Flexible Distributed Job Shop Scheduling Problem (FDJSSP)

Feodor Pisnichenko

July 15, 2025

Summary

Problem formulation

Problem Introduction: Objective

The primary objective of the FDJSSP is to **minimize the makespan**, i.e., the completion time of the last operation in the distributed job shop.

- The problem generalizes the classical Job Shop Scheduling Problem (JSSP) and is therefore **NP-hard**.
- For large instances, it becomes intractable. The goal is to develop **efficient heuristic or metaheuristic algorithms** to find high-quality schedules in reasonable time.
- These algorithms must optimize:
  - Assignment of operations to machines.
  - Sequencing of operations on each machine.
  - Scheduling of required data transfers.
- **Practical Relevance:** Distributed computing (e.g., GPU clusters) and multi-site manufacturing.

Problem Introduction: Background

FDJSSP extends the classical JSSP with two key complexities reflecting modern operations:

- **Flexible Machine Assignment:** Each operation can be processed on one of several candidate machines.
- **Distributed Environment:** Machines are distributed across a network. Dependent operations on different machines incur communication delays for data transfer.

Problem Introduction: Background

This introduces additional scheduling considerations:

- **Operation Dependencies:** Operations form a Directed Acyclic Graph (DAG). If $(i_1, i_2) \in B$, then i_2 cannot start until i_1 finishes and its output is available.
- **Communication Delays:** Data transfers between machines take time and must be explicitly scheduled.
- **Exclusivity Constraints:**
  - **Machine Exclusivity:** No two operations run at the same time on the same machine.
  - **Channel Exclusivity:** No two data transfers use the same communication link simultaneously.

Parameters and variables

Sets

- I: set of operations, $|I| = n$.
- J: set of machines, $|J| = m$.
- $A_i \subseteq J$: set of candidate machines for operation $i \in I$.
- $B \subseteq I \times I$: set of precedence edges; $(i_1, i_2) \in B$ means i_1 must finish before i_2 starts.
- $C = \{(j_1, j_2) \in J \times J \mid j_1 \neq j_2\}$: set of directed communication channels.

Parameters

- D_i^o: processing duration of operation i.
- D_{i_1,i_2}^c: communication duration (data transfer time) required if $(i_1, i_2) \in B$ and the operations are on different machines.
- M: a sufficiently large constant (big-M).

Decision Variables

- $s_i, e_i \geq 0$: start time and end time of operation i.
- $x_{i,j} \in \{0, 1\}$: 1 if operation i is assigned to machine j, 0 otherwise.
- $c_{i_1,i_2}, d_{i_1,i_2} \geq 0$: start and end time of communication for precedence $(i_1, i_2) \in B$.
- $z_{i_1,i_2,j_1,j_2} \in \{0, 1\}$: 1 if transfer for $(i_1, i_2) \in B$ uses channel $(j_1, j_2) \in C$.
- $y_{i_1,i_2} \in \{0, 1\}$: 1 if operation i_1 precedes i_2 on their assigned machine.
- $w_{i_1,i_2,i_3,i_4} \in \{0, 1\}$: 1 if communication for (i_1, i_2) precedes that for (i_3, i_4) on the same channel.
- $z \geq 0$: makespan.

MILP problem

Problem Formulation: Objective Function

The objective is to minimize the makespan z.

$$\min z$$

This is subject to a series of constraints that define a valid schedule.

Constraint 1: Makespan

Formulation:

$$z \geq e_i \quad \forall i \in I.$$

Explanation:

- This constraint ensures that the makespan, z, is greater than or equal to the completion time (e_i) of every operation i.
- By minimizing z, the model forces the makespan to be equal to the completion time of the very last operation to finish.

Constraint 2: Operation Duration

Formulation:

$$e_i = s_i + D_i^o \quad \forall i \in I.$$

Explanation:

- This equation defines the end time (e_i) of each operation i.
- It is simply the operation's start time (s_i) plus its fixed processing duration (D_i^o).

Constraint 3: Machine Assignment

Formulation:

$$\sum_{j \in A_i} x_{i,j} = 1 \quad \forall i \in I.$$

Explanation:

- This constraint ensures that each operation i is assigned to exactly one machine j from its set of allowed candidate machines (A_i).
- The binary variable $x_{i,j}$ is 1 if the assignment is made, and 0 otherwise.

Constraint 4: Machine Scheduling (Disjunctive)

Formulation: For any two distinct operations $i_1 \neq i_2$ and any machine $j \in J$:

$$e_{i_1} \leq s_{i_2} + M(3 - y_{i_1,i_2} - x_{i_1,j} - x_{i_2,j})$$

$$e_{i_2} \leq s_{i_1} + M(3 - y_{i_2,i_1} - x_{i_1,j} - x_{i_2,j})$$

Explanation:

- These "big-M" constraints enforce machine exclusivity.
- If two operations i_1 and i_2 are assigned to the same machine j (i.e., $x_{i_1,j} = x_{i_2,j} = 1$), then one must finish before the other starts. The binary variable y determines the sequence.

Constraint 5: Operation Order on Machines

Formulation:

$$y_{i_1,i_2} + y_{i_2,i_1} = 1 \quad \forall i_1 \neq i_2.$$

Explanation:

- This constraint establishes a strict processing order between any two operations, i_1 and i_2.
- If they are scheduled on the same machine, either $y_{i_1,i_2} = 1$ (meaning i_1 is before i_2) or $y_{i_2,i_1} = 1$ (meaning i_2 is before i_1), but not both.

