Hybridization of Mixed-Integer Linear Program and Discrete Event Systems for Robust Scheduling on Parallel Machines

A. Aubry[1](✉), P. Marangé[1], D. Lemoine[2], S. Himmiche[1], and S. Norre[3]

[1] Université de Lorraine, CNRS, CRAN, 54000 Nancy, France
{alexis.aubry,pascale.marange,sara.himmiche}@univ-lorraine.fr
[2] IMT-Atlantique, LS2N UMR 6004, Nantes, France
david.lemoine@imt-atlantique.fr
[3] Clermont Auvergne University, LIMOS UMR 6158, Clermont-Ferrand, France
sylvie.norre@uca.fr

Abstract. This paper proposes an approach for robust scheduling on parallel machines. This approach is based on a combination of robust mathematical and discrete event systems models which are iteratively called in order to converge towards a schedule with the required robustness level defined by the decision maker. Experimentations on a small instance (10 jobs and 2 unrelated machines) and a more complex one (30 jobs and 6 uniform machines) show that this approach permits to converge quickly to a robust schedule even if the probability distribution associated to the uncertainties are not symmetrical. The approach achieves a better rate of convergence than those of the literature's methods.

Keywords: Robust scheduling · Robust mixed integer programming model · Discrete event systems · Parallel machines

1 Introduction

Scheduling under uncertainties is still a present concern in Operation Research and Decision Aiding. Many researchers are interested in determining a robust schedule which is rather insensitive to the data uncertainties and which is able to absorb the perturbations without unreasonably degrading its performances. However, the concept of robustness is differently defined according to the domains. An approach for robust optimization has been proposed by [1], based on a Robust Mixed Integer Programming Model. In order to obtain the robustness level wanted by the decision maker, a vector Ω, which corresponds to the maximum allowed deviation of input parameters, has to be fixed. This vector can be interpreted as a robustness coefficient. The authors have shown

A. Dolgui et al. (Eds.): APMS 2021, IFIP AICT 630, pp. 73–80, 2021.
https://doi.org/10.1007/978-3-030-85874-2_8

that if the uncertainties on parameters are independent and symmetrically distributed, the vector Ω can be analytically computed. But, such an hypothesis is not often verified in real scheduling problems. In [2], a methodology for iteratively and numerically tuning Ω has been proposed thanks to a combination of a robust mathematical programming and Discrete Event Systems models when the probability distribution associated to the uncertainties are not symmetrical. This generic method has been applied on a scheduling problem with parallel machines and the results show that this approach is a good mean for tuning the Ω parameters. However, the mechanism for updating coefficients Ω from one iteration to another was simple and naive. Moreover, it did not take into account the characteristics of the current schedule solution (in particular the load of the machines). As a result, the convergence of the method toward a solution with the required robustness level was relatively slow. Our contribution in this paper is to propose a new mechanism for updating robustness coefficients Ω to increase the rate of convergence of our method in case of scheduling problem on parallel machines. The problem is thus denoted as $RK||C_{max}$ which consists in minimizing the makespan (C_{max}) on K parallel machines.

The paper is built as follows. The first section presents the robust mixed integer programming model for scheduling on parallel machines. The second section presents the methodology and details the mechanism for updating the robustness coefficients Ω. The third section discusses the results on two instances: a simple instance composed of 10 jobs and 2 unrelated machines and a more complex one composed of 30 jobs and 6 uniform machines. Finally, the last section deals with the conclusion and the perspectives.

2 Robust Formulation of $RK||C_{max}$

The main assumptions for $RK||C_{max}$ are the following:

- all jobs are available at time 0,
- the K machines are always available (no breakdown etc.),
- processing times for the jobs are independent,
- a machine cannot process more than one job at any time and preemption is not allowed.

Parameters and decision variables of the model are summarized in Table 1.

