Solving the Two-Stage Robust Flexible Job-Shop Scheduling Problem

Laurent Houssin

ENAC, Université de Toulouse, France

10-12-2025

Joint work with

Carla Juvin, TBS, Toulouse

Pierre Lopez, LAAS-CNRS, Toulouse

Joint work with

Carla Juvin, TBS, Toulouse

Pierre Lopez, LAAS-CNRS, Toulouse

ELSEVIER

European Journal of Operational Research
Volume 326, Issue 1, 1 October 2025, Pages 54-68

Discrete Optimization

Flow-shop and job-shop robust scheduling problems with budgeted uncertainty

Carla Juvin [a], Laurent Houssin [b c], Pierre Lopez [b]

Outline

Robust optimization background

General idea

- Optimization problems often contain uncertain parameters (e.g. measurement/estimation/implementation errors).
- Find a solution for the optimization problem that is robust against this uncertainty.
- First studies at the end of the 90's.

Robust optimization background

Solving an optimization problem with uncertainty

- Stochastic optimization
  - modelling with random variables
  - quite challenging to solve resulting problems
  - probability distribution have to be determined

Robust optimization background

Solving an optimization problem with uncertainty

- Stochastic optimization
  - modelling with random variables
  - quite challenging to solve resulting problems
  - probability distribution have to be determined
- Robust optimization
  - uncertainty comes from a known set: the uncertainty set
  - no information on probability distribution needed
  - seek for a solution with best worst-case objective guarantee

Nominal value: a good idea?

What if we consider averages?

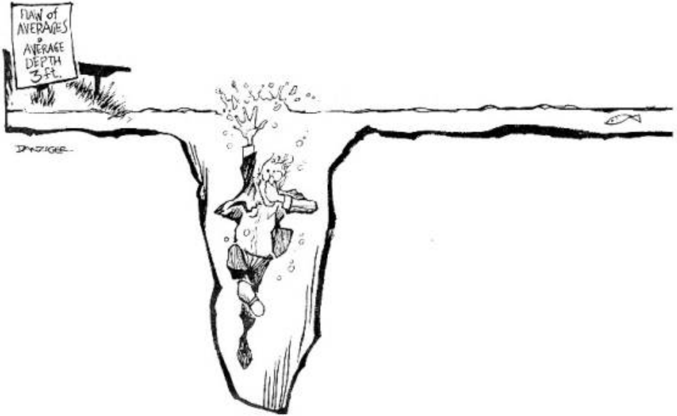
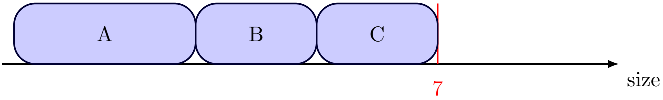


Figure: *The flaw of averages*, S.Savage

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	extended size	utility
A	3	7	12
B	2	3	6
C	2	3	5
D	1	2	5



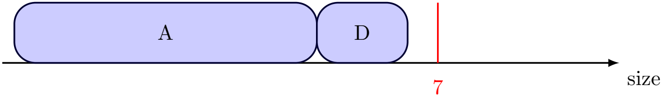
Nominal values

Best solution for nominal values: A-B-C. Utility: 23.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	average size	extended size	utility
A	3	5	7	12
B	2	2.5	3	6
C	2	2.5	3	5
D	1	1.5	2	5



Average values

Best solution for average values: A-D. Utility: 19.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	average size	extended size	utility
A	3	5	7	12
B	2	2.5	3	6
C	2	2.5	3	5
D	1	1.5	2	5

Average values

Best solution for average values: A-D. Utility: 19.

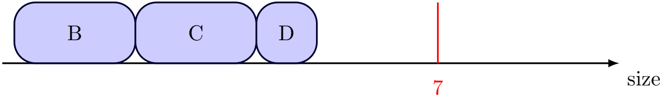
Average values

If probability to get the nominal size was 0.5 and probability to get the extended size was 0.5, the probability of infeasibility for A-D would be 0.5.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	extended size	utility
A	3	7	12
B	2	3	6
C	2	3	5
D	1	2	5



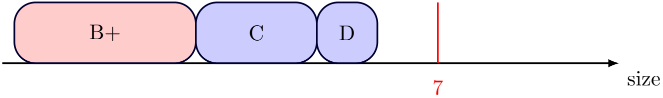
Uncertainty set $\Gamma = 1$

Best solution when 1 deviation is allowed: B-C-D. Utility: 16.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	extended size	utility
A	3	7	12
B	2	3	6
C	2	3	5
D	1	2	5



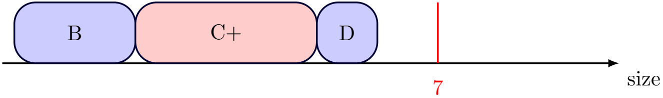
Uncertainty set $\Gamma = 1$

Best solution when 1 deviation is allowed: B-C-D. Utility: 16.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	extended size	utility
A	3	7	12
B	2	3	6
C	2	3	5
D	1	2	5



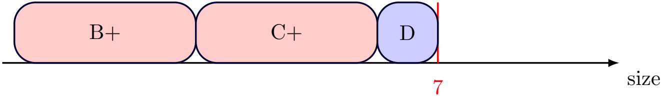
Uncertainty set $\Gamma = 1$

Best solution when 1 deviation is allowed: B-C-D. Utility: 16.

Example: Robust Knapsack

Knapsack size is 7.

	nominal size	extended size	utility
A	3	7	12
B	2	3	6
C	2	3	5
D	1	2	5



Uncertainty set $\Gamma = 1$

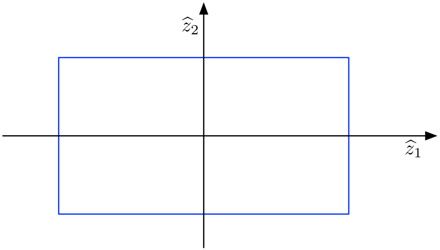
Best solution when 1 deviation is allowed: B-C-D. Utility: 16.

Still the best solution when $\Gamma = 2$.

Uncertainty sets

Three sets can be considered

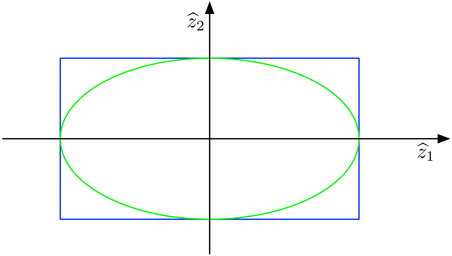
- box uncertainty



Uncertainty sets

Three sets can be considered

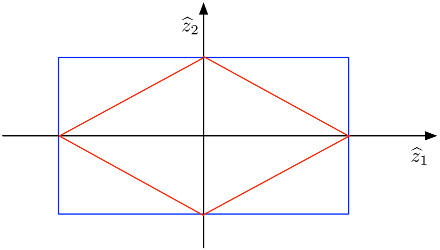
- box uncertainty
- ellipsoidal uncertainty



Uncertainty sets

Three sets can be considered

- box uncertainty
- ellipsoidal uncertainty
- polyhedral uncertainty



Multi stage robust optimization

Sometimes, static robust optimization models can be too conservative but the problem can be formulated as 2-stage robust optimization problems.

