



**HAL**  
open science

## Two-stage approaches to solving the robust job-shop problem with uncertainty budget

Carla Juvin, Laurent Houssin, Pierre Lopez

### ► To cite this version:

Carla Juvin, Laurent Houssin, Pierre Lopez. Two-stage approaches to solving the robust job-shop problem with uncertainty budget. 19th International Workshop on Project Management and Scheduling (PMS 2024), Apr 2024, Bern, Switzerland. hal-04675753

**HAL Id: hal-04675753**

**<https://laas.hal.science/hal-04675753>**

Submitted on 22 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Two-stage approaches to solving the robust job-shop problem with uncertainty budget

Carla Juvin<sup>1,2</sup>, Laurent Houssin<sup>1,3</sup>, Pierre Lopez<sup>1</sup>

<sup>1</sup> LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

{carla.juvin,pierre.lopez}@laas.fr

<sup>2</sup> ENAC, Université de Toulouse, France

<sup>3</sup> ISAE-SUPAERO, Université de Toulouse, France

laurent.houssin@isae-supaero.fr

**Keywords:** Job-Shop Scheduling, Robust Optimization, Uncertainty Budget, Mixed Integer Linear Programming, Constraint Programming.

## 1 Introduction

The job-shop scheduling problem (JSSP) is a well studied and NP-hard problem where a set of jobs are to be processed on a set of machines. Each job is composed of a sequence of operations that must be processed on machines with given processing times in a given job-dependent order, and each machine can process only one operation at a time. The JSSP has received considerable attention and both metaheuristics and exact methods have been developed to solve the problem, the majority of them with the assumption that the parameters are deterministically known. However, in the real world, many sources of uncertainty can affect the quality and even the feasibility of a schedule.

There exist two major approaches to deal with data uncertainty: stochastic optimization and robust optimization. While stochastic optimization considers probability distribution, robust optimization assumes that uncertain data belong to a given uncertainty set and aims to optimize performance considering the worst-case scenario within that set.

In this paper, we propose exact solution methods to solve the robust job-shop scheduling problem. A two-stage robust optimization approach is used to deal with processing times uncertainty, where the first stage fixes the sequence of operations on machines whilst the second stage sets the operation start times.

## 2 Problem statement

An instance of the JSSP implies a set of jobs  $\mathcal{J}$  and a set of machines  $\mathcal{M}$ . Each job  $i \in \mathcal{J}$  consists of a sequence of  $n_i$  operations. The  $j^{\text{th}}$  operation  $O_{i,j} \in \mathcal{O}_i$  of a job  $i$  must be performed by machine  $\mu_{i,j} \in \mathcal{M}$  (with  $\mu_{i,j} = m \iff O_{i,j} \in \mathcal{I}_m$ , where  $\mathcal{I}_m$  is the set of operations processed by machine  $m$ ) and  $p_{i,j}$  denotes its processing time. Each machine can process at most one operation  $O_{i,j} \in \mathcal{I}_m$  at a time, each job can only be processed on one machine at a time, and preemption is not allowed: once an operation is started, it must be processed without any interruption.

We consider that the processing times of operations are uncertain. Each processing time  $p_{i,j}$  of an operation  $O_{i,j} \in \mathcal{O}_i, i \in \mathcal{J}$ , belongs to the interval  $[\bar{p}_{i,j}, \bar{p}_{i,j} + \hat{p}_{i,j}]$ , where  $\bar{p}_{i,j}$  is the nominal value and  $\hat{p}_{i,j}$  the maximum deviation of the processing time from its nominal value.

The traditional robust optimization approach consists in protecting against the case when all parameters can deviate at the same time, which makes the solution overly conservative. Indeed, there is a very low probability that all parameters take their worst value all together. To overcome this limitation, Bertsimas (2004) introduces an uncertainty budget

approach that allows a restriction on the number of deviations that can occur simultaneously to a given budget. In order to reach a trade-off between robustness and solution quality, we exploit this approach to define the uncertainty set.

Let  $\Gamma$  be the budget of uncertainty, *i.e.*, the maximum number of operations whose processing time can deviate simultaneously. For each scenario  $\xi$ , the processing time of operation  $O_{i,j}$  is given by:

$$p_{i,j}(\xi) = \bar{p}_{i,j} + \xi_{i,j} \hat{p}_{i,j} \quad (1)$$

where  $\xi_{i,j}$  is equal to 1 if the processing time of the operation deviates, 0 otherwise

We then define the uncertainty set  $\mathcal{U}^\Gamma$  as:

$$\mathcal{U}^\Gamma = \{(\xi_{i,j})_{i \in \mathcal{J}, 1 \leq j \leq n_i} \mid \sum_{i \in \mathcal{J}} \sum_{j=1}^{n_i} \xi_{i,j} \leq \Gamma\}. \quad (2)$$

The robust multi-stage optimization, introduced by Ben-Tal (2004), considers that a part of the decision variables must be instantiated before the uncertainty is revealed, while the other variables can be adjusted to the uncertainty realization. In our problem, we consider that the purpose is to find the sequence on the machines (first stage decisions), allowing to define a start time for each operation and each scenario (second stage decisions), minimizing the makespan in the worst-case scenario.

