Comparing MIP and CP for the Job Shop Scheduling Problem

Analytical Decision Support Systems (MECD03)
Course Teachers: Lu'ıs Gonçalo Rodrigues Reis Figueira, Daniel Augusto Gama de Castro Silva

Julian Nunez Nova   Kirill Savin   Tanzim Hossain

Faculty of Engineering
University of Porto (FEUP)

16 Jan 2026

Motivation & Problem Context

Why Job Shop Scheduling?

- Classical NP-hard combinatorial optimization problem
- Critical in manufacturing and production planning
- Minimize makespan: completion time of last operation

Research Question:

How do Mixed-Integer Programming (MIP) and Constraint Programming (CP) compare in solving the Job Shop Scheduling Problem?

Evaluation Criteria:

- Solution quality (optimality)
- Resolution time
- Scalability and model complexity

Job Shop Scheduling Problem

Problem Components:

- **Jobs** $J = \{1, \dots, n\}$: Each composed of a fixed sequence of operations
- **Machines** $M = \{1, \dots, m\}$: Each operation requires exactly one machine
- **Operations** (j, k): Operation k of job j, with processing time $p_{j,k}$

Constraints:

1. **Precedence:** Operations within a job must follow sequence
2. **Machine capacity:** No machine can process more than one operation simultaneously

Objective:

$$\boxed{\text{Minimize makespan } C_{\max}}$$

Modeling Approaches Overview

Mixed-Integer Programming

- Start-time variables $t_{j,k}$
- Binary sequencing variables y
- Big-M disjunctive constraints
- Solved with CPLEX MIP

Constraint Programming

- Interval variables $O_{j,k}$
- Global constraints
- Strong propagation
- Solved with CP Optimizer

Implementation

Both models implemented in IBM ILOG CPLEX Optimization Studio

Benchmark Dataset: Fisher and Thompson (ft06, ft10)

MIP Formulation: Key Ideas

Decision Variables:

- $t_{j,k} \in \mathbb{Z}_{\geq 0}$: Start time of operation (j, k)
- $y_{j_1,k_1,j_2,k_2} \in \{0, 1\}$: Sequencing variable (1 if (j_1, k_1) precedes (j_2, k_2))
- $C_{\max} \in \mathbb{Z}_{\geq 0}$: Makespan

Key Constraints:

1. **Job precedence:** $t_{j,k+1} \geq t_{j,k} + p_{j,k}$
2. **Big-M disjunction (machine capacity):**

$$t_{j_1,k_1} + p_{j_1,k_1} \leq t_{j_2,k_2} + M(1 - y_{j_1,k_1,j_2,k_2})$$
$$t_{j_2,k_2} + p_{j_2,k_2} \leq t_{j_1,k_1} + M \cdot y_{j_1,k_1,j_2,k_2}$$

3. **Makespan:** $C_{\max} \geq t_{j,m} + p_{j,m}$ for all jobs

where $M = \sum_{j,k} p_{j,k}$ (Big-M constant), $P =$ conflict set (operations on same machine)

CP Formulation: Key Ideas

Decision Variables:

- $O_{j,k}$: Interval variable for operation (j,k) with fixed duration $p_{j,k}$
- $C_{\max} \in \mathbb{Z}_{\geq 0}$: Makespan

Global Constraints:

1. **Job precedence:**

$$\text{endBeforeStart}(O_{j,k}, O_{j,k+1}) \quad \forall j,k$$

2. **Machine capacity (no overlap):**

$$\text{noOverlap}(\{O_{j,k} \mid M_{j,k} = \mu\}) \quad \forall \mu \in M$$

3. **Makespan definition:**

$$C_{\max} \geq \text{endOf}(O_{j,m}) \quad \forall j$$

Advantage: Compact modeling with powerful constraint propagation

Experimental Setup

Benchmark Instances:

- Fisher and Thompson dataset (JSPLIB)
- **ft06:** 6 jobs $\times$ 6 machines
- **ft10:** 10 jobs $\times$ 10 machines

Solvers:

