Multi-Objective Sustainable Flexible Job Shop Scheduling Problem: Balancing Economic, Ecological, and Social Criteria

Candice Destouet, Houda Tlahig, Belgacem Bettayeb, Bélahcène Mazari

Multi-Objective Sustainable Flexible Job Shop Scheduling Problem: Balancing Economic, Ecological, and Social Criteria

Candice Destouet[a], Houda Tlahig[a], Belgacem Bettayeb[b], Bélahcène Mazari[c]

[a] *CESI LINEACT, 15c av. Albert Einstein, Villeurbanne, 69100, France*
[b] *CESI LINEACT, 8 bd. Louis XIV, Lille, 59800, France*
[c] *CESI LINEACT, 80, av. Edmund Halley Rouen Madrillet Innovation, Saint-Etienne-du-Rouvray, 76800, France*

Abstract

Industry 5.0 makes it imperative to reevaluate the manner of using resources in manufacturing systems to ensure sustainability. In this context, scheduling problems are encountering new environmental and human-related challenges, and the concept of sustainable scheduling has gained importance, aiming to balance economic, environmental, and human factors. In this paper, we propose two multi-objective mathematical models to simultaneously address these three factors as objective functions. In the first model, we consider the operator safety while using the Occupational Repetitive Action (OCRA) index to assess ergonomic risks related to task execution. The second model includes workers' preferences in terms of machines, shifts and task variety. The objective is to improve the general well-being of workers by proposing a schedule that respects as much as possible their preferences. Both models integrate the travel time of workers and products between machines. To solve these NP-hard scheduling problems, we use the Non-dominated Sorting Genetic Algorithms II and III (NSGA-II and NSGA-III), enhanced with a Q-learning strategy for parameter selection and a variable neighborhood search based on reinforcement learning. The obtained results provide a comprehensive analysis of the interactions between these criteria, demonstrating the capability of such approach to achieve a favorable balance between multiple objectives while addressing the new challenges of production scheduling in Industry 5.0 context.

Keywords: Industry 5.0, Sustainable Flexible Job-shops Scheduling Problem, Multi-objective optimization, Human Centric Manufacturing, Environmental consideration, Sustainability, NSGA-II, NSGA-III,

1. Introduction

Today's Industry 5.0 context is human-centered, and integrates societal and environmental concerns alongside advanced technologies. In this context, manufacturing systems have undergone significant changes, encompassing both organizational and functional transformations. After Industry 4.0 has amplified the digitization of data, facilitated transparent data exchange, and automated production processes, thereby improving the performance of manufacturing systems, Industry 5.0 aims to further enhance these systems' resilience by integrating environmental and social considerations (Xu et al., 2021b). In the industrial context, resilience refers to the capacity of a system to endure disruptions and promptly and autonomously return to its standard operation, with original performance. In Chen and Shi (2021), the authors propose a methodology for assessing resilience based on technical, organizational, social, and economic dimensions in the era of Industrial 4.0. At the organizational level, proactive recovery planning mechanisms across various decision-making levels are deemed essential, particularly concerning operational level for dynamic scheduling..

To address the impact of significant dynamic events on system performance, new scheduling techniques from the so-called "resilient scheduling" paradigm have emerged (Wang et al., 2018). Production scheduling must therefore adapt to incorporate new constraints related to human and environmental factors, employ new decision strategies (dynamic/real-time approaches), and align with the relevant objectives of the "5.0 production line", including minimizing energy consumption and maximizing operator well-being (Parente

et al., 2020). The "classical" production scheduling problems have been widely discussed in the scientific literature, and researchers have been confronted with the complexity of solving them, as they are generally NP-hard. Different mathematical single- and multi-objective optimization models have been suggested and different exact and approximate resolving methods have been proposed to solve them (e.g. Branch-and-Bound, Constraint Programming, Genetic Algorithm, Tabu Search, Ant Colony Optimization, etc.).

