



HAL
open science

On Solving the Job Sequencing and Tool Switching Problem with Non-identical Parallel Machines and Stochastic Processing Times

Khaled Khayati, Khadija Hadj Salem, Benoit Lardeux, Safa Bhar Layeb

► To cite this version:

Khaled Khayati, Khadija Hadj Salem, Benoit Lardeux, Safa Bhar Layeb. On Solving the Job Sequencing and Tool Switching Problem with Non-identical Parallel Machines and Stochastic Processing Times. The 2024 International Conference of the African Federation of Operational Research Societies (AFROS), Tlemcen University, Algeria, Nov 2024, Tlemcen, Algeria. <hal-04675154>

HAL Id: hal-04675154

<https://hal.science/hal-04675154v1>

Submitted on 26 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.



HAL Authorization

On Solving the Job Sequencing and Tool Switching Problem with Non-identical Parallel Machines and Stochastic Processing Times

Khaled Khayati

National Engineering School of Tunis
University of Tunis El Manar
Tunis, Tunisia
khaled.khayati@etudiant-enit.utm.tn

Khadija Hadj Salem

L@bISEN, Usine du Futur, ISEN Yncréa Ouest
33 Quater Avenue du Champ de Manœuvres
Nantes, France
khadija.hadj-salem@isen-ouest.yncrea.fr

Benoit Lardeux

L@bISEN, Usine du Futur, ISEN Yncréa Ouest
33 Quater Avenue du Champ de Manœuvres
Nantes, France
benoit.lardeux@isen-ouest.yncrea.fr

Safa Bhar Layeb

LR-OASIS, National Engineering School of Tunis
University of Tunis El Manar
Tunis, Tunisia
safa.layeb@enit.utm.tn

Abstract—This paper investigates the job sequencing and tool switching problem with non-identical parallel machines and stochastic processing times, referred to as SSP-NPM-SPT. It simultaneously determines the job processing sequence and the tool switching schedule while minimizing the makespan. Processing time uncertainty reflects the real-world conditions of flexible manufacturing systems where processing time variability can be attributed to many factors such as machines, humans, and supply chains. We propose a Monte Carlo Simulation approach to generate various scenarios and multiple levels of variability for stochastic processing times to determine optimal parameter settings. Extensive computational experiments were performed using benchmark instances from the literature and considering a representative set of scenarios. The results show that integrating stochastic modeling with Monte Carlo Simulation provides a robust framework to effectively address the complexities of SSP-NPM-SPT.

Index Terms—Scheduling, Job Sequencing, Tool Switching, Parallel Machines, Stochastic Processing Times, Monte Carlo and Scenarios Generation.

I. INTRODUCTION

In flexible manufacturing, transportation management systems encounter substantial challenges in efficiently optimizing job sequencing across machines with diverse capabilities. Effective management of manufacturing resources, particularly in Flexible Manufacturing Systems (FMS), is crucial to maintain competitiveness and improve productivity. FMSs are designed to be adaptable to changes in all aspects of production. Frequently, FMS utilizes parallel machines with different tool capacities and switching times.

In this context, this study addresses the Job Sequencing and Tool Switching Problem with Non-Identical Parallel Machines (SSP-NPM), with uncertainty in processing times. It consists of simultaneously finding (i) the best job sequence across a set of non-identical parallel machines with magazines of different

capacities and (ii) the best tool switching (loading/unloading) schedule such that the makespan is minimized and all operational constraints are respected. Researchers have studied the SSP-NPM, a variation of the classical SSP, for several decades in the context of Flexible Manufacturing Systems (FMS). Originally introduced by [20] and [1], it is a well-known NP-hard combinatorial problem [10]. However, given a fixed sequence of jobs to be processed, finding a schedule of tool switching can be optimally solved in deterministic polynomial time using the so-called *Keep Tool Needed Soonest* (KTNS) policy proposed by [20]. On the other hand, several approaches, including exact and heuristic/metaheuristic, were developed to tackle the NP-hard variant (see e.g., [10], [14], [12], etc). The reader is referred to [4] for a recent survey focusing on the review of the methods proposed in the literature for the solution of the SSP.

