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

Laurent Houssin

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

Angers, 2025

Court CV

- MCF à l'Université Paul Sabatier de 2007 à 2021 (HDR)
- Détachement à l'ISAE-SUPAERO de 2021 à 2024
- Recherche dans l'équipe ROC du LAAS-CNRS

LAAS-CNRS

- UPR du CNRS
- double tutelle des instituts INSIS et INS2I du CNRS
- 4 domaines scientifiques majeurs : l'automatique, l'informatique, la robotique et les micro/nanosciences

- 6 départements, 27 équipes
- $\approx$ 600 personnes

Equipe ROC

- 14 permanents (5 CNRS, 9 EC)
- $\approx$ 15 doctorants
- Responsable : Laurent Houssin

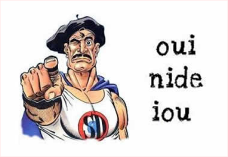
- Recherches sur les problèmes d'optimisation combinatoire et les méthodes algorithmiques pour les résoudre
- A la frontière entre Recherche Opérationnelle et Intelligence Artificielle
- Applications : aérospatiale, transport, production, logistique

Equipe ROC

- 14 permanents (5 CNRS, 9 EC)
- $\approx$ 15 doctorants
- Responsable : Laurent Houssin

- Recherches sur les problèmes d'optimisation combinatoire et les méthodes algorithmiques pour les résoudre
- A la frontière entre Recherche Opérationnelle et Intelligence Artificielle
- Applications : aérospatiale, transport, production, logistique

Poste MCF27 INSA en 2025



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?

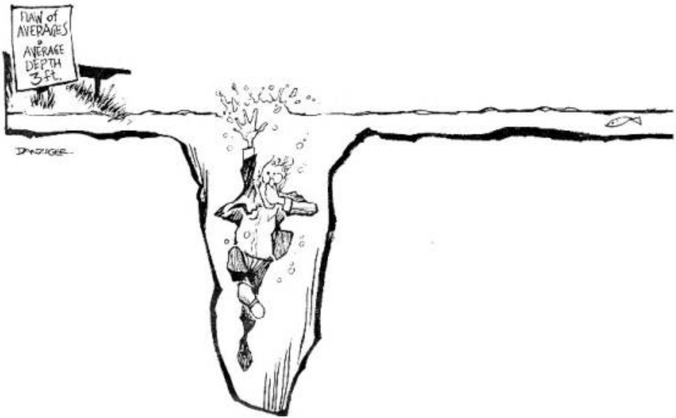


Figure: *The flaw of averages*, S.Savage

Nominal value: a good idea?

What if we consider averages?

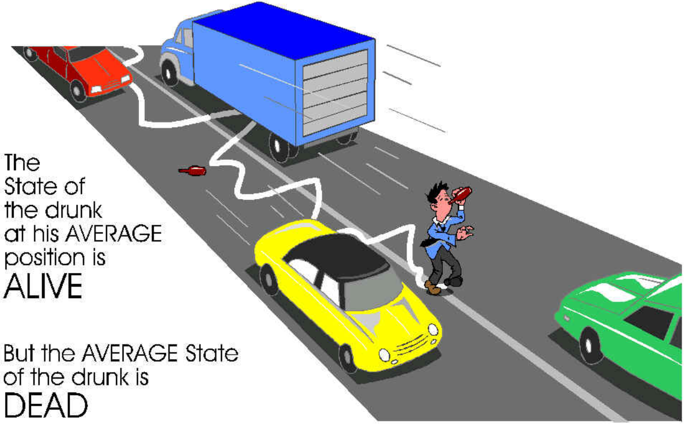


Figure: *The flaw of averages*, S.Savage

Nominal value: a good idea?

What if we consider averages in an optimization problem?

Consider the constraint $ax \leq b$ where a is uncertain $a = \overline{a} + \hat{a}\xi$ and $\xi \in [-1, 1]$ is uniformly distributed.

Nominal value: a good idea?