$$\begin{aligned} \min \quad & cy + dx & \\ \text{s.t.} \quad & Fy + Gx \leq b(\xi) & \forall \xi \in \Xi \\ & y \in \mathcal{Y}, x \in \mathcal{X} & \end{aligned}$$

Multi stage robust optimization

Sometimes, static robust optimization models can be too conservative but the problem can be formulated as 2-stage robust optimization problems.

$$\begin{aligned} \min \quad & cy + dx \\ \text{s.t.} \quad & Fy + Gx \leq b(\xi) \qquad\qquad \forall \xi \in \Xi \\ & y \in \mathcal{Y}, x \in \mathcal{X} \end{aligned}$$

- first stage: decide $y = \overline{y}$ s.t.

$$\exists x \in \{\mathcal{X} \mid Gx \leq b(\xi) - F\overline{y}, \forall \xi \in \Xi\}$$

- second stage (after uncertainty is revealed) : decide x

Multi stage robust optimization

$$\begin{aligned} \min \quad & cy + dx \\ \text{s.t.} \quad & Fy + Gx \leq b(\xi) && \forall \xi \in \Xi \\ & y \in \mathcal{Y}, x \in \mathcal{X} \end{aligned}$$

Even better!

- first stage: decide $y \in \mathcal{Y}$ that minimizes

$$cy + \max_{\xi \in \Xi} \min_{x \in F(y,\xi)} dx$$

where

$$F(y, \xi) = \{\mathcal{X} \mid Gx \leq b(\xi) - Fy\}$$

- second stage (after uncertainty is revealed): decide x

Robust optimization

Budgeted uncertainty

Γ: number of parameters that can deviate simultaneously from their nominal value

Robust optimization

Budgeted uncertainty

Γ: number of parameters that can deviate simultaneously from their nominal value

Two-stage robust optimization

Some variables can be decided after the realization of uncertainties (adjustable or recourse variables).

Robust optimization

Budgeted uncertainty

Γ: number of parameters that can deviate simultaneously from their nominal value

Two-stage robust optimization

Some variables can be decided after the realization of uncertainties (adjustable or recourse variables).

1: First stage/ Here and now decisions

- Realization of uncertainty

2: Second stage/ wait and see decisions

Outline

Job Shop Scheduling Problem (JSSP)

Definition

Data:

- Set of machines $M = \{1, 2, ...\}$
- Set of n jobs $J = \{1, 2, .., n\}$, each job i consists of n_i operations

Constraints:

- Precedence relation between consecutive operations of the same job.

$$\boxed{O_{i,j}} \rightarrow \boxed{O_{i,j+1}}$$

- A machine can only process a single task at a time.

Job Shop Scheduling Problem (JSSP)

Definition

Data:

- Set of machines $M = \{1, 2, ...\}$
- Set of n jobs $J = \{1, 2, .., n\}$, each job i consists of n_i operations

Constraints:

- Precedence relation between consecutive operations of the same job.

$$\boxed{O_{i,j}} \rightarrow \boxed{O_{i,j+1}}$$

- A machine can only process a single task at a time.

Objective function

- Minimize the total time of the schedule (Makespan): $C_{\max}$

Example – Robust JSSP

		$M1$	$M2$
$J1$	$O_{1,1}$		[7,12]
	$O_{1,2}$	[5,9]	
$J2$	$O_{2,1}$	[3,8]	
	$O_{2,2}$		[4,6]
$J3$	$O_{3,1}$	[10,11]	
	$O_{3,2}$		[10,12]

Example – Robust JSSP

		$M1$	$M2$
$J1$	$O_{1,1}$		[7,12]
	$O_{1,2}$	[5,9]	
$J2$	$O_{2,1}$	[3,8]	
	$O_{2,2}$		[4,6]
$J3$	$O_{3,1}$	[10,11]	
	$O_{3,2}$		[10,12]

Processing times:

	$C_{\max}$		Worst case
	$\Gamma = 0$	$\Gamma = 2$	
S_{nom}	23		

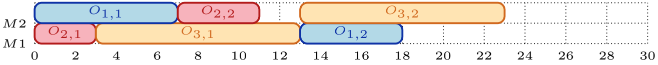


Figure: Nominal solution $\Gamma = 0$ $\quad C_{\max} = 23$

Example – Robust JSSP

		$M1$	$M2$
$J1$	$O_{1,1}$		[7,12]
	$O_{1,2}$	[5,9]	
$J2$	$O_{2,1}$	[3,8]	
	$O_{2,2}$		[4,6]
$J3$	$O_{3,1}$	[10,11]	
	$O_{3,2}$		[10,12]

Processing times:

	$C_{\max}$ $\Gamma = 0$	$C_{\max}$ $\Gamma = 2$	Worst case
S_{nom}	23	30	$\xi_{2,1} = \xi_{3,2} = 1$
S_{rob}			

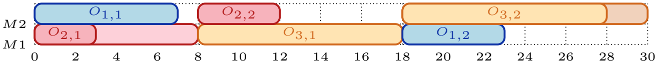


Figure: Nominal solution$\Gamma = 2$ $C_{\max} = 30$

Example – Robust JSSP

		$M1$	$M2$
$J1$	$O_{1,1}$		[7,12]
	$O_{1,2}$	[5,9]	
$J2$	$O_{2,1}$	[3,8]	
	$O_{2,2}$		[4,6]
$J3$	$O_{3,1}$	[10,11]	
	$O_{3,2}$		[10,12]

Processing times:

	$C_{\max}$		Worst case
	$\Gamma = 0$	$\Gamma = 2$	
S_{nom}	23	30	$\xi_{2,1} = \xi_{3,2} = 1$
S_{rob}	24		

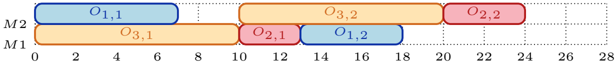


Figure: Robust solution $\Gamma = 0$ $\quad C_{\max} = 24$

Example – Robust JSSP

		$M1$	$M2$
$J1$	$O_{1,1}$		[7,12]
	$O_{1,2}$	[5,9]	
$J2$	$O_{2,1}$	[3,8]	
	$O_{2,2}$		[4,6]
$J3$	$O_{3,1}$	[10,11]	
	$O_{3,2}$		[10,12]

Processing times:

	$C_{\max}$		Worst case
	$\Gamma = 0$	$\Gamma = 2$	
S_{nom}	23	30	$\xi_{2,1} = \xi_{3,2} = 1$
S_{rob}	24	28	$\xi_{1,1} = \xi_{3,2} = 1$

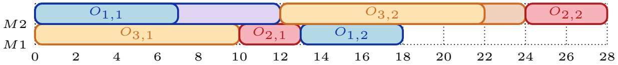


Figure: Robust solution $\Gamma = 2$ $\quad C_{\max} = 28$

Outline

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

first stage : **Determine a sequence of operations** on each machine and the **maximum makespan**

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

first stage : **Determine a sequence of operations** on each machine and the **maximum makespan**

second stage : **Find a start time** for each operation, for each scenario, respecting:

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

first stage : **Determine a sequence of operations** on each machine and the **maximum makespan**

second stage : **Find a start time** for each operation, for each scenario, respecting:

- Precedence constraints between operations of the same job

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

first stage : **Determine a sequence of operations** on each machine and the **maximum makespan**

second stage : **Find a start time** for each operation, for each scenario, respecting:

- Precedence constraints between operations of the same job
- Sequences on the machines

Robust optimization

Two-stage robust job-shop scheduling problem

Objective : minimize the makespan

first stage : **Determine a sequence of operations** on each machine and the **maximum makespan**

second stage : **Find a start time** for each operation, for each scenario, respecting:

- Precedence constraints between operations of the same job
- Sequences on the machines
- The maximum makespan

Extended Models

Mixed integer linear programming

Disjunctive Model:

- Shen, Dauzère-Pérès, and Neufeld 2018

Decision variables:

- $y_{i,j,k,l}$: equal to 1 operation $O_{i,j}$ is processed before operation $O_{k,l}$
- $t_{i,j,\xi}$: start time of operation $O_{i,j}$ in scenario ξ

Extended Models – Robust JSSP

Mixed integer linear programming

$$\min C_{\max}$$

t.q.

$$
\begin{aligned}
C_{\max} &\geq t_{i,n_i}(\xi) + p_{i,n_i}(\xi) && \forall i \in \mathcal{J}, \forall \xi \in \mathcal{U} \\
t_{i,j+1}(\xi) &\geq t_{i',j'}(\xi) + p_{i,j}(\xi) && \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\}, \forall \xi \in \mathcal{U} \\
t_{i',j'}(\xi) &\geq t_{i',j'}(\xi) + p_{i',j'}(\xi) - y_{i,j,i',j'} H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2, \forall \xi \in \mathcal{U} \\
t_{i',j'}(\xi) &\geq t_{i',j'}(\xi) + p_{i,j}(\xi) - (1 - y_{i,j,i',j'}) H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2, \forall \xi \in \mathcal{U} \\
y_{i,j,i',j'} &\in \{0, 1\} && \forall (i, i') \in \mathcal{J}^2, \forall j \in \{1, \dots, n_i\}, \\
& && \forall j' \in \{1, \dots, n_i'\} \\
t_{i',j'}(\xi) &\geq 0 && \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i\}, \forall \xi \in \mathcal{U}
\end{aligned}
$$

Extended Models

Constraint programming

Decision variables:

- $task_{i,j,\xi}$: interval variable between the start and the end of the processing of operation $O_{i,j}$ in scenario ξ
- $seqs_{m,\xi}$: sequence variable of tasks scheduled on machine m in scenario ξ

Extended Models

Constraint programming: *IBM CP Optimizer*

$$\min C_{\max}$$

t.q.

$$\begin{aligned}
C_{\max} \geq EndOf(task_{i,n_i,\xi}) &\qquad \forall i \in \mathcal{J}, \forall \xi \in \mathcal{U} \\
EndBeforeStart(task_{i,j,\xi}, task_{i,j+1,\xi}) &\qquad \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\}, \forall \xi \in \mathcal{U} \\
NoOverlap(seqs_{m,\xi}) &\qquad \forall m \in \mathcal{M}, \forall \xi \in \mathcal{U} \\
SameSequence(seqs_{m,1}, seqs_{m,\xi}) &\qquad \forall m \in \mathcal{M}, \forall \xi \in \mathcal{U} \setminus \{1\}
\end{aligned}$$

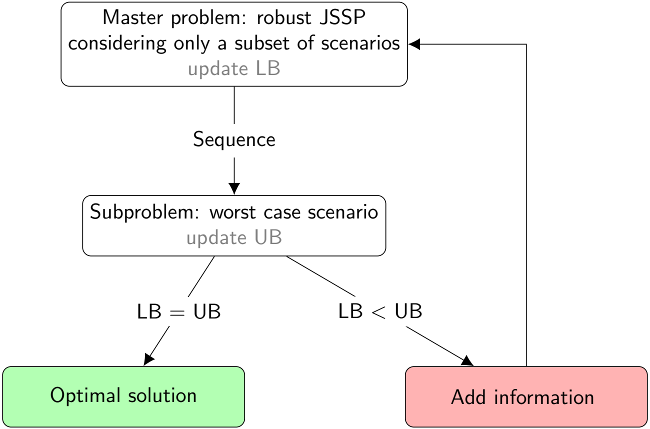
Decomposition methods are often used in robust optimization.

Decomposition methods are often used in robust optimization.

We need:

- a Master Problem
- identify a worst case scenario
- add information to the master problem

Decomposition algorithm



Master problem – MILP

Relaxed version: $S \subseteq \mathcal{U}$

$$\min C_{\max}$$

t.q.

$$\begin{aligned}
C_{\max} &\geq t_{i,n_i}(\xi) + p_{i,n_i}(\xi) && \forall i \in \mathcal{J}, \forall \xi \in \mathcal{S} \\
t_{i,j+1}(\xi) &\geq t_{i',j'}(\xi) + p_{i,j}(\xi) && \forall i \in \mathcal{J}, \forall j \in \{1,\ldots,n_i-1\}, \forall \xi \in \mathcal{S} \\
t_{i',j'}(\xi) &\geq t_{i',j'}(\xi) + p_{i',j'}(\xi) - y_{i,j,i',j'}H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2, \forall \xi \in \mathcal{S} \\
t_{i',j'}(\xi) &\geq t_{i',j'}(\xi) + p_{i,j}(\xi) - (1 - y_{i,j,i',j'})H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2, \forall \xi \in \mathcal{S} \\
y_{i,j,i',j'} &\in \{0,1\} && \forall (i,i') \in \mathcal{J}^2, \forall j \in \{1,\ldots,n_i\}, \\
& && \forall j' \in \{1,\ldots,n_i'\} \\
t_{i',j'}(\xi) &\geq 0 && \forall i \in \mathcal{J}, \forall j \in \{1,\ldots,n_i\}, \forall \xi \in \mathcal{S}
\end{aligned}$$

Master problem - CP

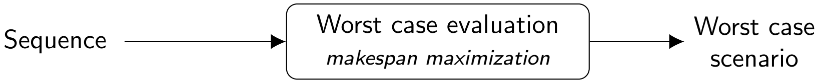
Relaxed version: $\mathsf{S} \subseteq \mathcal{U}$

$$\min C_{\max}$$

s.t.

$$\begin{aligned}
C_{\max} \geq EndOf(task_{i,n_i,\xi}) & \qquad \forall i \in \mathcal{J}, \forall \xi \in \mathcal{S} \\
EndBeforeStart(task_{i,j,\xi}, task_{i,j+1,\xi}) & \qquad \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\}, \forall \xi \in \mathcal{S} \\
NoOverlap(seqs_{m,\xi}) & \qquad \forall m \in \mathcal{M}, \forall \xi \in \mathcal{S} \\
SameSequence(seqs_{m,1}, seqs_{m,\xi}) & \qquad \forall m \in \mathcal{M}, \forall \xi \in \mathcal{S} \setminus \{1\}
\end{aligned}$$

Adversarial subproblem



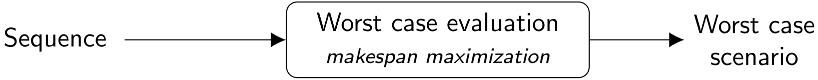
Adversarial subproblem

Sequence → **Worst case evaluation** *makespan maximization* → Worst case scenario

Methods :

- Mixed integer linear programming
- Constraint programming

Adversarial subproblem

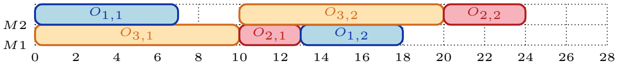


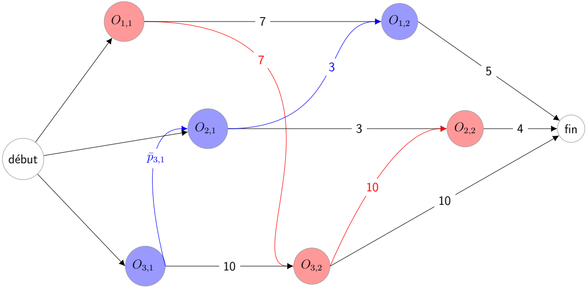
Methods :

- Mixed integer linear programming
- Constraint programming

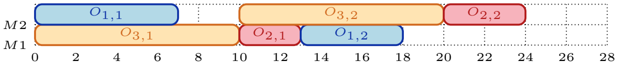
The subproblem can be solved in polynomial time!