For the SSP-NPM variant, [7] introduced three mathematical models and heuristics to solve it. His study highlights the need for adaptable strategies based on each production objective, e.g., minimizing the number of tool switches (TS), minimizing the makespan, or minimizing the total flowtime (TFT). The proposed models include the precedence-based, position-based, and time-index-based Mixed Integer Linear Programming formulation (MILP) formulations. Furthermore, Calmels [6] proposed an iterated local search heuristic to solve large-scale instances.

Not surprisingly, scheduling is one of the most strategic challenges in manufacturing and production systems. Historically, most research focused on deterministic scenarios, assuming that job processing time is known beforehand. But, in real-world manufacturing environments, it's uncertain and must follow a stochastic approach, which can render deterministic schedules ineffective once applied on the shop floor.

Many factors contribute to processing time uncertainty, namely human skills, technical issues, machine breakdowns, and tool wear.

When dealing with uncertainty, [16] proposed a method based on external sampling to generate scenarios and use the Sample Average Approximation (SAA) method to solve the models. This approach is applicable in systems where highly variable operational conditions, which affect job processing times are considered. On the other hand, [8] studied the use of gamma or log-normal distributions to model processing times in scheduling unrelated parallel machines.

The scope of this work is to address the SSP-NPM with stochastic processing times, referred to as SSP-NPM-SPT. To our knowledge, this is the first work that deals with uncertainty in the context of SSP-NPM. Therefore, this study presents a Monte Carlo simulation (MCS) approach to address SSP-NPM-SPT and extensive computational experiments were conducted on benchmark instances from the literature to evaluate its effectiveness.

The remainder of this paper is organized as follows. Section II is devoted to the problem description and Section III reports an analogy with a real case. Section IV briefly reviews the literature of recent related works on scheduling under uncertainty. Section V describes the proposed MCS approach. To evaluate the effectiveness of our approach, computational experiments are conducted on benchmark instances from the SSP-NPM literature, and numerical results are analyzed in Section VI. Finally, Section VII concludes the work and derives avenues for future research.

II. PROBLEM DESCRIPTION

This SSP-NPM in its deterministic variant can be described as follows: let $\mathcal{J} := \{1, \dots, J\}$ be a set of jobs to be processed on a set of machines $\mathcal{M} := \{1, \dots, M\}$. Each job j is assumed to be available for processing at time zero, and there are no restrictions on machine eligibility. Jobs are processed sequentially on each machine m . The processing time p_{jm} of job j depends on machine m , where each machine m has a tool magazine capacity C_m . Additionally, each machine m may have a distinct tool switching time sw_m . Let $\mathcal{T} := \{1, \dots, T\}$ be the set of tools required for processing all jobs \mathcal{J} . Each tool t occupies one slot in a machine's tool magazine. It's assumed that at least one machine had a sufficient capacity to accommodate all tools required for any job j in its tool magazine. Each job j requires a subset of tools \mathcal{T}_j , and job j can only be processed when all tools listed in \mathcal{T}_j are present in the tool magazine of machine m to which it is assigned. \mathcal{J}_t denotes the subset of jobs requiring a specific tool $t \in \mathcal{T}$. Tool switching may be mandatory when successive jobs are assigned to the same machine due to the limited machine capacity C_m . It is assumed that the initial loading does not count as a tool switch, and tool switches cannot occur while a job is in progress. The objective is to minimize the makespan C_{max} , which is the duration between the start of processing of the first job and the completion of the last job.