What if we consider averages in an optimization problem?

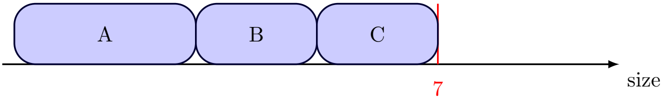
Consider the constraint $ax \leq b$ where a is uncertain $a = \overline{a} + \hat{a}\xi$ and $\xi \in [-1, 1]$ is uniformly distributed.

If the constraint is active for the optimal nominal solution $(a = \overline{a})$.
Infeasible for 50% of the scenarios.

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



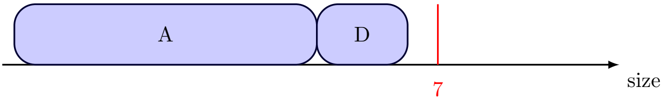
Nominal values

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.

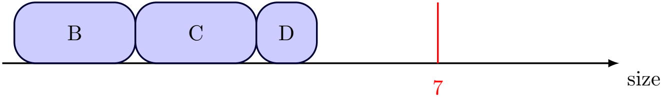
Average values

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



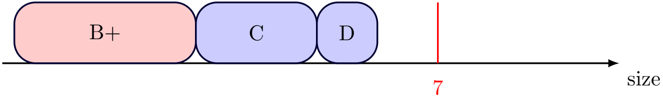
Uncertainty set $\Gamma = 1$

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



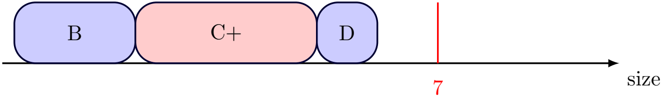
Uncertainty set $\Gamma = 1$

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



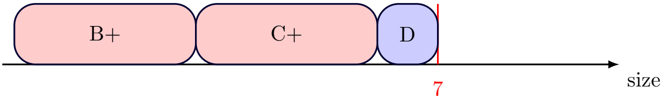
Uncertainty set $\Gamma = 1$

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 $\Gamma = 1$

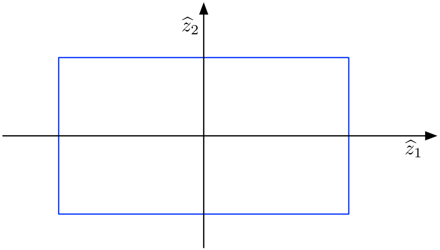
Best solution when 1 deviation is allowed: B-C-D. Utility: 16.

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

Uncertainty set

Three sets can be considered

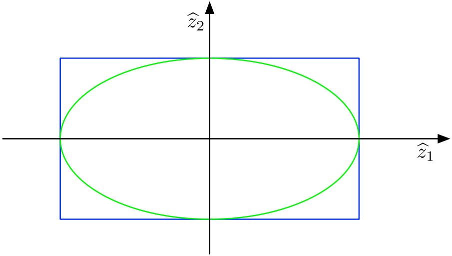
- box uncertainty



Uncertainty set

Three sets can be considered

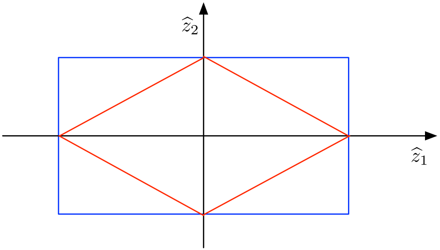
- box uncertainty
- ellipsoidal uncertainty



Uncertainty set

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.

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

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

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

- 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

Outline

Flexible Job Shop Scheduling Problem (FJSSP)

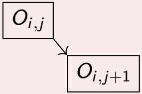
Definition

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.



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

Flexible Job Shop Scheduling Problem (FJSSP)

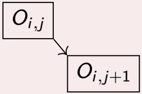
Definition

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.



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

Flexibility

Each operation can be processed on any machine among a subset of eligible machines.