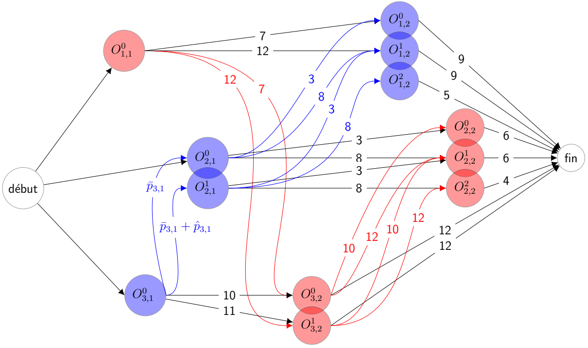
Worst case evaluation





Worst case evaluation





Worst case evaluation

Longest path:

$$\max \sum_{(u,v)\in\mathcal{A}'} a_{u,v} cost_{u,v}$$

s.t.

$$\sum_{(start,u)\in\mathcal{A}'} a_{start,u} = 1$$

$$\sum_{(u,end)\in\mathcal{A}'} a_{u,end} = 1$$

$$\sum_{(u,v)\in\mathcal{A}'} a_{u,v} - \sum_{(v,u)\in\mathcal{A}'} a_{v,u} = 0 \qquad \forall u \in \mathcal{N}'$$

$$a_{u,v} \in \{0,1\} \qquad \forall (u,v) \in \mathcal{A}'$$

Worst case evaluation

Longest path:

$$\max \sum_{(u,v)\in\mathcal{A}'} a_{u,v}\, cost_{u,v}$$

s.t.

$$\sum_{(start,u)\in\mathcal{A}'} a_{start,u} = 1$$

$$\sum_{(u,end)\in\mathcal{A}'} a_{u,end} = 1$$

$$\sum_{(u,v)\in\mathcal{A}'} a_{u,v} - \sum_{(v,u)\in\mathcal{A}'} a_{v,u} = 0 \qquad \forall u \in \mathcal{N}'$$

$$a_{u,v} \geq 0 \qquad \forall (u,v) \in \mathcal{A}'$$

Worst case evaluation

Longest path:

$$\max \sum_{(u,v)\in\mathcal{A}'} a_{u,v}\, cost_{u,v}$$

s.t.

$$\sum_{(start,u)\in\mathcal{A}'} a_{start,u} = 1$$

$$\sum_{(u,end)\in\mathcal{A}'} a_{u,end} = 1$$

$$\sum_{(u,v)\in\mathcal{A}'} a_{u,v} - \sum_{(v,u)\in\mathcal{A}'} a_{v,u} = 0 \qquad \forall u \in \mathcal{N}'$$

$$a_{u,v} \geq 0 \qquad \forall (u,v) \in \mathcal{A}'$$

Dual:

$$\min \; C_{end}$$

s.t.

$$C_u - C_v \geq cost_{u,v} \qquad \forall (u,v) \in \mathcal{A}'$$

$$C_u \geq 0 \qquad \forall u \in \mathcal{N}'$$

Compact model

By combining the dual of the worst case problem with the constraints of the deterministic problem, we can obtain a .

Compact Model

$C^{\gamma}_{i,j}$: worst-case end date of operation $O_{i,j}$ considering no more than γ deviations.

$$
\begin{aligned}
C_{\max} &\geq C^{\Gamma}_{i,n_i} && \forall i \in \mathcal{J} \\
C^{\gamma}_{i,j+1} &\geq C^{\gamma}_{i,j} + \bar{p}_{i,j+1} && \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\}, \forall \gamma \in \{0, \dots, \Gamma\} \\
C^{\gamma+1}_{i,j+1} &\geq C^{\gamma}_{i,j} + \bar{p}_{i,j+1} + \hat{p}_{i,j+1} && \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\}, \forall \gamma \in \{0, \dots, \Gamma - 1\}
\end{aligned}
$$

Adding information

- Logic Benders cuts
- Column and constraint generation with MIP Master
- Column and constraint generation with CP Master

Adding information I

Logic Benders cuts

$$C_{\max} \geq C^{h}_{\max}(1 - NomberOfChanges)$$

Adding information II

Column and constraint generation with MIP Master

Variables:

$$t_{i,j}(\xi_h^*) \quad \forall i \in \mathcal{J}, \forall j \in \{1,\dots,n_i\}$$

Constraints:

$$\begin{aligned}
C_{\max} &\geq t_{i,n_i}(\xi_h^*) + p_{i,n_i}(\xi_h^*) && \forall i \in \mathcal{J} \\
t_{i,j+1}(\xi_h^*) &\geq p_{i,j}(\xi_h^*) + t_{i',j'}(\xi_h^*) && \forall i \in \mathcal{J}, \forall j \in \{1,\dots,n_i - 1\} \\
t_{i',j'}(\xi_h^*) &\geq t_{i',j'}(\xi_h^*) + p_{i',j'}(\xi_h^*) - y_{i,j,i',j'} H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2 \\
t_{i',j'}(\xi_h^*) &\geq t_{i',j'}(\xi_h^*) + p_{i,j}(\xi_h^*) - (1 - y_{i,j,i',j'}) H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}_m^2
\end{aligned}$$

Adding information III

Column and constraint generation with CP Master

Variables:

$$\begin{aligned} & task_{i,j,\xi_h^*} && \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i\} \\ & seqs_{m,\xi_h^*} && \forall m \in \mathcal{M} \end{aligned}$$

Constraints:

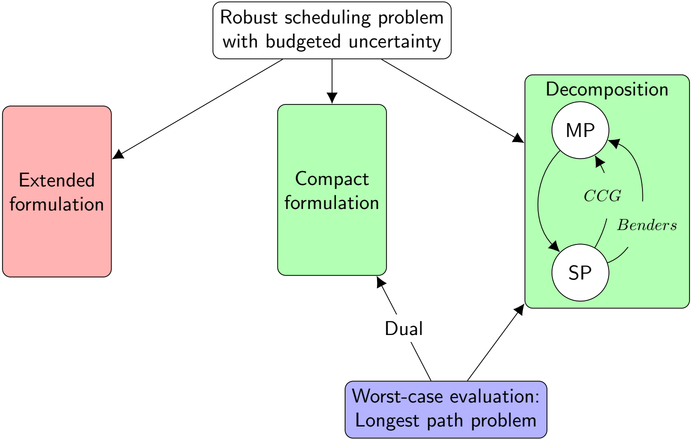
$$\begin{aligned} C_{\max} \geq EndOf(task_{i,|\mathcal{M}|,\xi_h^*}) & \quad \forall i \in \mathcal{J} \\ EndBeforeStart(task_{i,j,\xi_h^*}, task_{i,j,\xi_h^*}) & \quad \forall i \in \mathcal{J}, \forall j \in \{1, \dots, n_i - 1\} \\ NoOverlap(seqs_{m,\xi_h^*}) & \quad \forall m \in \mathcal{M} \\ SameSequence(seqs_{m,1}, seqs_{m,\xi_h^*}) & \quad \forall m \in \mathcal{M} \end{aligned}$$

Solving the problem: Summary

6 Methods

- Extended Models
  - MIP Master
  - CP Master
- Compact model
- Decomposition methods
  - Logic Benders cuts
  - CCG with MIP Master
  - CCG with CP Master

Solving the problem: Summary



Outline is still valid for the flowshop problem.

Outline is still valid for the flowshop problem.

Outline still valid for the flexible JSSP.

Outline is still valid for the flowshop problem.

Outline still valid for the flexible JSSP.

first stage :

Assign each operation to exactly one of the eligible machines

Determine a sequence of operations on each machine and the **maximum makespan**

second stage :

Find a start time for each operation, for each scenario, respecting:

- Precedence constraints between operations of the same job
- Sequences on the machines
- The maximum makespan

Computational experiments

- 90 small instances:
  - from 4 to 9 jobs
  - from 4 to 6 machines
  - nominal duration and deviations randomly generated
- 58 instances from literature:
  - from 6 to 30 jobs
  - from 6 to 15 machines
  - deviations randomly generated
- Uncertainty budget: $\Gamma \in \{5\%, 10\%, 15\%, 20\%\}$ of the number of operations
- Time limit: 1 hour

Results

Small instances:

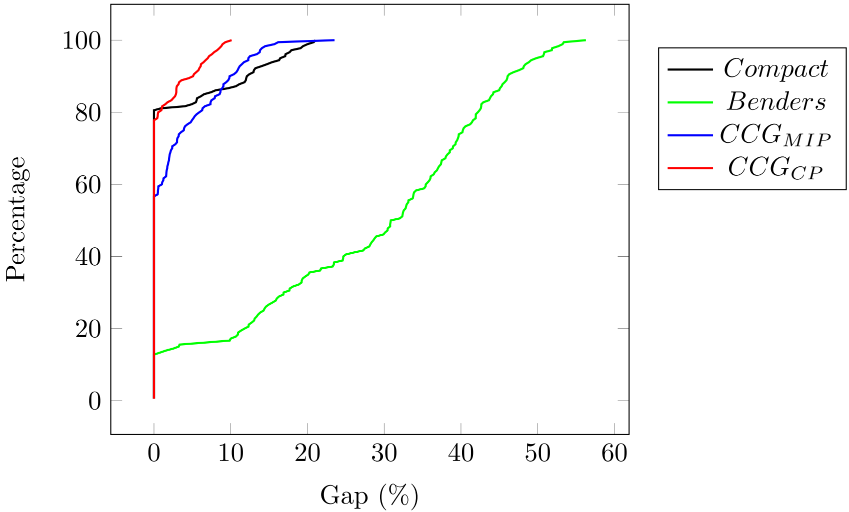
$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{M}\rvert$	*Compact*		*Benders*		CCG_{MIP}		CCG_{CP}	
		#opti.	t (s)	#opti.	t (s)	#opti.	t (s)	#opti.	t (s)
4	4	20	0.19	20	0.43	20	0.27	20	0.14
4	5	20	0.28	20	0.56	20	0.64	20	0.7
4	6	20	0.2	20	0.57	20	0.72	20	0.37
5	4	20	0.42	20	31.55	20	1.85	20	0.66
5	5	20	0.45	19	3.88	20	2.42	20	1.45
5	6	20	0.45	18	3.45	20	9.07	20	18.18
6	4	20	1.09	20	420.08	20	10.45	20	3.76
6	5	20	1.36	20	187.05	20	117.45	20	99.62
6	6	20	2.39	20	461.39	19	318.83	20	314.13
7	4	20	7.01	8	1018.85	19	219.45	20	220.39
7	5	20	8.1	10	430.8	16	229.13	18	126.83
7	6	20	59.44	5	995.98	10	463.2	13	720.04
8	4	18	614.37	0	-	16	199.86	20	32.9
8	5	20	123.93	0	-	8	439.96	14	772.53
8	6	20	314.32	0	-	6	697.24	11	254.55
9	4	6	1051.24	0	-	17	502.85	20	12.23
9	5	9	749.57	0	-	6	766.1	16	131.94
9	6	12	1580.57	0	-	4	187.6	8	80.37

Results

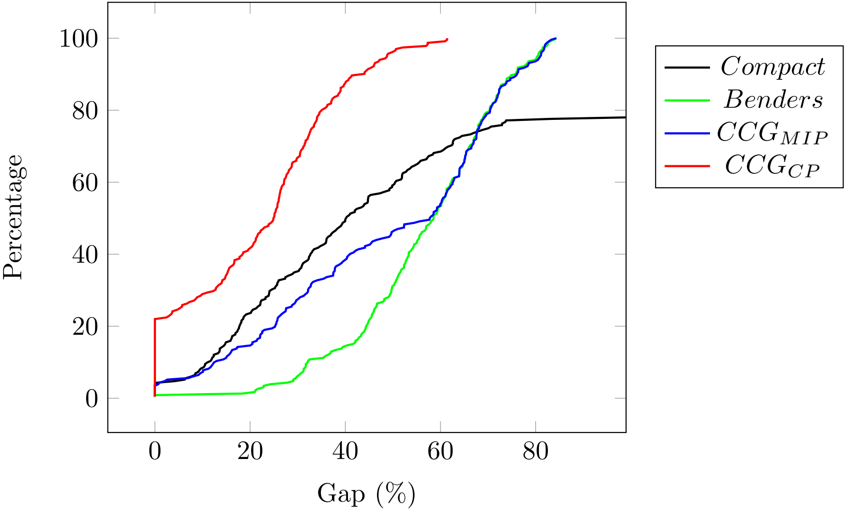
Instances from literature:

$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{M}\rvert$	#	*Compact*		*Benders*		CCG_{MIP}		CCG_{CP}	
			#Best	Gap(%)	#Best	Gap(%)	#Best	Gap(%)	#Best	Gap(%)
6	6	4	4	0	4	0	3	2	4	0
10	5	20	18	15.28	10	48.2	7	11.13	15	7.67
10	10	72	29	21.87	30	52.39	9	27.89	11	24.42
15	5	20	12	44.15	19	66.45	6	57.35	19	21
15	10	20	1	44	16	62.9	1	58.15	2	33.8
15	15	20	0	40.7	15	61.95	3	59.5	2	42.4
20	5	24	2	61.92	16	75.67	4	70.5	22	11.5
20	10	20	0	50.9	4	71.15	1	67.65	15	31.85
20	15	12	0	-	3	66.25	0	65.83	9	47.25
30	10	20	0	-	0	80.25	0	79.35	20	19.05