For the Position-Based MILP as detailed in [7] and [6], the following set of decision variables is considered: binary variables x_{jkm} are equal to one if job j is processed in the k th position on machine m , binary variables v_{tkm} are equal to one if tool t is present on machine m during the processing of the job in the k th position, and binary variables w_{tkm} are equal to one if tool t is inserted into the tool magazine of machine m immediately before processing the k th job. Continuous variables f_{jkm} are also considered to define the completion time of job j in the k th position on machine m to compute the makespan C_{max} .

III. ANALOGY WITH REAL-WORLD CASES

In this section, we provide an analogy between our theoretical problem, the SSP-NPM-SPT, and a real-world example of Computer Numerical Control (CNC) machine operations. In the CNC machines, we have a turret that holds various tools needed for different jobs, it's analogous to unrelated machines, where each machine has its specified characteristics. The turret's capacity limits the number of tools it can hold at once, and it's analogous to machine capacity. Each job in the SSP-NPM problem corresponds to manufacturing a piece in a real-world CNC operation. The processing time for each job is influenced by many factors such as tools and materials. In the case of CNC machines, the changing tools require significant time and impact the efficiency of the process. So, with an optimized tool sequencing, we can improve production efficiency. Besides, constraints such as tool wear or corrective maintenance and factors such as raw material or tool types introduce uncertainty into job duration. Understanding these analogies helps us appreciate the complexity and practical implications of the SSP-NPM-SPT problem.

Table I shows an instance for the SSP-NPM and Figure 1 provides an optimal solution with a makespan $C_{max} = 14$.

TABLE I
SSP-NPM INSTANCE, WITH $J = 5, T = 14$ AND $M = 2$ WITH
 $C_1 = 6, C_2 = 5, sw_0 = 2$ AND $sw_1 = 2$.

Jobs	1	2	3	4	5
	1	3	2	11	7
Tools	7	10	10	12	9
	12		4		14
p_{j1}	6	7	6	1	2
p_{j2}	6	8	9	1	9

IV. RELATED WORKS

This SSP-NPM in its deterministic variant was first tackled in [7], [5] through three MILP formulations and an ILS heuristic. More recently, [11] developed a new hybrid algorithm that combines a genetic algorithm (GA) and a local search (LS) technique. In the same way, [19] proposed a Biased random-key GA. Its resolution remains a challenge in the context of the FMS.

To the best of our knowledge and among the existing research on scheduling under uncertainty, no previous studies

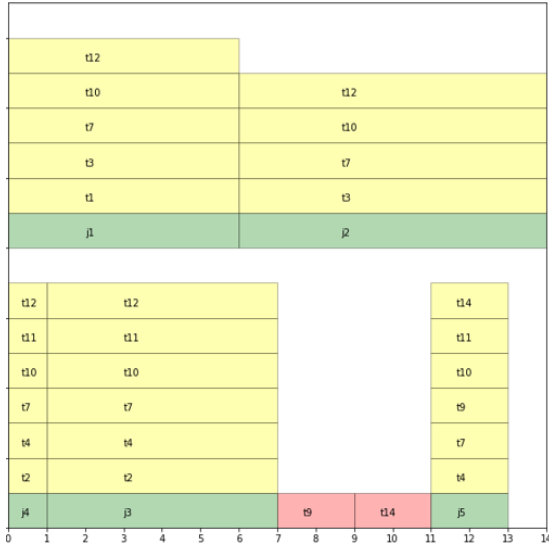


Fig. 1. An optimal solution to the example in Table I, with a makespan $C_{max} = 14$ and a number of switches $TS = 2$

have explored the simultaneous integration of tool constraints, unrelated parallel machines, and stochastic processing times for solving SSP-NPM issues.

As few works focus on solving the SSP-NPM, here we review some recent works from the scheduling literature dealing with uncertainty. In this regard, [17] examined the distribution of processing time, [16] implemented the SAA method for uncertainty, and [8] applied Monte Carlo and distribution models in the context of the Just-In-Time manufacturing. Recently, [18] developed a Monte Carlo tree search algorithm for flexible job-shop scheduling with stochastic processing times, highlighting the efficiency of this approach in dynamic manufacturing systems. In the same way, [13] explored robust scheduling for job shops with interval uncertainty, emphasizing the importance of considering job processing time variability. [9] provided an overview and new results on handling uncertainty in scheduling problems, discussing robust optimization techniques and their applications.