Flexible Job Shop Scheduling Problem (FJSSP)

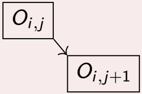
Definition

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.



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

Flexibility

Each operation can be processed on any machine among a subset of eligible machines.

Objective function

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

FJSSP – Example

Processing times:

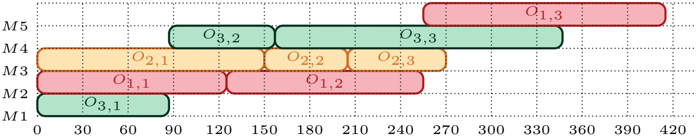
		$M1$	$M2$	$M3$	$M4$	$M5$
$J1$	$O_{1,1}$	117	125			
	$O_{1,2}$		130		140	
	$O_{1,3}$				150	160
$J2$	$O_{2,1}$	214		150		
	$O_{2,2}$		66	55		
	$O_{2,3}$			65		78
$J3$	$O_{3,1}$	87	62			
	$O_{3,2}$			80	70	
	$O_{3,3}$				190	100

FJSSP – Example

Processing times:

		$M1$	$M2$	$M3$	$M4$	$M5$
$J1$	$O_{1,1}$	117	125			
	$O_{1,2}$		130		140	
	$O_{1,3}$				150	160
$J2$	$O_{2,1}$	214		150		
	$O_{2,2}$		66	55		
	$O_{2,3}$			65		78
$J3$	$O_{3,1}$	87	62			
	$O_{3,2}$			80	70	
	$O_{3,3}$				190	100

Optimal solution:



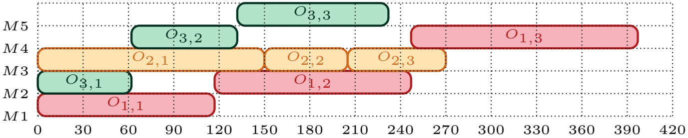
$$C_{max} = 415$$

FJSSP – Example

Processing times:

		M1	M2	M3	M4	M5
J1	$O_{1,1}$	117	125			
	$O_{1,2}$		130		140	
	$O_{1,3}$				150	160
J2	$O_{2,1}$	214		150		
	$O_{2,2}$		66	55		
	$O_{2,3}$			65		78
J3	$O_{3,1}$	87	62			
	$O_{3,2}$			80	70	
	$O_{3,3}$				190	100

Optimal solution:



$$C_{max} = 397$$

Robust optimization

Optimization under uncertainty

Sources of uncertainty: **Processing times variation**, machine breakdown, addition of new operations, etc.

Stochastic optimization :

- probability distribution
- expected objective function

Robust optimization:

- uncertainty set
- worst-case criterion

Outline

Robust optimization

Budgeted uncertainty

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

Robust optimization

Budgeted uncertainty

Γ: number of operations 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 operations 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

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

first stage :

Assign each operation to exactly one of the eligible machines

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

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

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:

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

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

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

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

Robust optimization

Two-stage robust flexible job-shop scheduling problem

Objective : minimize the makespan

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

Example

		$M1$	$M2$	$M3$	$M4$	$M5$
$J1$	$O_{1,1}$	[117-227]	[125-240]			
	$O_{1,2}$		[130-204]		[140-220]	
	$O_{1,3}$				[150-334]	[160-208]
$J2$	$O_{2,1}$	[214-384]		[150-225]		
	$O_{2,2}$		[66-230]	[55-155]		
	$O_{2,3}$			[65-187]		[78-275]
$J3$	$O_{3,1}$	[87-244]	[62-264]			
	$O_{3,2}$			[80-228]	[70-230]	
	$O_{3,3}$				[190-238]	[100-172]

