} \qquad \forall j \in V,\ \forall k \in U$$

$$\tau_k \geq 0,\ z_k \in \{0,1\},\ y_k \in \{0,1\} \qquad \forall k \in U$$

Constraints (1) ensure that each task is allocated to exactly one block. Inequalities (2) enforce that each block contains at most $r_{\max}$ tasks. Any block having at least one assigned task is considered as non-empty, as enforced by (3) and (4). Constraints (5) ensures that block $k+1$ has to be empty if block k is empty. The working time of the block is not less than the processing time of any task allocated to it, as provided by (6). Constraints (7) express the definition of $\Delta^{(p)}_{\min}$. Inequalities (8) ensure that z_k is set to one if block k processes an uncertain task. Constraints (9) state that the load of any machine cannot exceed the cycle time, and the precedence constraints are enforced by inequalities (10), where A is the set of all the pairs of tasks involved in the precedence constraints. Constraints (11) – (12) implies that a_p is strictly positive if machine p has at least one uncertain task, and zero otherwise.

3 Reduction of the assignment interval of tasks

Initially, all the tasks have an assignment interval equal to $[1, mb_{\max}]$, but the precedence constraints can help reducing them, which allows to set some $x_{j,k}$ decision variables to 0 in the MILP formulation of P_1. This is achieved by computing the earliest completion time $\theta_j^{(EC)}$ and the latest starting time $\theta_j^{(LS)}$ of task $j \in V$ with the following induction formula initialized with $\theta_0^{(EC)} = 0$ and $\theta_{n+1}^{(LS)} = m \cdot T$ (tasks 0 and $n+1$ are the dummy start and end of the schedule):

$$\theta_j^{(EC)} = t_j + \max_{q \in P(j)} \max \left\{ \theta_q^{(EC)}, \left(\left\lceil \frac{\theta_q^{(EC)} + t_j}{T} \right\rceil - 1 \right) \cdot T \right\},$$

$$\theta_j^{(LS)} = \min_{q \in S(j)} \min \left\{ \theta_q^{(LS)}, \left(1 + \left\lfloor \frac{\theta_q^{(LS)} - t_j}{T} \right\rfloor \right) \cdot T \right\} - t_j.$$

Here, $P(j)$ (resp. $S(j)$) is the set of direct predecessors (resp. successors) of j in the precedence graph. From $\theta_j^{(EC)}$ and $\theta_j^{(LS)}$, the lower and upper bounds of the assignment interval of task j, denoted by l_j and u_j, can be derived:

$$l_j = \left(\left\lceil \frac{\theta_j^{(EC)}}{T} \right\rceil - 1 \right) \cdot b_{\max} + 1, \quad u_j = \left(1 + \left\lfloor \frac{\theta_j^{(LS)}}{T} \right\rfloor \right) \cdot b_{\max}.$$

Since no task can be assigned to the same block as its predecessors or successors, the first rule is to apply the following formula as long as it brings improvements over current bound values:

$$l_j = \max \left\{ l_j, \max_{q \in P(j)} l_q + 1 \right\}, \quad u_j = \min \left\{ u_j, \min_{q \in S(j)} u_q - 1 \right\}.$$

The second rule is to compute $b_{\max}^{(p)} \leq b_{\max}$, an upper bound on the number of non-empty blocks at machine p. Minimizing $b_{\max}^{(p)}$ permits to find empty blocks and allows to set many decision variables to zero. Because of space limitation, this rule is not presented.

Finally, based on a set of tasks that have to be processed (resp. can be possibly processed) by the machine p, noted as $D(p)$ (resp. $V(p)$), the third rule is to assess the maximum remaining working time of the machine p, for all $p \in W$. If $|D(p)| = r_{\max} \cdot b_{\max}^{(p)}$, then no task in $V(p) \backslash D(p)$ can be assigned to the machine p. If $|D(p)| < r_{\max} \cdot b_{\max}^{(p)}$, then the remaining working time of the machine p is upper bounded by $r_{\max} \cdot T - \sum_{j \in D(p)} t_j$. Hence, any task in $V(p) \backslash D(p)$ whose duration is strictly larger than $r_{\max} \cdot T - \sum_{j \in D(p)} t_j$ should be removed from $V(p)$.