V. MONTE CARLO-BASED APPROACH

Monte Carlo Simulation (MCS) is a statistical technique to model and analyze complex systems under uncertainty. A recent survey on the MCS-based methods for stochastic optimization was presented by [15]. To solve SSP-NPM-SPT using this approach, we generated a set of job scenarios with stochastic processing times based on predefined probability distributions. The MCS approach is used to understand and measure the impact of uncertainty in processing times on overall system performance to develop robust scheduling strategies. To describe our MCS approach, we first consider the Position-based MILP formulation described in [5]. We introduce then some additional notations as mentioned below:

- ξ : index of scenario.
- S : number of scenarios.
- $\Xi = \{1, \dots, S\}$: set of scenarios.

- $p_{jm\xi}$: processing time of job j on machine m in scenario ξ .
- θ : level of variability for Normal Distribution (ND).
- C_{max}^ξ : makespan of each scenario ξ .
- C_{max^Ξ} : mean of makespan across all scenarios.

A. Distribution Law

while exploring processing times within manufacturing systems, the development of distribution laws plays a crucial role. Inspired by the work of [17], which developed distribution laws for processing times and the recent work of [8], who applied these distribution laws in the Just-In-Time manufacturing scenarios with generated scenarios and a determined level of variability ranging from 0.1 to 0.2, we chose to use the ND law. This choice is appropriate because the level of variability is not very high to reach the potential negative values that could occur with the ND. This ensures that all generated processing times are feasible within the context of our model. The ND is useful when processing times are distributed around the mean but it's less effective for scenarios with high variability. The variance for the ND is defined as:

$$\sigma = p_{jm} \cdot \theta \quad (1)$$

The processing time $p_{jm\xi}$ is modeled as:

$$p_{jm\xi} \sim \text{Normal}(p_{jm}, \sigma^2(j, m)) \quad (2)$$

To ensure a complete analysis, we varied the parameter θ to capture different levels of variability in processing times. Specifically, we tested the values of θ at 0.1, 0.15, and 0.2.

B. Scenario Generation and Evaluation

In the proposed MCS approach, we have opted for several steps. Firstly, we generate a set of scenarios Ξ , each representing a variation of possible processing times for all jobs \mathcal{J} on all machines \mathcal{M} . For each scenario ξ , the processing time $p_{jm\xi}$ for job j on machine m is drawn from the ND with the given variability level. Regarding the number of scenarios S to generate, 30, 40, and 50 scenarios are applied as [16]. The general steps involved in the MCS approach are as follows:

- 1) **Scenario generation:** We generate S scenarios where each scenario represents a possible realization of the processing times for all jobs on all machines. For each scenario ξ , the processing time $p_{jm\xi}$ for job j on machine m in scenario ξ is drawn from an ND with given parameters.
- 2) **Evaluation of each scenario:** For each generated scenario, we evaluate the makespan C_{max}^ξ using the Position-Based MILP model from [7] extended with scenario-based variables.
- 3) **Statistical Analysis:** After evaluating all scenarios, we perform statistical analysis on the results. This includes calculating the mean $\overline{C_{max^\Xi}}$ of the makespan across all scenarios, providing insights into the expected performance of the scheduling system under uncertainty:

$$\overline{C_{max^\Xi}} = \frac{1}{S} \sum_{\xi \in \Xi} C_{max}^\xi \quad (3)$$

- 4) **Result recording:** We save all the results of each scenario resolution in files (.csv) to facilitate further analysis.