- MIP: CPLEX MIP Solver
- CP: CP Optimizer

Key Performance Indicators (KPIs):

1. Optimal makespan ($C_{\max}$)
2. Total solve time
3. Time to first feasible solution
4. Search effort (nodes/branches explored)

Scalability Note

MIP model exceeded CPLEX Community Edition limits for ft10 due to rapid growth in binary variables and Big-M constraints

Schedule Visualization: Gantt Chart - ft06 (CP Solution)

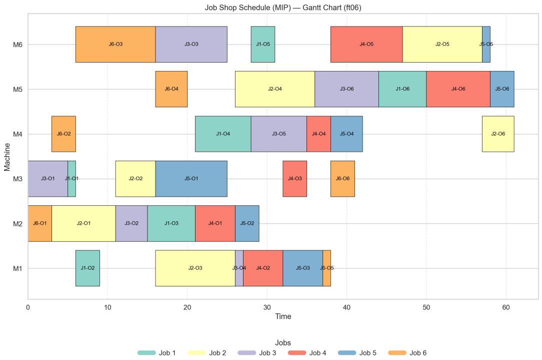

Figure: Gantt chart for ft06 instance (CP solution)

Visual Confirmation:

- No machine conflicts observed
- All job precedence constraints satisfied
- Efficient resource utilization across all machines

Schedule Visualization: Gantt Chart - ft10 (CP Solution)

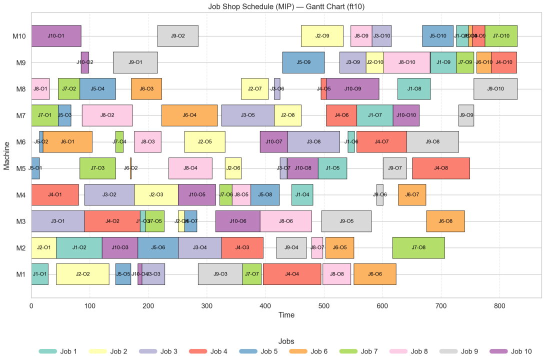

Figure: Gantt chart for ft10 instance (10 jobs $\times$ 10 machines)

Scalability Demonstration:

- CP successfully solves larger instance (ft10)
- Maintains feasibility and optimality
- MIP could not solve this instance due to model size limits
- Demonstrates CP's superior scalability for job shop scheduling

Key Results: ft06 Instance

Table: Performance comparison for ft06 (6 jobs × 6 machines)

Metric	**MIP**	**CP**
Makespan ($C_{\max}$)	61	61
Solve time (s)	0.23	**0.07**
Time to first feasible (s)	0.11	**0.06**
Nodes / Branches	**184**	140,983

Observations:

- Both approaches find the **optimal solution** (makespan = 61)
- CP converges faster: 3× speedup in solve time
- CP finds feasible solutions earlier: 2× faster
- CP explores many more branches, but each is computationally cheaper
- MIP explores fewer nodes, but each requires solving LP relaxation

KPI Analysis: Makespan Comparison

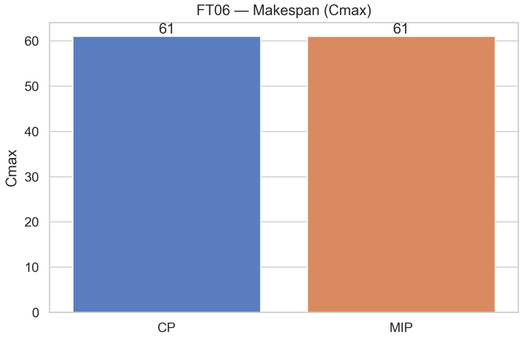

Figure: Makespan comparison for ft06 instance

Key Finding:

- Both MIP and CP achieve **identical optimal makespan = 61**
- Solution quality is equivalent across both modeling paradigms