Table 1. Notations of the model

N:	Number of jobs which have to be scheduled
K:	Number of parallel machines
t_{jk}:	Processing time for job j on the machine k, $\forall (j,k) \in \{1,\ldots,N\} \times \{1,\ldots,K\}$
x_{jk}:	$\begin{cases} 1 & \text{if } j \text{ is executed on machine } k \\ 0 & \text{otherwise} \end{cases}, \forall (j,k) \in \{1,\ldots,N\} \times \{1,\ldots,K\}$
C_{max}:	is the makespan value

Since processing times are uncertain, we assume that

$$\forall (j,k) \in \{1,\ldots,N\} \times \{1,\ldots,K\},\ t_{jk} \in \left[t_{jk}^{\min}, t_{jk}^{\max}\right]$$

Following the methodology of [1], uncertain processing times are modelled as

$$t_{jk} = \bar{t}_{jk} + \zeta_{jk}\hat{t}_{jk}$$

where

- $\bar{t}_{jk} = \frac{t_{jk}^{\max}+t_{jk}^{\min}}{2}$ and $\hat{t}_{jk} = \frac{t_{jk}^{\max}-t_{jk}^{\min}}{2}$
- ζ_{jk} is a random variable which takes its values in $[-1,1]$

Thus, the robust model can be formulated as follows:

$$\text{Minimize } C_{max} \tag{1}$$

s.t.

$$\sum_{k=1}^{K} x_{jk} = 1 \quad \forall j \in \{1,\ldots,N\} \tag{2}$$

$$\sum_{j=1}^{N} \bar{t}_{jk}x_{jk} + \max_{\sum_{j=1}^{N}|\zeta_{jk}|\le\Omega_k} \left(\sum_{j=1}^{N} \hat{t}_{jk}\zeta_{jk}x_{jk}\right) \le C_{max} \quad \forall k \in \{1,\ldots,K\} \tag{3}$$

$$x_{jk} \in \{0,1\} \quad \forall (j,k) \in \{1,\ldots,N\} \times \{1,\ldots,K\} \tag{4}$$

where $\Omega = (\Omega_k)_{k\in\{1,\ldots,K\}}$ constrains the maximum deviation of processing times allowed on each machine k. It can be interpreted as a robustness coefficient: the larger Ω_k is, the more conservative is the constraint (3). We denote as X^Ω an optimal schedule provided by the robust model with the input Ω and as $C_{\max}\left(X^\Omega\right)$, the associated optimal makespan.

Let $\widetilde{C}_{\max}\left(X^\Omega\right)$ be defined as the random variable associated to the makespan, when X^Ω is executed in the workshop (with the uncertain values t_{jk}). It is possible to define a robustness indicator as the probability that $\widetilde{C}_{\max}\left(X^\Omega\right)$ is lower than $C_{\max}\left(X^\Omega\right)$. More formally, this indicator is given by the Eq. (5):

$$\Gamma(C_{\max}\left(X^\Omega\right)) = \mathbb{P}\left[\tilde{C}_{\max}\left(X^\Omega\right) \le C_{\max}\left(X^\Omega\right)\right] \tag{5}$$

Our goal is to find Ω to guarantee that the scheduling X^Ω reaches a certain level of robustness Γ^{ref}. As [1] have shown that in case of each ζ_{jk} is symmetrically distributed in $[-1,1]$, determining such Ω can be done analytically. Next section provides a methodology which allows to reach this goal in a general case.

3 Our Methodology

In this section, we denote as X_k^Ω the schedule extracted from X^Ω by considering only the set of jobs processed on machine k, $k \in \{1,\ldots,K\}$.

Basically, our methodology is based on the following observations:

1. In a workshop of parallel machines, machines are independent meaning that

$$\Gamma(C_{\max}\left(X^{\Omega}\right)) = \prod_{k=1}^{K} \Gamma(C_{\max}\left(X_k^{\Omega}\right))$$

2. if $\Gamma(C_{\max}\left(X^{\Omega}\right)) < \Gamma^{ref}$ then there exists at least one $k \in \{1, \ldots, K\}$ such that $\Gamma(C_{\max}\left(X_k^{\Omega}\right)) < \sqrt[K]{\Gamma^{ref}}$ which is equivalent to

$$\mathbb{P}\left[\tilde{C}_{\max}\left(X_k^{\Omega}\right) \leq C_{\max}\left(X_k^{\Omega}\right)\right] < \sqrt[K]{\Gamma^{ref}}$$

Thus, for such a k, we estimate a value $\tilde{d}\left(X_k^{\Omega}\right) > C_{\max}\left(X_k^{\Omega}\right)$ for which the following constraint is satisfied.

$$\mathbb{P}\left[\tilde{C}_{\max}\left(X_k^{\Omega}\right) \leq \tilde{d}\left(X_k^{\Omega}\right)\right] = \sqrt[K]{\Gamma^{ref}} \tag{6}$$