In conclusion, the MC-based approach provides a robust framework to solve SSP-NPM-SPT and analyze the impact of uncertainty. By using different variability levels and scenario number generation, this approach helps us to develop effective scheduling strategies that can adapt to other real-world cases from the FMS environments.

VI. COMPUTATIONAL EXPERIMENTS

The proposed MCS approach was implemented in Python 3.11, and the mathematical model was solved using the Gurobi Optimizer version 11.0.2. The experiments were conducted on a single thread of a 1.90 GHz Intel Core i3-4030U processor with 4 GB of RAM running on a PC equipped with the Windows 10 operating system.

A. Instances description

To evaluate our MCS approach, we used a set of 50 instances, where their characteristics are given in Table II. These instances are categorized into classes, *sparse* and *dense*, based on the tool matrix density. Sparse matrices are generated randomly based on the required number of tools per job, $|\mathcal{T}_j|$, using a uniform distribution over the interval $[\min\{\frac{T}{4}, \min\{\frac{T}{2}\}\}]$. However, for dense matrices, $|\mathcal{T}_j|$ is generated over the interval $[\max\{\frac{T}{2}, \max(C_m)\}]$. This approach to instance generation is adopted by [2] and incorporates findings from [3], who explored the impact of the difference between magazine capacity and the number of tools required per job on the performance of mathematical models and heuristics. The instances used are available on <http://doi.org/dj5t>.

TABLE II
INSTANCES CHARACTERISTICS

# Inst.	M	J	T
1	2	5	10
2	2	5	15
3	2	10	10

B. Numerical results and discussion

To assess the performance of the proposed approach, MCSs were applied under three variability levels ($\theta = 0.1, 0.15, 0.2$), and each level was conducted across three numbers of scenarios (30, 40, and 50) to determine the best configuration in terms of numbers of scenarios and variability level. The collection of all results of all possible combinations or configuration facilitate a comparison of the efficiency and responsiveness of the simulations relative to the introduced variability and the number of scenarios and helps us to provide an overview of the robustness of the MCS approach in this specific context.

- 1) *Comparison of Gaps:* Figure 2 displays the gap values across different (θ, S).

$$\text{Gap} = |C_{\max} - \overline{C_{\max}^{\Xi}}| \quad (4)$$

The blue boxes for 30 scenarios exhibit relatively low gaps with minimal variability, particularly for $\theta = 0.1$ and $\theta = 0.2$. The yellow boxes for 40 scenarios show greater variability, especially for $\theta = 0.1$ and $\theta = 0.15$. The red boxes for 50 scenarios indicate higher gaps, especially for $\theta = 0.15$ and $\theta = 0.2$, but a minimal gap for 50-0.1 which might seem promising at first glance. Although these observations suggest that 30 scenario configurations offer the most optimal performance regarding gap minimization, a further examination involving CPU time is necessary for a comprehensive understanding.

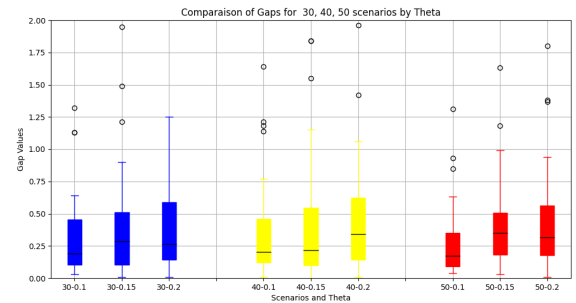


Fig. 2. Comparison of gaps across different θ and S .

- 2) *Gap/CPU Time Ratios for each θ and scenarios number S :* Figure 3 illustrates the gap/CPU time ratios for each configuration. 30 scenarios show lower ratios, indicating high efficiency between gap magnitude and CPU time, particularly for $\theta = 0.1$. Generating 40 and 50 scenarios exhibits generally higher ratios. Notably, the ratio for the case of 50-0.1, despite its lower gap, is significantly higher due to increased CPU time, thus reaffirming the choice of the case 30 scenarios with $\theta = 0.1$ as the more efficient configuration.