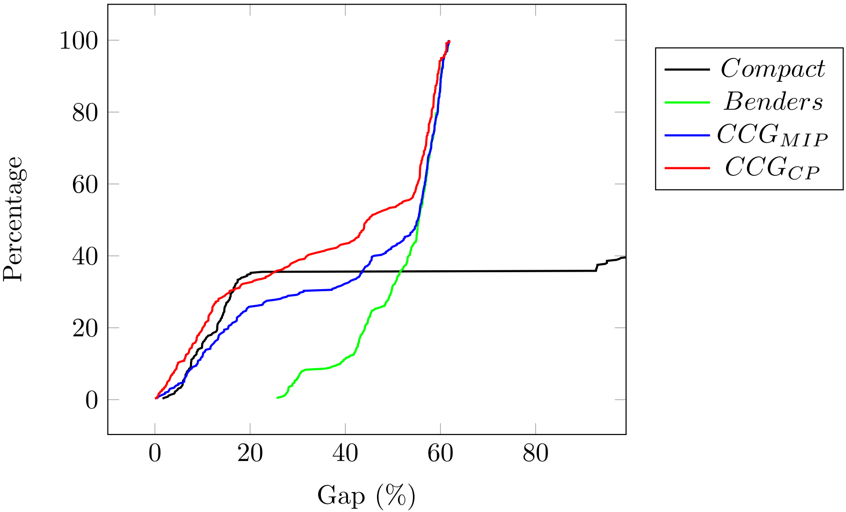
Jobshop: Small and medium instances



Jobshop: Instances from literature



Flowshop: Instances from literature



Acceleration method

In CCG, solving the MP is the most time consuming part.

Acceleration method

In CCG, solving the MP is the most time consuming part.

Operations Research Letters

Volume 51, Issue 1, January 2023, Pages 92-98

An inexact column-and-constraint generation method to solve two-stage robust optimization problems

Man Yiu Tsang, Karmel S. Shehadeh, Frank E. Curtis

Tsang et al. (2023) proposed an inexact CCG but this method does not, in general, converge to the optimal solution.

Acceleration method

In CCG, solving the MP is the most time consuming part.

ELSEVIER

Operations Research Letters

Volume 51, Issue 1, January 2023, Pages 92-98

An inexact column-and-constraint generation method to solve two-stage robust optimization problems

Man Yiu Tsang, Karmel S. Shehadeh, Frank E. Curtis

Tsang et al. (2023) proposed an inexact CCG but this method does not, in general, converge to the optimal solution.

We propose an acceleration method with convergence guarantee to optimal solution.

Acceleration method

Acceleration method

Iteration h of the MP is stopped when

- ◇ optimal solution of MP_h is reached
- ◇ new potential best solution is found[a] and time(MP_h) $\geq$ *threshold*

---

[a] $obj_{MP_h}(\sigma_i) < UB$

Acceleration method

Acceleration method

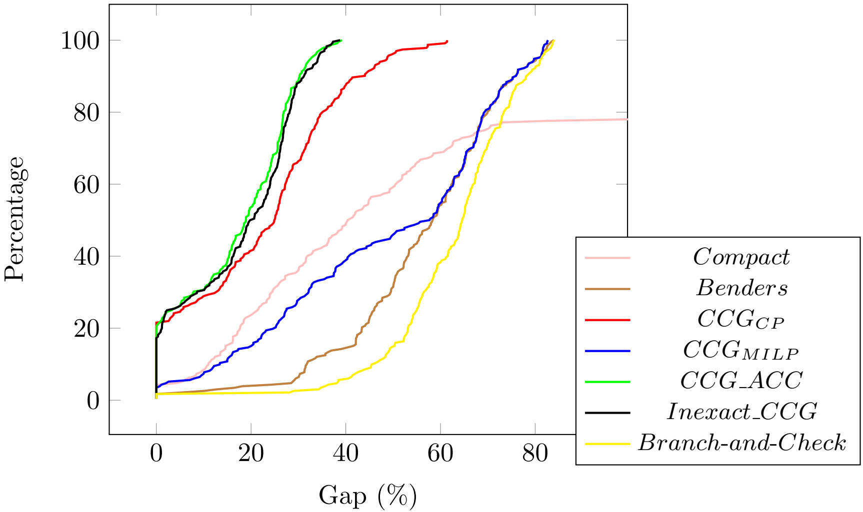
Iteration h of the MP is stopped when

- $\diamond$ optimal solution of MP_h is reached
- $\diamond$ new potential best solution is found[a] and time(MP_h) $\geq$ *threshold*

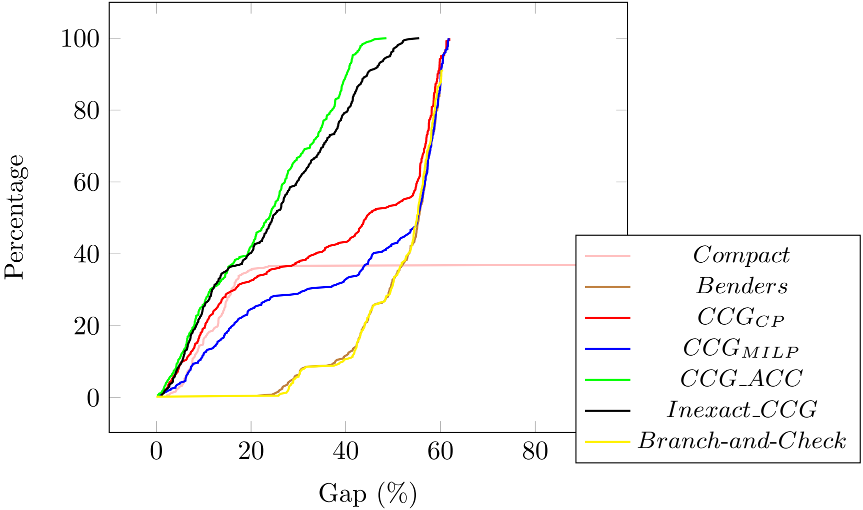
[a] $obj_{MP_h}(\sigma_i) < UB$

Acceleration method converges to an optimal solution with finite steps.

Jobshop: Instances from literature



Flowshop: Instances from literature



Conclusion

Conclusion

- Two-stage jobshop/flowshop scheduling problem with budgeted uncertainty
- Six exact methods

Further works

- Consider new paradigm for the uncertainty set: DRO.
- Extending the hybrid method to other robust scheduling problems

No-wait Flowshop problem

Joint work with R. McGarvey (IESEG, Paris). 

Problem

- Flowshop
- Set-up times
- No-wait constraint (no idle time between elementary operations)
- Minimize makespan

No-wait Flowshop problem

Joint work with R. McGarvey (IESEG, Paris). 