Therefore, to obtain this robustness score for the scheduling X_k^{Ω}, we have to compute a new robustness coefficient Ω_k^{new} in order to satisfy the constraint (7).

$$\max_{\sum_{j \in X_k^{\Omega}} |\zeta_{jk}| \leq \Omega_k^{new}} \left(\sum_{j \in X_k^{\Omega}} \hat{t}_{jk} \zeta_{jk} \right) = \tilde{d}\left(X_k^{\Omega}\right) - \sum_{j \in X_k^{\Omega}} \bar{t}_{jk} \tag{7}$$

This constraint is derived from (3) in which we allow the deviation of uncertainties constrained by Ω_k^{new} until we reach $\tilde{d}\left(X_k^{\Omega}\right)$.

As we want to be the less conservative as possible (that means, in our case, that Ω_k^{new} should be as small as possible), the following proposition can be stated:

Proposition 1. *Solving Eq. (7) is equivalent to solve the following continuous Knapsack problem:*

$$Minimize\ \Omega_k^{new} = \sum_{j \in X_k^{\Omega}} \zeta_{jk} \tag{8}$$

$$s.t.$$

$$\sum_{j \in X_k^{\Omega}} \hat{t}_{jk} \zeta_{jk} = \tilde{d}\left(X_k^{\Omega}\right) - \sum_{j \in X_k^{\Omega}} \bar{t}_{jk} \tag{9}$$

$$\zeta_{jk} \in [0, 1] \quad \forall j \in X_k^{\Omega} \tag{10}$$

As the continuous Knapsack problem is known to be polynomial and can be easily solved by a greedy algorithm, we can use such algorithm to find the value of Ω_k^{new} that satisfies (7).

A similar reasoning can be applied if $\Gamma(C_{\max}\left(X^{\Omega}\right)) > \Gamma^{ref}$ and leads to the same result.

Based on the previous proposition, our methodology can be summarized into 4 steps:

Step 1: For $\Omega = (\Omega_k)_{k \in \{1,\ldots,K\}}$, an optimization module based on a linear solver provides X^{Ω} an optimal schedule according to the robust model presented in the previous section [1].
Step 2: Thanks to Discrete Event Systems models and tools, $\Gamma(C_{\max}(X^{\Omega}))$ is then evaluated. If $\Gamma(C_{\max}(X^{\Omega})) \in [\Gamma^{ref} - \epsilon, \Gamma^{ref} + \epsilon]$ then the process stops and X^{Ω} is the required schedule solution, else the third step is engaged. ϵ is a parameter that helps to fix the required level of accuracy.
Step 3: For each $k \in \{1, \cdots, K\}$, the value of $\tilde{d}(X_k^{\Omega})$ that satisfies constraint (6) is then determined thanks to another Discrete Event Systems module. Ω is then updated in the Step 4.
Step 4: Thanks to Eq. (7), $\forall k \in \{1, \cdots, K\}$, Ω_k^{new} is then computed using the Proposition 1 and we loop back to Step 1 with $\Omega = (\Omega_k^{new})_{k \in \{1,\ldots,K\}}$.

Steps 2 and 3 use models and tools from discrete event systems (DES) [3]. Discrete event systems (DES) allow to model the behavior of a system by considering its possible states and the possible events (allowing the evolution from one state to another). The event is an instantaneous occurrence of an action or a phenomenon in the system environment. The evolution on the event occurrence can be deterministic when the behavior is known with certainty or stochastic when the behavior is uncertain and the evolution can lead to different states. The works of [4–6] have shown that DES are particularly relevant for scheduling evaluation due to their ability to model the behavior of industrial systems and perturbations. Indeed, DES allow to represent many dynamic features such as the communication between the elements of the workshop (jobs, resources), the time and the probabilistic behavior of perturbations. Many stochastic discrete event system languages allow to model these characteristics: Stochastic Petri Nets [7], Stochastic automata [8], Stochastic Automata Networks [9].

The method for evaluating the impact of a set of perturbations on a given schedule uses the approach proposed by [4]. The approach models the characteristics of the workshop (operations, resources), and the probability distribution associated to the perturbation by a discretization of this one and allows the evaluation of different elements: the robustness indicator $\Gamma(C_{\max}(X^{\Omega}))$ as in step 2, the minimal duration $\tilde{d}(X_k^{\Omega})$ as in step 3, ...