Constraint 6: Channel Selection (Linearization)

Formulation: For each $(i_1, i_2) \in B$ and $(j_1, j_2) \in C$:

$$
\begin{aligned}
z_{i_1,i_2,j_1,j_2} &\leq x_{i_1,j_1}, \\
z_{i_1,i_2,j_1,j_2} &\leq x_{i_2,j_2}, \\
z_{i_1,i_2,j_1,j_2} &\geq x_{i_1,j_1} + x_{i_2,j_2} - 1.
\end{aligned}
$$

Explanation:

- This set of inequalities is a standard way to linearize a product of two binary variables.
- It ensures that $z_{i_1,i_2,j_1,j_2} = 1$ if and only if operation i_1 is on machine j_1 AND operation i_2 is on machine j_2. This activates the communication channel between them.

Constraint 7: Communication Duration

Formulation:

$$d_{i_1,i_2} = c_{i_1,i_2} + D^c_{i_1,i_2} \cdot \sum_{(j_1,j_2)\in C} z_{i_1,i_2,j_1,j_2} \quad \forall (i_1, i_2) \in B.$$

Explanation:

- This defines the end time (d_{i_1,i_2}) of a data transfer.
- If operations i_1 and i_2 are on different machines, the sum term becomes 1, and the transfer takes time $D^c_{i_1,i_2}$.
- If they are on the same machine, the sum is 0, and the transfer duration is zero $(d_{i_1,i_2} = c_{i_1,i_2})$.

Constraint 8: Start After Transfer

Formulation:

$$s_{i_2} \geq d_{i_1,i_2} \quad \forall (i_1, i_2) \in B.$$

Explanation:

- This is a precedence constraint related to data.
- A successor operation (i_2) cannot start until the required data transfer from its predecessor (i_1) is completed (at time d_{i_1,i_2}).

Constraint 9: Transfer After Operation

Formulation:

$$c_{i_1,i_2} \geq e_{i_1} \quad \forall (i_1, i_2) \in B.$$

Explanation:

- This is another data-related precedence constraint.
- The data transfer for a precedence pair (i_1, i_2) cannot begin (at time c_{i_1,i_2}) until the sending operation (i_1) has finished (at time e_{i_1}).

Constraint 10: Channel Scheduling (Data Transfers)

Formulation: For any two distinct edges $(i_1, i_2) \neq (i_3, i_4) \in B$ and any channel $(j_1, j_2) \in C$:

$$c_{i_3,i_4} \geq d_{i_1,i_2} - M(3 - w_{i_1,i_2,i_3,i_4} - z_{i_1,i_2,j_1,j_2} - z_{i_3,i_4,j_1,j_2}).$$

Explanation:

- This is another "big-M" disjunctive constraint, but for communication channels.
- It ensures that if two different data transfers use the same channel, they cannot overlap in time. The binary variable w determines their sequence.

Constraint 11: Order on Channels

Formulation:

$$w_{i_1,i_2,i_3,i_4} + w_{i_3,i_4,i_1,i_2} = 1 \quad \forall (i_1, i_2) \neq (i_3, i_4).$$

Explanation:

- This constraint ensures a strict order for any two data transfers if they happen to share the same communication channel.
- Either transfer (i_1, i_2) precedes (i_3, i_4), or vice-versa.

Constraint 12: Variable Domains

Formulation:

$$s_i, e_i, c_{i_1,i_2}, d_{i_1,i_2}, z \geq 0;$$
$$x_{i,j}, z_{i_1,i_2,j_1,j_2}, y_{i_1,i_2}, w_{i_1,i_2,i_3,i_4} \in \{0,1\}.$$

Explanation:

- These constraints define the domains of all decision variables.
- Time-related variables must be non-negative.
- Assignment and sequencing variables must be binary (0 or 1).

Possible solutions

Possible Solutions

Exact Methods (MIP Solvers)

- The presented Mixed-Integer Programming (MIP) model can be solved using commercial or open-source solvers (e.g., CPLEX, Gurobi, SCIP).
- Exact methods are only feasible for small problem instances.

Heuristics and Metaheuristics

- For practical, large-scale instances, the primary approach is to develop **efficient heuristic or metaheuristic algorithms**.
- These algorithms aim to find high-quality, near-optimal solutions in a reasonable amount of computational time.
- Examples include Graph heuristics, Genetic Algorithms, Tabu Search, Simulated Annealing, etc.

Bibliography

Literature Review

- **Dauzère-Pérès et al. (2024)** – *"The flexible job shop scheduling problem: A review."* European Journal of Operational Research, 314(2), 409–432.
- **Li et al. (2020)** – *"An improved Jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times."* Knowledge-Based Systems, 200.
- **Ren et al. (2021)** – *"Joint optimization for dynamic flexible job-shop scheduling problem with transportation time and resource constraints."* International Journal of Production Research, 60(18).
- **Luo et al. (2022)** – *"A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm."* Expert Systems with Applications, 207:117984.
- **Yuan et al. (2025)** – *"Distributed heterogeneous flexible job-shop scheduling problem considering automated guided vehicle transportation via improved deep Q network."* Swarm and Evolutionary Computation, 94:101902.

URL: `http://github.com/feodorp/lips`

Questions?