### 3 Solution methods

A robust problem can be solved using an extended formulation, which consists in duplicating the set of constraints involving uncertain parameters (in the present case, operation processing times) for all possible scenarios  $\xi \in \mathcal{U}^\Gamma$ . Usually formulated as a linear programming problem, it is also possible to adopt a constraint programming approach (Juvin 2023).

However, according to the structure of our uncertainty set, the number of scenarios increases exponentially with the number of operations, which quickly makes these models intractable. Therefore, in this section we deal with the evaluation of a worst-case scenario. This study then allows us to propose a compact formulation and decomposition methods of the problem.

#### 3.1 Worst-case evaluation

In this section, it is assumed that a first-stage solution  $\sigma$  is given. Considering an uncertainty budget  $\Gamma$ , the worst-case evaluation is to identify a scenario, with at most  $\Gamma$  operations whose duration deviates, and that leads to the largest possible makespan. This problem can also be treated as the evaluation of a longest path in an directed acyclic graph (DAG). Such a method is actually used by Bold (2021) in the context of a robust resource-constrained project scheduling problem (RCPSp).

#### 3.2 Compact model

As Bold (2021) for the robust RCPSp, we propose a compact formulation of the robust JSSP, based on the dual of the worst-case evaluation subproblem. We introduce the variables  $C_{i,j}^\gamma$ , which represent the end date of operation  $O_{i,j}$  in the worst case, taking into account at most  $\gamma$  deviations. The compact model is as follows:

$$\min C_{\max} \quad (3)$$

$$s.t. \quad C_{\max} \geq C_{i,n_i}^\Gamma \quad \forall i \in \mathcal{J} \quad (4)$$

$$C_{i,j}^\gamma \geq C_{i,j-1}^\gamma + \bar{p}_{i,j} \quad \forall i \in \mathcal{J}, \forall j \in \{2, \dots, n_i\}, \forall \gamma \in \{0, \dots, \Gamma\} \quad (5)$$

$$C_{i,j}^\gamma \geq C_{i,j-1}^{\gamma-1} + \bar{p}_{i,j} + \hat{p}_{i,j} \quad \forall i \in \mathcal{J}, \forall j \in \{2, \dots, n_i\}, \forall \gamma \in \{1, \dots, \Gamma\} \quad (6)$$

$$C_{i,j}^\gamma \geq C_{i',j'}^\gamma + \bar{p}_{i,j} - y_{i,j,i',j'} \cdot H \quad \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i',j'}) \in \mathcal{I}_m^2, \forall \gamma \in \{0, \dots, \Gamma\} \quad (7)$$

$$C_{i,j}^\gamma \geq C_{i',j'}^{\gamma-1} + \bar{p}_{i,j} + \hat{p}_{i,j} - y_{i,j,i',j'} \cdot H \quad \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i',j'}) \in \mathcal{I}_m^2, \forall \gamma \in \{1, \dots, \Gamma\} \quad (8)$$

$$C_{i',j'}^\gamma \geq C_{i,j}^\gamma + \hat{p}_{i,j} - (1 - y_{i,j,i',j'}) \cdot H \quad \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i',j'}) \in \mathcal{I}_m^2, \forall \gamma \in \{0, \dots, \Gamma\} \quad (9)$$

$$C_{i',j'}^\gamma \geq C_{i,j}^{\gamma-1} + \hat{p}_{i,j} + \hat{p}_{i,j} - (1 - y_{i,j,i',j'}) \cdot H \quad \forall m \in \mathcal{M}, \forall (O_{i,j}, O_{i',j'}) \in \mathcal{I}_m^2, \forall \gamma \in \{1, \dots, \Gamma\} \quad (10)$$

$$C_{i,1}^0 \geq \bar{p}_{i,1} \quad \forall i \in \mathcal{J} \quad (11)$$

$$C_{i,1}^\gamma \geq \bar{p}_{i,1} + \hat{p}_{i,1} \quad \forall i \in \mathcal{J}, \forall \gamma \in \{1, \dots, \Gamma\} \quad (12)$$

### 3.3 Decomposition methods

We present a logic-based Benders decomposition method (Hooker 2000) as well as a column and constraint generation method (Zeng 2013). These two iterative approaches aim to decompose the problem into a master problem, and an adversarial subproblem, and share the same pattern. The master problem is formulated with an extended model for the robust job-shop problem (using MILP or CP) considering only a subset of scenarios and the subproblem evaluates the worst-case scenario. At each iteration, information relating to this worst-case scenario is added to the master problem.