Table: FJSSP example: processing times

Example

	C_{max}		
	$\Gamma = 0$	$\Gamma = 1$	Worst case
$S_{nominal}$	397		

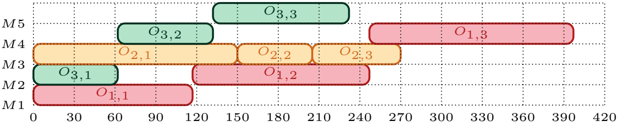


Figure: Optimal solution for $\Gamma = 0$ $C_{max} = 397$

Example

	C_{max}		
	$\Gamma = 0$	$\Gamma = 1$	Worst case
$S_{nominal}$	397	581	$\xi_{1.3} = 1$

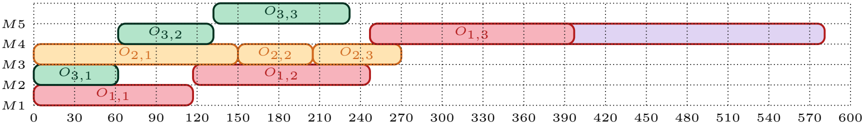


Figure: Worst case $\Gamma = 1$ $C_{max} = 581$

Example

	C_{max}		
	$\Gamma = 0$	$\Gamma = 1$	Worst case
$S_{nominal}$	397	581	$\xi_{1.3} = 1$
S_{robust}	415		

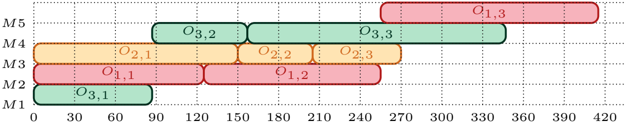


Figure: Robust solution $\Gamma = 0$ $C_{max} = 415$

Example

	C_{max} $\Gamma = 0$	C_{max} $\Gamma = 1$	Worst case
$S_{nominal}$	397	581	$\xi_{1.3} = 1$
S_{robust}	415	530	$\xi_{1.1} = 1$

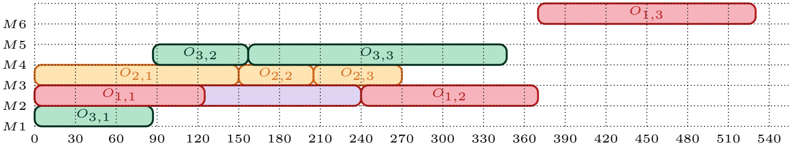


Figure: Robust solution $\Gamma = 1$ $C_{max} \leq 530$

Example

	C_{max}		Worst case
	$\Gamma = 0$	$\Gamma = 1$	
$S_{nominal}$	397	581	$\xi_{1.3} = 1$
S_{robust}	415	530	$\xi_{1.1} = 1$

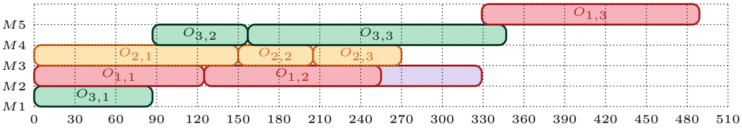


Figure: Robust solution $\Gamma = 1$ $C_{max} \leq 530$

Example

	C_{max}		
	$\Gamma = 0$	$\Gamma = 1$	Worst case
$S_{nominal}$	397	581	$\xi_{1.3} = 1$
S_{robust}	415	530	$\xi_{1.1} = 1$

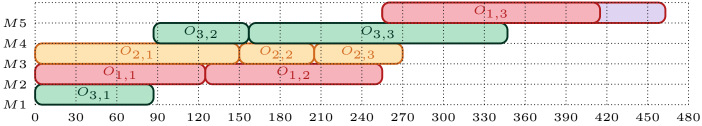


Figure: Robust solution $\Gamma = 1$ $C_{max} \leq 530$

Extended Models

Mixed integer linear programming

Disjunctive Model:

- Shen, Dauzère-Pérès, and Neufeld 2018

Decision variables:

- $x_{i,j,m}$: equal to 1 if operation $O_{i,j}$ is processed on machine m
- $y_{i,j,k,l}$: 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 - MILP

$$\min C_{\max} \quad (1)$$

$$\sum_{m \in M_{i,j}} x_{i,j,m} = 1 \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i \quad (2)$$

$$t_{i,j,\xi} \geq t_{i,j-1,\xi} + \sum_{m \in M_{i,j-1}} x_{i,j-1,m} \times p_{i,j-1,m,\xi} \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i, \xi \in \mathcal{U} \quad (3)$$

$$t_{i,j,\xi} \geq t_{k,l,\xi} + p_{k,l,m,\xi} - (2 - x_{i,j,m} - x_{k,l,m} + y_{i,j,k,l}) \times H \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \in \mathcal{O}_k$$
$$O_{i,j} \neq O_{k,l},\ m \in M_{i,j} \cap M_{k,l},\ \xi \in \mathcal{U} \quad (4)$$

$$t_{k,l,\xi} \geq t_{i,j,\xi} + p_{i,j,m,\xi} - (3 - x_{i,j,m} - x_{k,l,m} - y_{i,j,k,l}) \times H \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \in \mathcal{O}_k$$
$$O_{i,j} \neq O_{k,l},\ m \in M_{i,j} \cap M_{k,l},\ \xi \in \mathcal{U} \quad (5)$$

$$C_{\max} \geq t_{i,n_i,\xi} + \sum_{m \in M_{i,n_i}} x_{i,n_i,m} \times p_{i,n_i,m,\xi} \qquad \forall i \in J,\ \xi \in \mathcal{U} \quad (6)$$

$$x_{i,j,m} \in \{0,1\} \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ m \in M_{i,j} \quad (7)$$

$$y_{i,j,k,l} \in \{0,1\} \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \quad (8)$$

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
- $mode_{i,j,m,\xi}$: interval variable between the start and the end of the processing of operation $O_{i,j}$ on machine m in scenario (optional variable)

Extended Models - CP

$$\text{minimize}\, C_{\max} \tag{9}$$

$$s.t. \quad C_{\max} \geq task_{i,n_i,\xi}.end \quad \forall i \in \mathcal{J},\ \xi \in \mathcal{U} \tag{10}$$

$$Alternative(task_{i,j,\xi}, mode_{i,j,m,\xi} : \forall m \in \mathcal{M}_{i,j}) \quad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ \xi \in \mathcal{U} \tag{11}$$

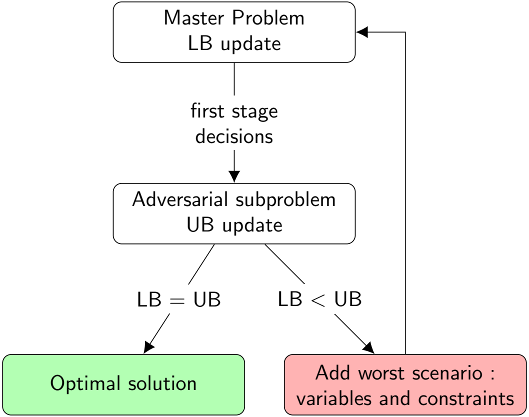
$$EndBeforeStart(task_{i,j,\xi}, task_{i,j+1,\xi}) \quad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i \setminus \{O_{i,n_i}\},\ \xi \in \mathcal{U} \tag{12}$$

$$NoOverlap(seqs_{m,\xi}) \quad \forall m \in \mathcal{M},\ \xi \in \mathcal{U} \tag{13}$$

$$PresenceOf(mode_{i,j,m,0}) \Rightarrow PresenceOf(mode_{i,j,m,\xi}) \quad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ \in \mathcal{M}_{i,j},\ \xi \in \mathcal{U} \setminus \{0\} \tag{14}$$

$$SameSequence(seqs_{m,0}, seqs_{m,\xi}) \quad \forall m \in \mathcal{M},\ \xi \in \mathcal{U} \setminus \{0\} \tag{15}$$