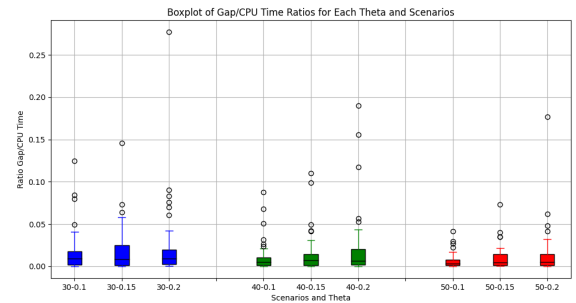


Fig. 3. Gap/CPU time ratios for each configuration.

- 3) *Pareto Chart of Adjusted Composite Scores for Configurations:* With this Pareto chart in Figure 4, we visualize the composite scores that offer the possibility to evaluate each configuration regarding its efficiency and performance.

The Configurations of 30 scenarios with $\theta = 0.1$ had the lowest scores, indicating superior overall performance. This composite score represents the combination of gap and CPU time into a single metric, where a lower score indicates better performance. This lower score indicates better performance because we have a negative weighting of the gap and inverted CPU time favors configurations with low gaps and low CPU times.

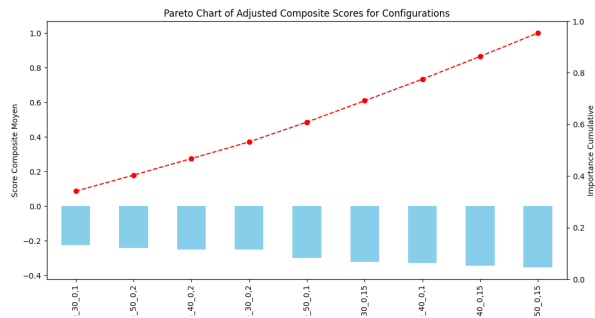


Fig. 4. Pareto chart of adjusted composite scores for configurations.

Analyzing the derived graphs shows that the configuration of 30 scenarios with $\theta = -0.1$ is the optimal choice among all other configurations. It doesn't reach just the lowest gaps compared with original results without variation, but it has the best gap/CPU time ratios and the most favorable composite scores. This makes it an ideal configuration to minimize the gap and maximize CPU time efficiency. Thus, we recommend making it a judicious choice to maximize outcomes while minimizing computational costs.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents an MCS approach to solve the SSP-NPM-SPT, where the objective function is to minimize the makespan. ND was employed to generate scenarios for 50 benchmark instances adapted from the SSP-NPM literature. In our study, we varied the number of scenarios between 30, 40, and 50 and the variability levels between 0.1, 0.15, and 0.2. This variation influences the computation time needed to solve the problem, it increases with the difficulty of instance.

Numerical results show that this approach allows a deeper understanding of operational dynamics under uncertainty, but also improves planning efficiency by identifying optimal conditions for the SSP-NPM-SPT. This real-world applicability ensures that the developed framework can be directly used to improve the agility and responsiveness of modern manufacturing environments.

Future research focuses on using other distribution laws in different contexts such as just-in-time manufacturing, while considering the proposed solution framework to add more robustness, i.e., solution bounds, reinforcement learning, etc. Nonetheless, extensive experiments and some improvements should be considered to deal with real-world size instances. On the other hand, other stochastic parameters based on

industrial needs, as well as different objective functions, i.e. job tardiness, TFT, etc., should perhaps be considered.