4 Case Studies

4.1 A Simple Instance

In this first application, 10 jobs are considered. These jobs can be executed on two unrelated parallel machines (the problem is $R2||C_{max}$). Γ^{ref} is fixed to 90% and ϵ is fixed to 1% such that $\Gamma(C_{\max}(X^{\Omega}))$ has to be in $[89\%, 91\%]$ for concluding to the acceptability of the solution X^{Ω}. We applied the two approaches defined (from [2] and the presented one) by starting with $\Omega = [0, 0]$ (meaning that no uncertainty is considered).

The Table 2 summarizes the results by giving the extremal iterations of the combined approach presented in [2]. Two solutions are explored during the different iterations. The solution $X1$ allocates the first machine to jobs 1, 4, 6, 7, 8 and the second machine to jobs 2, 3, 5, 9, 10. The solution $X2$ allocates the first machine to jobs 1, 4, 6, 7, 8, 10 and the second machine to jobs 2, 3, 5, 9.

Table 2. Obtained iterations for $R2||C_{max}$ with the approach from [2]

Iteration i	Ω	Solution		Ω^{new}
		X^{Ω}	$C_{\max}\left(X^{\Omega}\right)$	
0	[0, 0]	X_1	12	[0.10, 0.18]
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	[0.53, 0.66]	X_1	13.98	$\emptyset$

Table 3. Obtained iterations for the problem $R2||C_{max}$ with our approach

Iteration i	Input Ω	Step 1		Step 2	Step 3	Step 4
		X^{Ω}	$C_{\max}\left(X^{\Omega}\right)$	$\Gamma\left(C_{\max}\left(X^{\Omega}\right)\right)$	$\left[\tilde{d}\left(X_k^{\Omega}\right)\right]_{k\in\{1,2\}}$	Ω^{new}
0	[0, 0]	X_1	12	0.65	[13, 14]	[0.50, 0.66]
1	[0.50, 0.66]	X_1	14	0.90	$\emptyset$	$\emptyset$

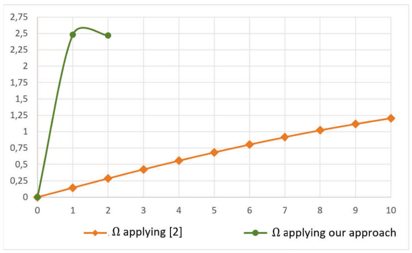
We can conclude that the improved approach finds the right solution in the first iteration (as $\Gamma\left(C_{\max}\left(X^{\Omega}\right)\right) = 90\%$, it is not necessary to update Ω and to run a new iteration). Thus, the improved method decreases drastically the number of necessary iterations for converging to a comparable solution.

4.2 A More Complex Instance

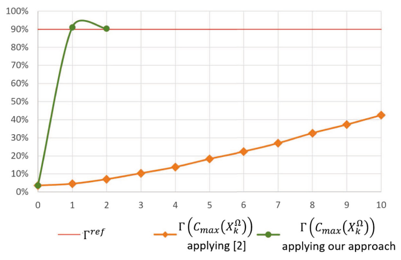
We consider 30 jobs which can be executed on 6 uniform machines (the problems is $Q6||C_{\max}$). The machines are such that the first machine is the fastest and the sixth is the slowest. Γ^{ref} is fixed to 90% and ϵ is fixed to 1%.

We applied the two approaches by starting with $\Omega = [0, 0, 0, 0, 0, 0]$.

The Fig. 1 presents the evolution of the average value of Ω and $\Gamma(C_{\max}\left(X_k^{\Omega}\right))$ according to the iteration when applying the two approaches. First, it can be observed that even after 10 iterations, the targeted performance Γ^{ref} is not reached when applying the approach of [2] and $\Gamma\left(C_{\max}\left(X_k^{\Omega}\right)\right)$ remains low. Moreover, Ω increases very slowly when applying the approach of [2] in comparison with our approach. We can postulate that a lot of iterations is still necessary for reaching the target when applying the approach of [2].