Column and Constraint Generation Algorithm



Master problem- MILP

MILP :

$$\min C_{\max} \quad (16)$$

$$\sum_{m \in M_{i:j}} x_{i,j,m} = 1 \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i \quad (17)$$

$$t_{i,j,\xi} \geq t_{i,j-1,\xi} + \sum_{m \in M_{i,j-1}} x_{i,j-1,m} \times p_{i,j-1,m,\xi} \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i, \xi \in \mathcal{U_S} \subset \mathcal{U} \quad (18)$$

$$t_{i,j,\xi} \geq t_{k,l,\xi} + p_{k,l,m,\xi} - (2 - x_{i,j,m} - x_{k,l,m} + y_{i,j,k,l}) \times H \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \in \mathcal{O}_k \quad O_{i,j} \neq O_{k,l},\ m \in M_{i,j} \cap M_{k,l},\ \xi \in \mathcal{U_S} \subset \mathcal{U} \quad (19)$$

$$t_{k,l,\xi} \geq t_{i,j,\xi} + p_{i,j,m,\xi} - (3 - x_{i,j,m} - x_{k,l,m} - y_{i,j,k,l}) \times H \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \in \mathcal{O}_k \quad O_{i,j} \neq O_{k,l},\ m \in M_{i,j} \cap M_{k,l},\ \xi \in \mathcal{U_S} \subset \mathcal{U} \quad (20)$$

$$C_{\max} \geq t_{i,n_i,\xi} + \sum_{m \in M_{i,n_i}} x_{i,n_i,m} \times p_{i,n_i,m,\xi} \qquad \forall i \in J,\ \xi \in \mathcal{U_S} \subset \mathcal{U} \quad (21)$$

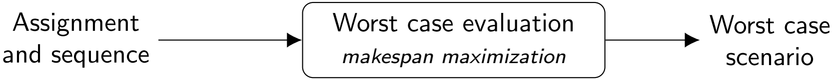
$$x_{i,j,m} \in \{0,1\} \qquad \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ m \in M_{i,j} \quad (22)$$

$$y_{i,j,k,l} \in \{0,1\} \qquad \forall (i,k) \in J \times J, O_{i,j} \in \mathcal{O}_i, O_{k,l} \quad (23)$$

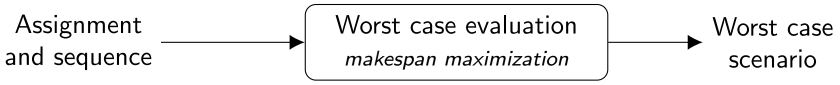
Master problem - CP

$$\begin{aligned}
& \textit{minimize}\, C_{\max} && && (24)\\
\text{s.t.} \quad & C_{\max} \geq task_{i,n_i,\xi}.end && \forall i \in \mathcal{J},\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (25)\\
& EndBeforeStart(task_{i,j,\xi}, task_{i,j+1,\xi}) && \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i \setminus \{O_{i,n_i}\},\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (26)\\
& Alternative(task_{i,j,\xi}, mode_{i,j,m,\xi} : \forall m \in \mathcal{M}_{i,j}) && \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (27)\\
& PresenceOf(mode_{i,j,m,0}) \Rightarrow PresenceOf(mode_{i,j,m,\xi}) && \forall i \in \mathcal{J},\ O_{i,j} \in \mathcal{O}_i,\ \in \mathcal{M}_{i,j},\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (28)\\
& SameSequence(seqs_{m,0}, seqs_{m,\xi}) && \forall m \in \mathcal{M},\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (29)\\
& NoOverlap(seqs_{m,\xi}) && \forall m \in \mathcal{M},\ \xi \in \mathcal{U}_{\mathcal{S}} \subset \mathcal{U} && (30)
\end{aligned}$$

Adversarial subproblem



Adversarial subproblem



Methods :

- Mixed integer linear programming
- Constraint programming

Computational experiments

Methodology :