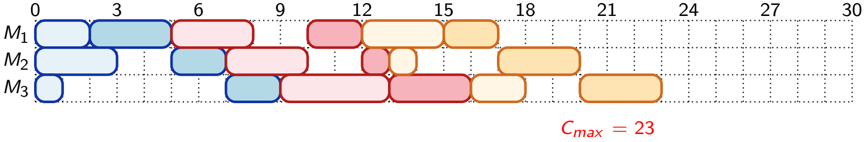
Problem

- Flowshop
- Set-up times
- No-wait constraint (no idle time between elementary operations)
- Minimize makespan

CJSP Example

- Data:
  - 3 Jobs
  - 3 machines

No-wait Flowshop problem



Robust No-wait Flowshop problem

Robust version

Unknown processing times

Robust No-wait Flowshop problem

Robust version

Unknown processing times

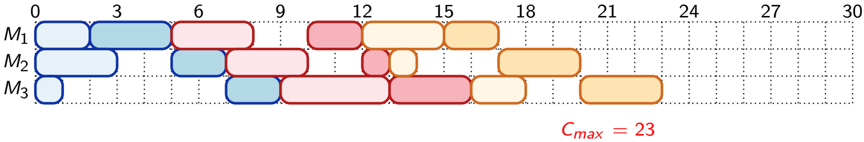
Model as 2-stage robust optimization problem

- First level variable: sequence
- recourse variables: starting time of all tasks

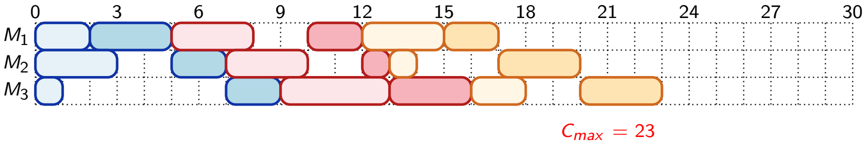
Fun fact

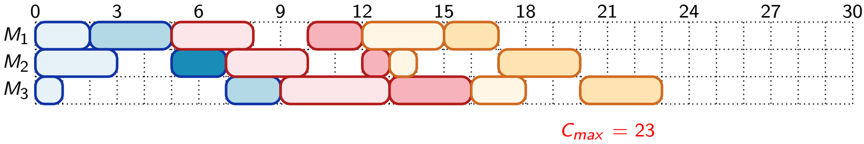
Surprisingly, increasing the duration of a task can lead to a lower makespan!

No-wait Flowshop problem

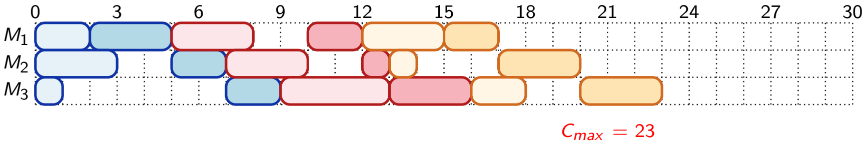


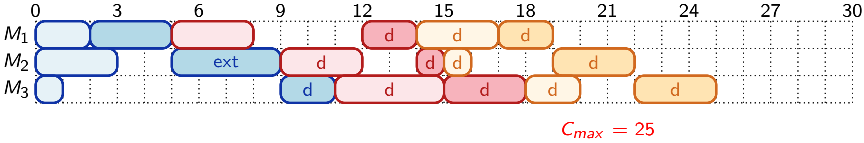
No-wait Flowshop problem



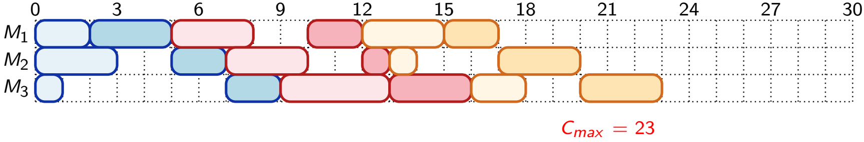


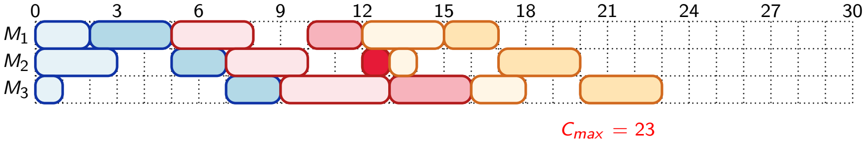
No-wait Flowshop problem



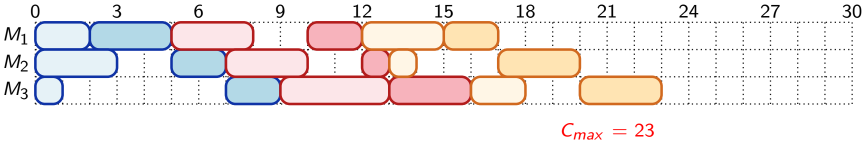


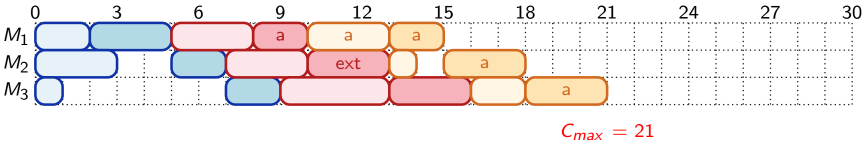
No-wait Flowshop problem





No-wait Flowshop problem





Advertisement: PMS 2026

Advertisement: PMS 2026

IMPORTANT DATES

- Extended abstract submission deadline: ~~November 28, 2025~~ Extended (final) deadline December 15, 2025
- Acceptance notification: January 30, 2026
- Final paper submission deadline: February 13, 2026
- Early bird registration deadline: February 13, 2026

Resources I

Shen, Liji, Stéphane Dauzère-Pérès, and Janis S. Neufeld (2018). "Solving the flexible job shop scheduling problem with sequence-dependent setup times". In: European Journal of Operational Research 265.2, pp. 503–516.

Worst case evaluation - MILP

$$\max \; C_{\max} \tag{1}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{O}_i} \xi_{i,j} = \Gamma \tag{2}$$

$$t_{i,j} - (t_{i',j'} + \bar{p}_{i',j'} + \xi_{i,'j'} \cdot \hat{p}_{i',j'}) \leq H \cdot (1 - b_{i,j,i',j'}) \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i, \forall (i',j') \in A_{i,j} \tag{3}$$

$$\sum_{(i',j') \in A_{i,j}} b_{i,j,i',j'} \geq 1 \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i \mid A_{i,j} \neq \emptyset \tag{4}$$

$$t_{i,1} = 0 \quad \forall i \in \mathcal{J} \mid A_{i,0} = \emptyset \tag{5}$$

$$C_{\max} - (t_{i,n_i} + \bar{p}_{i,n_i} + \xi_{i,n_i} \cdot \hat{p}_{i,n_i}) \leq H \cdot (1 - d_i) \quad \forall i \in \mathcal{J} \tag{6}$$

$$\xi_{i,j} \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i \tag{7}$$

$$b_{i,j,i',j'} \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i, \forall (i',j') \in A_{i,j} \tag{8}$$

$$d_i \in \{0,1\} \quad \forall i \in \mathcal{J} \tag{9}$$

with $A_{i,j}$ the set of immediate predecessors of operation $O_{i,j}$

Worst case evaluation - CP

$$\max C_{\max} task_{i,1}.start = 0 \quad \forall i \in \mathcal{J} \mid A_{i,1} = \emptyset \tag{10}$$