REFERENCES

- [1] Jonathan F. Bard. A heuristic for minimizing the number of tool switches on a flexible machine. *IIE Transactions*, 20(4):382–391, 1988.
- [2] Andreza Cristina Beezão, Jean-François Cordeau, Gilbert Laporte, and Horacio Hideki Yanasse. Scheduling identical parallel machines with tooling constraints. *European Journal of Operational Research*, 257(3):834–844, 2017.
- [3] Alewyn P Burger, CG Jacobs, Jan H van Vuuren, and Stephan Esterhuysen Visagie. Scheduling multi-colour print jobs with sequence-dependent setup times. *Journal of Scheduling*, 18:131–145, 2015.
- [4] Dorothea Calmels. The job sequencing and tool switching problem: state-of-the-art literature review, classification, and trends. *International Journal of Production Research*, 57(15-16):5005–5025, 2019.
- [5] Dorothea Calmels. A comparison of different mathematical models for the job sequencing and tool switching problem with non-identical parallel machines. *International Journal of Operational Research*, 2021. Advance online publication.
- [6] Dorothea Calmels. An iterated local search procedure for the job sequencing and tool switching problem with non-identical parallel machines. *European Journal of Operational Research*, 297:66–85, 2022. Available online 21 May 2021.
- [7] Dorothea Calmels, Chandrasekharan Rajendran, and Hans Ziegler. Heuristics for solving the job sequencing and tool switching problem with non-identical parallel machines. In Bernard Fortz and Martine Labbé, editors, *Operations Research Proceedings 2018*, pages 459–465, Cham, 2019. Springer International Publishing.
- [8] Zhengcai Cao, Chengran Lin, Mengchu Zhou, Chuanguang Zhou, and Khaled Sedraoui. Two-stage genetic algorithm for scheduling stochastic unrelated parallel machines in a just-in-time manufacturing context. *IEEE Transactions on Automation Science and Engineering*, 20(2):936–949, 2023.
- [9] F. Clautiaux, B. Detienne, and H. Lefebvre. Handling uncertainty in scheduling problems: An overview and new results. *Journal of Scheduling*, 2023.
- [10] Yves Crama, Antoon W.J. Kolen, Alwin G. Oerlemans, and Frits C.R. Spieksma. Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6(1):33–54, 1994.
- [11] Tunchan Cura. Hybridizing local searching with genetic algorithms for the job sequencing and tool switching problem with non-identical parallel machines. *Expert Systems with Applications*, 223:119908, 2023.
- [12] H. Djellab, K. Djellab, and M. Gourgand. A new heuristic based on a hypergraph representation for the tool switching problem. *International Journal of Production Economics*, 64:165–176, 2000.
- [13] H. Díaz, J.J. Palacios, I. Díaz, C.R. Vela, and I. González-Rodríguez. Robust schedules for tardiness optimization in job shop with interval uncertainty. *Logic Journal of the IGPL*, 2022.
- [14] M. Hirvikorpi, K. Salonen, T. Knuutila, and O. S. Nevalainen. The general two-level storage management problem: A reconsideration of the kns-rule. *European Journal of Operational Research*, 171(1):189–207, 2006.
- [15] Tito Homem-de Mello and Güzin Bayraksan. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.
- [16] Ming Liu and Rongfan Liu. Risk-averse scheduling-location (scheloc) problem. In *2019 International Conference on Industrial Engineering and Systems Management (IESM)*. IEEE, 2019.
- [17] SWENSETHJ Muralidhar and WILSON. Describing processing time when simulating jit environments. *International Journal of Production Research*, 1992.
- [18] M. Saqlain, S. Ali, and J.Y. Lee. A monte-carlo tree search algorithm for the flexible job-shop scheduling in manufacturing systems. *Flexible Services and Manufacturing Journal*, 35:548–571, 2023.
- [19] Leonardo C.R. Soares and Marco A.M. Carvalho. Biased random-key genetic algorithm for the job sequencing and tool switching problem with non-identical parallel machines. *Computers Operations Research*, 163:106509, 2024.
- [20] C.S. Tang and E.V. Denardo. Models arising from a flexible manufacturing machine, part ii: Minimizing the number of tool switches. *Operations Research*, 36(5):778–784, 1988.