- Six exact methods:
  - Two extended models: MILP and CP
  - Four decomposition methods: column and constraint generation algorithm mixing both MILP and CP
- FJSSP instances from literature : (Fattahi, Mehrabad, and Jolai 2007)
  - 2 to 12 jobs
  - 4 to 48 operations
  - 2 to 8 machines
  - Deviations randomly generated
  - 60 instances
- Uncertainty budget: $\Gamma \in \{20\%, 40\%, 60\%, 80\%\}$ of the number of operations
- Time limit : 1 hour

Results

Uncertainty budget	extended		column & constraint generation *(MP/SP)*			
	MILP	CP	MILP/MILP	MILP/CP	CP/CP	CP/MILP
20 %	35	37	38	38	42	42
40 %	30	31	39	39	45	48
60 %	30	30	42	43	49	49
80 %	35	35	48	46	54	54
Total	**130**	**133**	**167**	**166**	**190**	**193**

Table: Methods performance comparison grouping by uncertainty budget (number of instances, out of 60, solved to optimality).

Conclusion

Conclusion

- Two-stage flexible job-shop scheduling problem with budgeted uncertainty
- Six exact methods:
  - Two extended models: MILP and CP
  - Four decomposition methods: column and constraint generation algorithm mixing both MILP and CP

Further works

- Solving the subproblem as a longest path problem in an acyclic graph
- Extending the hybrid method to other robust scheduling problems

Resources I

- Bougeret, Marin, Artur Alves Pessoa, and Michael Poss (2019). "Robust scheduling with budgeted uncertainty". In: Discrete applied mathematics 261, pp. 93–107.
- Fattahi, Parviz, Mohammad Saidi Mehrabad, and Fariborz Jolai (2007). "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems". In: Journal of Intelligent Manufacturing 18.3, pp. 331–342.
- Hamaz, Idir, Laurent Houssin, and Sonia Cafieri (2022). The robust cyclic job shop problem.
- Levorato, Mario, Rosa Figueiredo, and Yuri Frota (2022). "Exact solutions for the two-machine robust flow shop with budgeted uncertainty". In: European Journal of Operational Research 300.1, pp. 46–57.

Resources II

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.

Table: Characteristics of FJSSP instances.

Instance	#Jobs	#Machines	#Operations
SFJS1	2	2	4
SFJS2	2	2	4
SFJS3	3	2	6
SFJS4	3	2	6
SFJS5	3	2	6
SFJS6	3	3	9
SFJS7	3	5	9
SFJS8	3	4	9
SFJS9	3	3	9
SFJS10	4	5	12
MFJS1	5	6	15
MFJS2	5	7	15
MFJS3	6	7	18
MFJS4	7	7	21
MFJS5	7	7	21
MFJS6	8	7	24
MFJS7	8	7	32
MFJS8	9	8	36
MFJS9	11	8	44
MFJS10	12	8	48

Instance	milp		cp		ccg_milp		ccg_milp_cp		ccg_cp		ccg_cp_milp	
	#Solv.	t(s)	#Solv.	t(s)	#Solv.	t(s)	#Solv.	t(s)	#Solv.	t(s)	#Solv.	t(s)
SFJS1	12	0.04	12	0.03	12	0.06	12	0.98	12	1.13	12	0.05
SFJS2	12	0.03	12	0.03	12	0.03	12	0.25	12	0.38	12	0.04
SFJS3	12	0.4	12	0.08	12	0.5	12	3.03	12	3.17	12	0.11
SFJS4	12	0.25	12	0.05	12	0.23	12	1.65	12	1.77	12	0.08
SFJS5	12	0.83	12	0.12	12	1.03	12	2.93	12	3.8	12	0.2
SFJS6	12	4.55	12	0.44	12	1.49	12	5.62	12	4.89	12	0.25
SFJS7	12	1.3	12	0.25	12	0.31	12	1.46	12	1.73	12	0.11
SFJS8	12	7.32	12	0.62	12	1.79	12	5.68	12	6.22	12	0.31
SFJS9	12	11.5	12	2.05	12	1.42	12	2.47	12	3.19	12	0.16
SFJS10	12	128	12	3.1	12	2.01	12	8.59	12	11.5	12	0.2
MFJS1	6	1115	6	1445	12	257	12	297	12	73.1	12	19.9
MFJS2	4	1246	6	858	12	161	12	199	12	44.6	12	9.55
MFJS3	0	-	1	1301	11	549	10	414	11	174	12	99.5
MFJS4	0	-	0	-	5	849	5	1715	11	539	11	450
MFJS5	0	-	0	-	5	694	6	1089	10	544	10	328
MFJS6	0	-	0	-	2	1643	1	2522	7	255	9	739
MFJS7	0	-	0	-	0	-	0	-	4	1006	4	815
MFJS8	0	-	0	-	0	-	0	-	3	1144	3	880
MFJS9	0	-	0	-	0	-	0	-	0	-	0	-
MFJS10	0	-	0	-	0	-	0	-	0	-	0	-