(a) Evolution of the average value of Ω with the iterations



(b) Evolution of $\Gamma(C_{\max}\left(X_k^{\Omega}\right))$ with the iterations

Fig. 1. Comparison of the results when applying the two approaches

Table 4. Obtained iterations for the problem $Q6||C_{\max}$ with our approach

Iteration i	Input Ω	Step 1		Step 2	Step 3	Step 4
		X^{Ω}	$C_{\max}\left(X^{\Omega}\right)$	$\Gamma\left(C_{\max}\left(X^{\Omega}\right)\right)$	$\left[\tilde{d}\left(X_k^{\Omega}\right)\right]_{k\in\{1,2\}}$	Ω^{new}
0	$[0, 0, 0, 0, 0, 0]$	X_0	926	3.6%	$[968, 969, 974, 971, 976, 979]$	$[2.67, 2.80, 2.50, 2.50, 2.23, 2.18]$
1	$[2.67, 2.80, 2.50, 2.50, 2.23, 2.18]$	X_1	973	91.1%	$[972, 970, 971, 973, 973, 976]$	$[2.67, 2.67, 2.39, 2.50, 2.26, 2.32]$
2	$[2.67, 2.67, 2.39, 2.50, 2.26, 2.32]$	X_2	972.72	90.3%	$\emptyset$	$\emptyset$

The Table 4 detailed the iterations when applying our improved approach. After the first iteration, the obtained value of $\Gamma\left(C_{\max}\left(X^{\Omega}\right)\right)$ is lightly too high such that a second iteration for adjusting the values of Ω is necessary. The second iteration allows to reach the targeted performance. If we accept to degrade the optimal deterministic makespan ($C_{\max} = 926$ with the solution X_0) of only 5%, then it is possible to guarantee this makespan despite the uncertainties with a probability of 90% giving a good compromise between optimality (C_{max}) and robustness (Γ). This confirms globally the results obtained with the simple instance: the improved approach permits to converge faster to a robust solution.

5 Conclusion and Perspectives

We have proposed an approach combining robust mathematical programming and Discrete Event Systems models for the building of a robust scheduling on parallel machines. This allows to reach the level of robustness desired by the decision-maker by finely assessing the degree of robustness of the solutions provided by the optimization module, regardless of the probability distributions that follow the uncertainties on the model input data. The probability distribution associated to the uncertainties are supposed independent but not necessarily symmetrical. Experiments on two instances show that our approach permits to converge quickly to a robust schedule and improves the rate of convergence of literature's methods. Several perspectives to this work can be considered. First, at short term, a more specific distribution of levels of robustness can be considered, taking into account, for example, the configuration of the production system, the criticality of certain machines (for instance, requiring greater robustness for bottleneck machines, ...). It would also be interesting to consider dependent probability distributions associated to the uncertainties. Long term perspectives will concern the extension of this approach to more complex shop scheduling problems as flow shop, job shop or hybrid flow shop.

References

1. Bertsimas, D., Sim, M.: The price of robustness. Oper. Res. **2**(1), 35–53 (2004)
2. Marangé, P., et al.: Coupling robust optimization and model-checking techniques for robust scheduling in the context of *Industry 4.0*. In: Sokolov, B., Ivanov, D., Dolgui, A. (eds.) Scheduling in Industry 4.0 and Cloud Manufacturing. ISORMS, vol. 289, pp. 103–124. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43177-8_6
3. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems, 3rd edn. Springer, Cham (2021)
4. Himmiche, S., Aubry, A., Marangé, P., Pétin, J.-F., Duflot, M.: Using statistical-model-checking-based simulation for evaluating the robustness of a production schedule. In: 7th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, SOHOMA 2017, Nantes, France, October 2017 (2017)
5. Lefebvre, D., Mejia, G.: Robust scheduling in uncertain environment with Petri nets and beam search. IFAC-PapersOnLine **51**(11), 1077–1082 (2018)
6. Cherif, G., Leclercq, E., Lefebvre, D.: Scheduling problems for a class of hybrid FMS using T-TPN and beam search. J. Control Autom. Electr. Syst. **32**(3), 591–604 (2021). https://doi.org/10.1007/s40313-021-00700-5
7. Chiola, G., Marsan, M.A., Balbo, G., Conte, G.: Generalized stochastic Petri nets: a definition at the net level and its implications. IEEE Trans. Softw. Eng. **19**(2), 89–107 (1993)
8. Alur, R., Dill, D.L.: A theory of timed automata. Theoret. Comput. Sci. **126**(2), 183–235 (1994)
9. Plateau, B., Atif, K.: Stochastic automata network of modeling parallel systems. IEEE Trans. Softw. Eng. **17**(10), 1093–1108 (1991)