For the Benders decomposition method, adding information about the violated scenarios consists in adding cuts in the master problem. Note that only the MILP formulation of the master is considered. The added cuts are as follows:

$$C_{\max} \geq \psi_h^* \cdot (1 - \text{NumberOfChanges}_h) \quad (13)$$

where  $\psi_h^*$  is the worst-case makespan obtained by the adversarial subproblem at iteration  $h$  and  $\text{NumberOfChanges}_h$  is the number of changes, compared with the decisions made at the first stage of iteration  $h$ , that could affect the value of makespan. If no influential changes occur, then the makespan is at least equal to  $\psi_h^*$ ; otherwise, the constraint is inactive.

For the column and constraint generation procedure, adding information about the violated scenario consists in generating the corresponding second-stage decision variables and the associated constraints. This is simply adding the worst-case scenario to the set of scenarios considered in the next iteration of the master:

$$\mathcal{U}^{k+1} = \mathcal{U}^k \cup \{\xi_k^*\}. \quad (14)$$

## 4 Numerical results

For computational experiments, we consider 58 classical instances of the job-shop problem from the literature, adapted to the robust context by randomly generating deviation values. We test all the models by varying the uncertainty budget according to four ratios: 5,%, 10,%, 15,% and 20,%, i.e. a total of 232 experiments per method.

The results of Table 1 are presented in terms of the number of best solutions found compared to the other methods ("best"), as well as the optimality gap ("gap (%)") obtained by each method.

For the smallest instances ( $6 \times 6$ ), all the methods succeed in finding the optimal solution (except one for the  $CCG_{MILP}$  method). For 10-machine instances, the *Benders* method

$\mathcal{J}$	$\mathcal{M}$	#	<i>Compact</i>		<i>Benders</i>		<i>CCG<sub>MILP</sub></i>		<i>CCG<sub>CP</sub></i>	
			best	gap (%)	best	gap (%)	best	gap (%)	best	gap (%)
6	6	4	4	0	4	0	3	0.5	4	0
10	5	20	17	13.8	16	34	7	8.45	15	3.5
10	10	72	28	20.64	43	45.15	5	27.85	9	24.49
15	5	20	12	43.75	6	57.3	6	57.4	20	1.05
15	10	20	1	44	7	58.3	6	58.45	10	33.8
15	15	20	0	40.8	6	60.25	6	60.1	11	42.4
20	5	24	2	61.63	4	70.54	4	70.58	24	3.88
20	10	20	0	50.8	1	67.5	2	67.45	18	31.85
20	15	12	0	–	1	65.75	0	65.67	11	47.25
30	10	20	0	–	0	79.65	0	79.25	20	18.3

**Table 1.** Number of best solutions found and average optimality gap for job-shop instances from the literature, categorized according to the instance size.

obtains the largest number of best solutions, but with relatively high optimality gaps. Finally, the *CCG<sub>CP</sub>* method obtains the highest number of best solutions for the largest instances.

## 5 Conclusion

In this paper, we study the robust job-shop scheduling problem where operation processing times are uncertain and modeled by an uncertainty budget. We consider a two-stage decision process, where the sequences of operations must be decided before knowing the realization of the uncertainty, in order to be feasible for all scenarios, but where the processing dates of the operations can be adapted according to the observed durations. We propose a compact formulation and two decomposition methods based on solving a relaxed master problem and finding violated constraints at each iteration. For the largest instances, decomposition methods, in particular the column and constraint generation method with a master problem solved using constraint programming, yields better quality solutions.

## References

- Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovsk, 2004, “Adjustable robust solutions of uncertain linear programs”, *Mathematical Programming*, Vol. 99, pp. 351–376.
- Dimitris Bertsimas and Melvyn Sim, 2004, “The price of robustness”, *Operations Research*, Vol. 52, pp. 35–53.
- Matthew Bold and Marc Goerigk, 2021, “A compact reformulation of the two-stage robust resource-constrained project scheduling problem”, *Computers & Operations Research*, Vol. 130, pp. 105232.
- John N. Hooker, 2000, “Logic based methods for optimization: Combining optimization and constraint satisfaction”, John Wiley & Sons, New York, 2000.
- Carla Juvin, Laurent Houssin, and Pierre Lopez, 2023, “Constraint programming for the robust two-machine flow-shop scheduling problem with budgeted uncertainty”, *CPAIOR 2023*.
- Bo Zeng and Long Zhao, 2013, “Solving two-stage robust optimization problems using a column-and-constraint generation method”, *Operations Research Letters*, Vol. 41, pp. 457–461.