Literature

Robust scheduling under uncertainty budget

- Marin Bougeret, Artur Alves Pessoa, and Michael Poss (2019). "Robust scheduling with budgeted uncertainty". In: Discrete applied mathematics 261, pp. 93–107
- Mario Levorato, Rosa Figueiredo, and Yuri Frota (2022). "Exact solutions for the two-machine robust flow shop with budgeted uncertainty". In: European Journal of Operational Research 300.1, pp. 46–57
- Idir Hamaz, Laurent Houssin, and Sonia Cafieri (2022). The robust cyclic job shop problem.

Worst case evaluation - MILP

$$\max \; C_{\max} \tag{31}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{O}_i} \xi_{i,j} = \Gamma \tag{32}$$

$$t_{i,j} - (t_{i',j'} + \bar{p}_{i',j'} + \xi_{i,'j'} \cdot \hat{p}_{i',j'}) \leq H \cdot (1 - b_{i,j,i',j'}) \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i, \; \forall (i',j') \in A_{i,j} \tag{33}$$

$$\sum_{(i',j') \in A_{i,j}} b_{i,j,i',j'} \geq 1 \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i \mid A_{i,j} \neq \emptyset \tag{34}$$

$$t_{i,1} = 0 \quad \forall i \in \mathcal{J} \mid A_{i,0} = \emptyset \tag{35}$$

$$C_{\max} - (t_{i,n_i} + \bar{p}_{i,n_i} + \xi_{i,n_i} \cdot \hat{p}_{i,n_i}) \leq H \cdot (1 - d_i) \quad \forall i \in \mathcal{J} \tag{36}$$

$$\xi_{i,j} \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i \tag{37}$$

$$b_{i,j,i',j'} \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{O}_i, \forall (i',j') \in A_{i,j} \tag{38}$$

$$d_i \in \{0,1\} \quad \forall i \in \mathcal{J} \tag{39}$$

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

Worst case evaluation - CP

$$\max C_{\max} \mathit{task}_{i,1}.\mathit{start} = 0 \quad \forall i \in \mathcal{J} \mid A_{i,1} = \emptyset \tag{40}$$

$$\mathit{task}_{i,j}.\mathit{start} = \max(\{\mathit{task}_{i',j'}.\mathit{end} \mid O_{i',j'} \in A_{i,j}\}) \quad \forall i \in \mathcal{J}, \forall O_{i,j} \in \mathcal{O}_i \tag{41}$$

$$\mathit{StartAtEnd}(\mathit{dev}_{i,j}, \mathit{task}_{i,j}) \quad \forall i \in \mathcal{J}, \forall O_{i,j} \in \mathcal{O}_i \tag{42}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{O}_i} \mathit{HeightAtStart}(\mathit{dev}_{i,j}, \mathit{StepAtStart}(\mathit{dev}_{i,j}, 1)) = \Gamma \tag{43}$$

$$C_{\max} = \max(\cup_{\forall i \in \mathcal{J}}(\{\mathit{task}_{i,n_i}.\mathit{end}\} \cup \{\mathit{dev}_{i,n_i}.\mathit{end}\})) \tag{44}$$

Subproblem - Graph