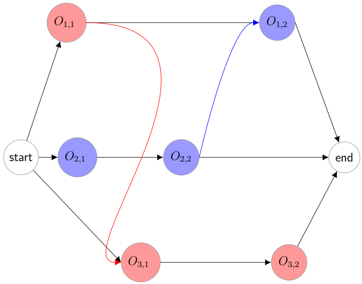
$$task_{i,j}.start = \max(\{task_{i',j'}.end \mid O_{i',j'} \in A_{i,j}\}) \quad \forall i \in \mathcal{J}, \forall O_{i,j} \in \mathcal{O}_i \tag{11}$$

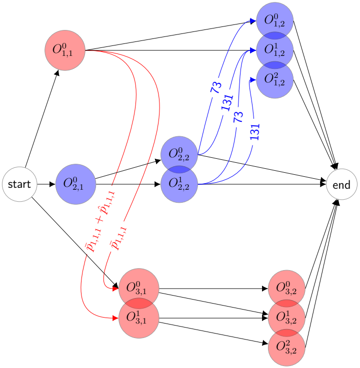
$$StartAtEnd(dev_{i,j}, task_{i,j}) \quad \forall i \in \mathcal{J}, \forall O_{i,j} \in \mathcal{O}_i \tag{12}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{O}_i} HeightAtStart(dev_{i,j}, StepAtStart(dev_{i,j}, 1)) = \Gamma \tag{13}$$

$$C_{\max} = \max(\cup_{\forall i \in \mathcal{J}}(\{task_{i,n_i}.end\} \cup \{dev_{i,n_i}.end\})) \tag{14}$$

Subproblem - Graph





Adding information I

Logic Benders cuts

$$C_{\max} \geq C^h_{\max}(1 - \sum_{(i,j,i',j') \in \mathcal{C}} (1 - y_{i,j,i',j'}))$$

with $\mathcal{D}_h = \{(i,j,i',j') \,|\, \bar{y}^h_{i,j,i',j'} = 1\}$, the set of disjunctions selected at iteration h.

Compact model Jobshop

$$\min C_{\max}$$

s.t.

$$
\begin{aligned}
C_{\max} &\geq C^{\Gamma}_{i,n_i} && \forall i \in \mathcal{J} \\
C^{\gamma}_{i,j+1} &\geq C^{\gamma}_{i,j} + \bar{p}_{i,j+1} && \forall i \in \mathcal{J}, \forall j \in \{1,\dots,n_i-1\},\ \forall \gamma \in \{0,\dots,\Gamma\} \\
C^{\gamma+1}_{i,j+1} &\geq C^{\gamma}_{i,j} + \bar{p}_{i,j+1} + \hat{p}_{i,j+1} && \forall i \in \mathcal{J}, \forall j \in \{1,\dots,n_i-1\}, \forall \gamma \in \{0,\dots,\Gamma-1\} \\
C^{\gamma}_{i,j} &\geq C^{\gamma}_{i',j'} + \bar{p}_{i,j} - y_{i,j,i',j'}H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}^2_m,\ \forall \gamma \in \{0,\dots,\Gamma\} \\
C^{\gamma+1}_{i,j} &\geq C^{\gamma}_{i',j'} + \bar{p}_{i,j} + \hat{p}_{i,j} - y_{i,j,i',j'}H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}^2_m, \forall \gamma \in \{0,\dots,\Gamma-1\} \\
C^{\gamma}_{i',j'} &\geq C^{\gamma}_{i,j} + \hat{p}_{i,j} - (1 - y_{i,j,i',j'})H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}^2_m,\ \forall \gamma \in \{0,\dots,\Gamma\} \\
C^{\gamma+1}_{i',j'} &\geq C^{\gamma}_{i,j} + \hat{p}_{i,j} + \hat{p}_{i,j} - (1 - y_{i,j,i',j'})H && \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i,j'}) \in \mathcal{I}^2_m, \forall \gamma \in \{0,\dots,\Gamma-1\} \\
C^{0}_{i,1} &\geq \bar{p}_{i,1} && \forall i \in \mathcal{J} \\
C^{\gamma}_{i,1} &\geq \bar{p}_{i,1} + \hat{p}_{i,1} && \forall i \in \mathcal{J},\ \forall \gamma \in \{1,\dots,\Gamma\}
\end{aligned}
$$

Robust flowshop – Extended fomulation – MIP

$$\min C_{\max}$$

s.t.

$$\sum_{i=1}^{|\mathcal{J}|} z_{i,l} = 1 \qquad \forall l \in \{1, \dots, |\mathcal{J}|\}$$

$$\sum_{l=1}^{|\mathcal{J}|} z_{i,l} = 1 \qquad \forall i \in \mathcal{J}$$

$$\sum_{i=1}^{|\mathcal{J}|} (p_{i,j}(\xi) z_{i,l+1}) + y_{l+1,m}(\xi) + x_{l+1,m}(\xi) =$$

$$\sum_{i=1}^{|\mathcal{J}|} (p_{i,j+1}(\xi) z_{i,l}) + y_{l,m}(\xi) + x_{l+1,m+1}(\xi) \qquad \forall m \in \mathcal{M} \setminus \{M_{|\mathcal{M}|}\},\ \forall \xi \in \mathcal{S}, \forall l \in \{1, \dots, |\mathcal{J}| - 1\}$$

$$\sum_{m'=1}^{m-1} \sum_{i=1}^{|\mathcal{J}|} (p_{i,j'}(\xi) z_{i,1}) = x_{1,m}(\xi) \qquad \forall m \in \mathcal{M},\ \forall \xi \in \mathcal{S}$$

$$C_{\max} \geq \sum^{|\mathcal{J}|} p_{i,|\mathcal{M}|}(\xi) + \sum^{|\mathcal{J}|} x_{l,|\mathcal{M}|}(\xi) \qquad \forall \xi \in \mathcal{S}$$

JSSP instances from literature

Benchmark	Reference	#Instances	Size	Processing time
`ft6,10,20`	fisher1963probabilistic	3	6×6, 10×10, 20×20	[1,10], [1,99]
`la01-40`	lawrence1984resource	40	10×5, 15×5, 20×5, 10×10, 15×10, 20×10, 30×10, 15×15	[5,99]
`abz5-9`	adams1988shifting	5	10×10, 20×15	[50,100], [25,100], [11,40]
`orb1-10`	applegate1991computational	10	10×10	[5,99]

Table: Characteristics of JSSP instances from literature.

CCG_ACC

Algorithm 2: Acceleration method for column and constraint generation algorithm (CCG_ACC)

Input: $threshold$, $\mathcal{S}^1$

Initialisation:

  $LB \leftarrow 0$, $UB \leftarrow +\infty$, $h \leftarrow 1$

Master problem:

  Start a new search to solve MP_h with $\mathcal{S} = \mathcal{S}^h$

  repeat

    Improve the current MP_h solution σ_i

  until *optimality* ***or*** $(obj_{MP_h}(\sigma_i) < UB$ ***and*** *time*$(MP_h) \geq threshold)$

  $\sigma_h \leftarrow \sigma_i$

  $LB \leftarrow \max(LB, L_{MP_h})$

Subproblem:

  Solve worst-case scenario evaluation for sequence σ_h

  $UB \leftarrow \min(UB, obj^*_{SP}(\sigma_h))$

if $LB < UB$ **then**

  $\mathcal{S}^{h+1} = \mathcal{S}^h \cup \xi^*_h$

  $h \leftarrow h + 1$

  Go back to **Master problem**

else

  Return σ_h