KPI Analysis: Solve Time Comparison

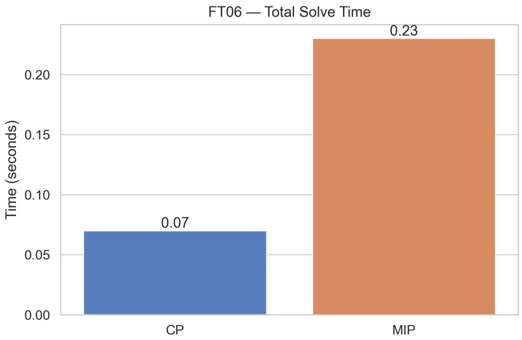

Figure: Solve time comparison for ft06 instance

Key Finding:

- **CP is 3× faster** (0.07s vs 0.23s)
- Constraint propagation enables rapid convergence
- MIP incurs overhead from LP relaxations at each node

KPI Analysis: Model Size & Search Effort

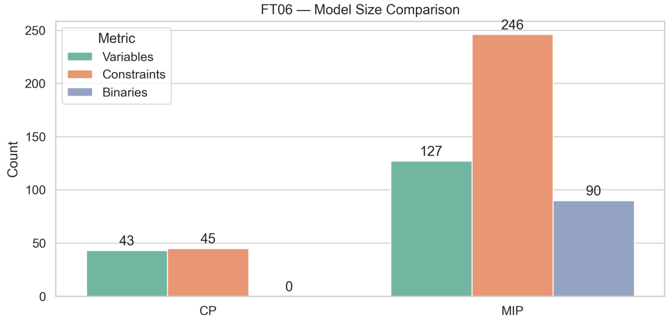

Figure: Model size and search effort comparison for ft06 instance

Key Finding:

- CP explores **140,983 branches** vs MIP's **184 nodes**
- Despite larger search space, CP's lightweight branching is more efficient
- MIP's fewer nodes are computationally expensive (LP solve per node)

Discussion & Insights

Why is CP faster despite exploring more branches?

- **Strong constraint propagation:** Global constraints (e.g., `noOverlap`) prune infeasible schedules early
- **Lightweight branching:** Each CP decision is computationally inexpensive
- **MIP overhead:** Each node requires solving a linear relaxation (costly)

Scalability Comparison:

- MIP fails on ft10 due to model size explosion (binary variables scale quadratically)
- CP remains solvable thanks to compact interval representation

Key Insight

Modeling paradigm significantly impacts problem complexity:
CP distributes complexity across many lightweight decisions; MIP concentrates it into fewer but more expensive steps

Conclusions

Benchmark Comparison Summary:

Metric	MIP (ft06)	CP (ft06)	CP (ft10)
Makespan	61	61	Optimal
Solve time (s)	0.23	**0.07**	Solved
Nodes/Branches	184	140,983	—
Scalability	Failed (ft10)	**Success**	**Success**

Key Findings:

- **CP advantages:** Faster convergence, better scalability, compact modeling
- **MIP advantages:** Fewer nodes, higher cost per node, limited scalability
- Both achieve optimal solutions for small instances
- CP scales to ft10; MIP encounters model size limits

Main Takeaway

Modeling paradigm choice profoundly impacts solver performance. CP leverages global constraints for efficiency; MIP offers flexibility but incurs overhead.

Future Work

Extending the Research:

1. **Alternative MIP formulations:**
   - Time-indexed models
   - Flow-based formulations
   - Tighter linearization techniques
2. **Hybrid MIP/CP approaches:**
   - Use CP for rapid feasibility search
   - Apply MIP to prove optimality
   - Combine strengths of both paradigms
3. **Enhanced solver tuning:**
   - Evaluate different parameter settings
   - Test licensed configurations (beyond Community Edition)
4. **Larger benchmark instances:**
   - Extend tests beyond ft06 and ft10
   - Assess scalability on industrial-sized problems

Thank You

Questions?

Contact:
Julian Nunez Nova, Kirill Savin, Tanzim Hossain
Faculty of Engineering, University of Porto (FEUP)

Code & Data:
GitHub: `github.com/nnovajulian08/saad_project`
IBM ILOG CPLEX Optimization Studio

Project completed as part of Analytical Decision Support Systems course