4 Improvement of the MILP formulation

From the previous section, variable $x_{j,k}$ is set to zero for all $j \in V$ and for all $k \notin [l_j, u_j]$. Similarly, if $V(p) \subset \widetilde{V}$, then a_p is set to 1.

The following valid inequalities are added to link the y_k and z_k variables (if a block processes an uncertain task, it should be open):

$$z_k \leq y_k, \ \forall k \in U(p), \ \forall p \in W.$$

Constraints (7) can be reinforced by replacing $T(2 - y_k - z_k)$ with $T(1 - z_k)$. In addition, T can be replaced by the constant $\Delta_{\max}^{(p)}$, which is an upper bound on the save time of any block in the machine p:

$$\Delta_{\max}^{(p)} = \begin{cases} 0, & \text{if } V(p) \cap (V \backslash \widetilde{V}) = V(p) \text{ or } V(p) \cap \widetilde{V} = V(p), \\ 0, & \text{else if } t_{\max}^{(p)} \leq \widetilde{t}_{\min}^{(p)}, \\ t_{\max}^{(p)} - \widetilde{t}_{\min}^{(p)}, & \text{otherwise.} \end{cases}$$

Here, $t_{\max}^{(p)}$ is the maximum processing time among certain tasks that can be processed by machine p, whereas $\widetilde{t}_{\min}^{(p)}$ is the minimum processing time among uncertain tasks that can be processed by machine p: $t_{\max}^{(p)} = \max_{j \in V(p) \cap (V \backslash \widetilde{V})} t_j$ and $\widetilde{t}_{\min}^{(p)} = \min_{j \in V(p) \cap \widetilde{V}} t_j$.

Indeed, when machine p can only process certain tasks, there is no save time, so $\Delta_{\max}^{(p)}$ has to be set to 0. When machine p can only process uncertain tasks (or when certain tasks are shorter than any uncertain task), $\Delta_{\min}^{(p)}$ is zero, so $\Delta_{\max}^{(p)}$ can also be set to 0. In all other cases, the save time is upper bounded by the difference between the longest certain processing time, and the shortest uncertain processing time. Hence, constraints (7) are replaced with:

$$\Delta_{\min}^{(p)} \leq \Delta_{\max}^{(p)} \cdot (1 - z_k) + \tau_k - t_j \cdot x_{j,k}, \quad \forall p \in W, \ \forall k \in U(p), \ \forall j \in \widetilde{V}.$$

Constraints (11) can be strengthened to:

$$\sum_{k \in U(p)} x_{j,k} \leq a_p, \quad \forall p \in W, \ \forall j \in \widetilde{V}.$$

The following constraints state that the processing time of an open block cannot be less than the processing time of the shortest task that can be part of this block.

$$\min_{j \in V(p)} t_j \cdot y_k \leq \tau_k, \ \forall p \in W, \ \forall k \in U(p).$$

The following inequalities declare that if a machine processes an uncertain task, then at least one its block has to accommodate an uncertain task:

$$a_p \leq \sum_{k \in U(p)} z_k, \ \forall p \in W.$$

And finally, all the a_p variables should be declared as binary.

5 Conclusion

When applying the improvements proposed in this paper, 829 instances out of 900 from Pirogov (2019) have been solved to optimality within the time limit of 600 seconds per instance. Originally, only 467 instances were solved to optimality with the initial model. These ideas may be applied to address another problem version, where the stability radius is based on a different metric.

Acknowledgements

This work was partially supported by the council of the French Région «Pays de la Loire».

References

Pirogov A., 2019, «Robust balancing of production lines: MILP models and reduction rules». *Ph.D. Thesis, IMT Atlantique, Nantes, France.*