Hybrid flow shop scheduling

The hybrid flow shop scheduling problem (H-FSP) extends the traditional flow shop environment by allowing more than one machine to operate in parallel at each stage. These machines are functionally identical, meaning that each job can be processed by any machine at a given stage with the same processing performance. Consequently, the H-FSP not only involves determining the sequence in which jobs are processed but also deciding which machine at each stage is assigned to each job. This makes H-FSP a joint assignment and sequencing problem, often with the objective of minimizing the makespan—i.e., the time by which all jobs have completed processing on all required stages.

MIP

Objective:

$$\min \quad C_{max} \tag{MIP-HFSP}$$

Subject to:

$$c_{j1} \geq P_{j1}, \quad \forall j \in \mathcal{J} \tag{1}$$

$$c_{ji} \geq c_{ji-1} + P_{ji}, \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \setminus \{1\} \tag{2}$$

$$\sum_{k \in \mathcal{M}_i} w_{jik} = 1, \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{3}$$

$$c_{ji} \geq c_{j'i} + P_{ji} - M(3 - x_{ijj'} - w_{jik} - w_{j'ik}), \quad \forall i \in \mathcal{I}, k \in \mathcal{M}_i, j, j' \in \mathcal{J} : j > j' \tag{4}$$

$$c_{j'i} \geq c_{ji} + P_{j'i} - M(2 + x_{ijj'} - w_{jik} - w_{j'ik}), \quad \forall i \in \mathcal{I}, k \in \mathcal{M}_i, j, j' \in \mathcal{J} : j > j' \tag{5}$$

$$C_{max} \geq c_{ji}, \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{6}$$

$$c_{ji} \geq 0, \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{7}$$

$$w_{jik}, x_{ijj'} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, j, j' \in \mathcal{J} : j > j' \tag{8}$$

Constraint (1) ensures that the completion time of each job j on the first machine is greater than its processing time on the machine. Constraint (2) indicates the completion time of job j on different machines. Constraint (3) ensures every job is assigned to exactly one machine at each stage. Constraints (4) and (5) ensure that no two operations for two jobs j and j' can be processed at the same machine at the same time. Constraint (6) calculates the makespan. Constraints (7) and (8) define the nature of decision variables.

CP Model 1

Objective:

$$\min \quad C_{max} \tag{CP-HFSP 1}$$

Subject to:

$$Task_{ji} = \text{IntervalVar}(P_{ji}), \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{1}$$

$$\text{EndBeforeStart}(Task_{ji}, Task_{ji-1}), \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \setminus \{1\} \tag{2}$$

$$\sum_{j \in \mathcal{J}} \text{Pulse}(Task_{ji}, 1) \leq |\mathcal{M}_i|, \quad \forall i \in \mathcal{I} \tag{3}$$

$$C_{max} = \max_j(\text{EndOf}(Task_{j|\mathcal{I}|})) \tag{4}$$

Constraint (1) defines the interval variable, one for each job at each stage. Constraint (2) ensures that for job j, stage i must start only after stage $i-1$ has finished. In constraint (3), we use function "Pulse" to limit the cardinality of simultaneous jobs being processed at any stage to the number of identical machines in each stage. Constraint (4) is the objective calculation that uses the function "EndOf" over interval variables of jobs at the last stage $|\mathcal{I}|$.

CP Model 2

Objective:

$$\min \quad C_{max} \tag{CP-HFSP 2}$$

Subject to:

$$Task^*_{jik} = \text{IntervalVar}(P_{ji}, \text{Optional}), \quad \forall j \in \mathcal{J}, i \in \mathcal{I}, k \in \mathcal{M}_i \tag{1}$$

$$\text{Alternative}(Task_{ji}, Task^*_{jik} : k \in \mathcal{M}_i), \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{2}$$

$$\text{EndBeforeStart}(Task_{ji}, Task_{ji-1}), \quad \forall j \in \mathcal{J}, i \in \mathcal{I} \setminus \{1\} \tag{3}$$

$$\text{NoOverlap}(Task^*_{jik} : j \in \mathcal{J}), \quad \forall i \in \mathcal{I}, k \in \mathcal{M}_i \tag{4}$$

$$C_{max} = \max_j(\text{EndOf}(Task_{j|\mathcal{I}|})) \tag{5}$$

Constraint (1) defines an optional interval variable for each operation on each machine. Constraint (2) selects one interval variable for each operation at each stage that determines the assignment. Constraint (3) ensures that stage i starts after stage $i-1$ is completed for each job j. Constraint (4) ensures no more than one job can be processed on a machine at a time. Constraint (5) is the objective calculation that uses the function "EndOf" over interval variables of jobs at the last stage $|\mathcal{I}|$.