Mathematical formalisation of the Job-shop problem with power constraints (Bi-objective case)

Notations

The notations used in the problem are as follows:

- H : a large positive number;
- M : set of machines;
- J : set of jobs;
- V : set of all the operations ($|V|=|M|.|J|$);
- i,j : indexes for the different operations ($i=O_i$);
- J_i : job of operation i;
- SO_i : set of sub-operations of operation i;
- k,l : indexes representing the different sub-operations of operations;
- m_i : machine required to process operation i, $m_i \in M$;
- $P_{i,k}$: duration of k^{th} sub-operation of operation i;
- $W_{i,k}$: power required for processing the k^{th} sub-operation of operation i;
- w_{max} : maximum power that must never be exceeded;
- c_{max} : completion date of all operations also called makespan of the schedule;
- $s_{i,k}$: starting time of k^{th} sub-operation of operation i;
- $x_{i,k,j,l}$: binary variable equal to 1 if k^{th} sub-operation of operation i is realised before the l^{th} sub-operation of operation j and equal to 0 otherwise;
- $y_{i,k,j,l}$: binary variable equal to 1 if there is a non-null power flow from operation i to the operation j and equal to 0 otherwise;
- $\varphi_{i,k,j,l}$: denotes the number of power units transferred from the k^{th} sub-operation of operation i to the l^{th} sub-operation of operation j. $\varphi_{0,0,j,l}$ concerns the power units transferred from the source node to other sub-operations;

Mathematical Formalisation

The first line (1) of the mathematical formalisation refers to the objective of the problem which is the minimisation of both the completion time of all operations (makespan) and the power threshold:

$$\begin{aligned} &Min\ c_{max} \\ &Min\ w_{max} \end{aligned} \tag{1}$$

Job-shop model

$$s_{i,|SO_i|} - c_{max} \leq -P_{i,|SO_i|}, \forall i \in V \tag{2}$$

$$x_{i,|SO_i|,j,1} + x_{j,|SO_j|,i,1} = 1, \forall (i,j) \in V, m_i = m_j \quad (3)$$

$$s_{j,1} - s_{i,|SO_i|} \geq P_{i,|SO_i|}, \forall (i,j) \in V, i < j, J_i = J_j \quad (4)$$

$$s_{i,k} - s_{i,k-1} = P_{i,k-1}, \forall i \in V, \forall k \in SO_i, k > 1 \quad (5)$$

$$s_{j,1} - s_{i,|SO_i|} - Hx_{i,|SO_i|,j,1} \geq P_{i,|SO_i|} - H, \forall (i,j) \in V, m_i = m_j \quad (6)$$

The first set of constraints (2) gives the expression of the makespan, which must be greater or equal to the end date of all the operations (as operations are split into sub-operations, the last sub-operation of each operation is considered, hence the use of the cardinality of the subset SO_i). Constraints (3) represent the disjunctions constraints for operations occurring on the same machines. In these constraints, if two operations *i* and *j* of different jobs must be scheduled on the same machine, then *i* is processed before *j*, or *j* is processed before *i*. Constraints (4) define the starting dates of operations of a job according to its sequence of operations. Constraints (5) ensure that, each sub-operations referring to the same operation are processed without delays (i.e. no-wait constraints). Constraints (6) adjust the starting dates of operations that belong to different Jobs but need the same machine, as they cannot be processed simultaneously.

Power consumption model

$$\sum_{j \in V} \sum_{k \in SO_j} \varphi_{0,0,j,k} - w_{max} \leq 0 \quad (7)$$

$$\varphi_{0,0,j,l} + \sum_{i \in V \setminus j} \sum_{k \in SO_i} \varphi_{i,k,j,l} + \sum_{k=1}^{l-1} \varphi_{j,k,j,l} = W_{j,l}, \forall j \in V, \forall l \in SO_j \quad (1)$$

$$\sum_{j \in V \setminus i} \sum_{l \in SO_j} \varphi_{i,k,j,l} + \sum_{l=k+1}^{|SO_i|} \varphi_{i,k,i,l} \leq W_{i,k}, \forall i \in V, \forall k \in SO_i \quad (2)$$

$$\varphi_{i,k,j,l} - Hy_{i,k,j,l} \leq 0, \forall (i,j) \in V, \forall k \in SO_i, \forall l \in SO_j \quad (3)$$

$$y_{i,k,j,l} - \varphi_{i,k,j,l} \leq 0, \forall (i,j) \in V, \forall k \in SO_i, \forall l \in SO_j \quad (4)$$

$$s_{j,l} - s_{i,k} - Hy_{i,k,j,l} \geq P_{i,k} - H, \forall (i,j) \in V, \forall (k,l) \in \left(SO_i, SO_j\right), J_i \neq J_j \quad (5)$$

$$\varphi_{i,k,j,l} = 0, \forall (i,j) \in V, \forall (k,l) \in (SO_i, SO_j), (j < i, J_i = J_j) \vee (i = j, l < k) \quad (6)$$

The constraint (7) avoids to exceed the Power Threshold when processing the operations as it cannot be allocated more power to the operations than w_{max}. Constraints (8) ensure that the sum of power flows from sub-operations and initial power threshold is equal to the power needed for the k^{th} sub-operation of operation *j*. Constraints (9) ensure that the sum of power flows from the considered k^{th} sub-operation of operation *i* to other sub-operations never exceeds the power that was used for its processing. Constraints (10) ensure that if there is a power flow from k^{th} sub-operation of *i* to l^{th} sub-operation of *j*, then $y_{i,k,j,l} = 1$ (flow detection). If $y_{i,k,j,l} = 0$ then no flow is possible from k^{th} sub-operation of i to l^{th} operation of j. Constraints (11) stipulate that if there is no need of a flow from i to j ($\varphi_{i,k,j,l} = 0$), then necessarily $y_{i,k,j,l} = 0$; if $y_{i,k,j,l} = 1$, then $\varphi_{i,k,j,l} \geq 1$. Constraints (12) fix the power flow between sub-operations of an operation according to the power available. Constraints (13) adjust the starting dates of sub-operations which need to wait before the end of previous operations, in order to not exceed the power threshold and receive an power flow from a previously scheduled operation. Constraints (14) stipulate that no flow is possible between two sub-operations i and j, if i and j belong to the same job and if i is processed before j. All these constraints work together to find an exact solution of the problem.