In this paper, we consider the scheduling problem of a flexible job shop and we propose a model that takes into account human and environmental factors alongside the traditional economic considerations. For the remainder of the paper, this problem is referred to as Sustainable Flexible Job Shop Scheduling Problem (SFJSSP) (Destouet et al., 2023b). We present two multi-objective optimization models designed to simultaneously optimize various criteria associated with economic, environmental, and societal factors. Thus, we are looking for a solution that concurrently achieves favorable outcomes for all three criteria. In the first model, the makespan (maximum completion time), the total energy consumption and the exposure to musculoskeletal risks of the operators are jointly minimized. This last criterion is represented by the OCRA index, which is a standardized measure used to evaluate operators' exposure to tasks involving various upper limb musculoskeletal risk factors (Occhipinti, 1998). In the second model, the human criterion is substituted by the minimum level of satisfaction among all workers. This criterion, which is to be maximized, is computed on the basis of workers' preferences regarding machines, variety of tasks, and working shifts. To the best of our knowledge, only a few scheduling research works have considered the three sustainability indicators simultaneously in the same multi-objective optimization model (Gong et al., 2018; Hongyu and Xiuli, 2021; Coca et al., 2019). In this paper, we also take into account product transport time as well as worker travel time between processing machines, making our study original and challenging. These transportation constraints are frequently overlooked or oversimplified in the literature, despite their potential to significantly impact the production system performance indicators, such as the makespan (Homayouni and Fontes, 2021) and the energy consumption (Jiang et al., 2022). Neglecting these constraints can introduce time delays to task initiation when workers or products are not available at the designated machines, and increase the overall energy consumption. Additionally, we take into account setup constraints between two jobs on the same machine, along with defining working hours and rest time for each worker.

To solve our multi-objective model, we propose using two non-dominated sorting-based evolutionary algorithms, namely NSGA-II and NSGA-III (Non-dominated Sorting Genetic Algorithm II and III) enhanced with a Q-learning strategy for parameter selection during the offspring creation, and a variable neighborhood search based on reinforcement learning to improve the quality of the solutions. We test our model on different simple and extended instances form literature, which have been tailored to our problem by generating certain parameters associated to the human aspect, travel distances and speeds, and setup times. We evaluate our model's performance by defining and testing some typical worker preference profiles, analyzing how variations in these profiles impact the results. Furthermore, we explore the influence of the number of workers on the OCRA index. This comprehensive analysis provides insights into the robustness and applicability of our proposed model across different scenarios.

The rest of this article will be structured as follows. Section 2 provides a comprehensive review of the current state of the art by exploring transportation constraints and discussing various methodologies used to solve the multi-objective SFJSSP problems. In Sections 3 and 4, we introduce the proposed optimization models and describe the algorithms used to solve them. Section 5 presents the instances and metrics used to test the proposed algorithms and to analyze the obtained results. Section 6 studies the performance of the algorithms used, and section 7 reports our results and sensitivity analysis regarding the number of workers and their different profiles. Section 8 concludes our paper and outlines some research perspectives.

2. Related works

2.1. Sustainable scheduling problems

The growing emphasis on sustainability within the Industry 5.0 context has redirected attention towards energy efficiency and operator well-being considerations (Grabowska et al., 2022). Today, the impact of environmental and social factors on scheduling in manufacturing requires equal analyze alongside economic considerations. As a result, sustainable scheduling has emerged in production systems, necessitating a balance among the economic, the human and the environmental aspects. In Destouet et al. (2023a), the authors provide a comprehensive analysis of the flexible job shop scheduling problem considering both human and environmental factors. Sections 2.1.1 and 2.1.2 summarize the literature on scheduling problems aimed at reducing the energy consumption and improving the human factor, respectively.

2.1.1. Energy consumption considerations

Today, preserving the planet in the face of a growing population and depleting natural resources is a major challenge. Knowing that industrial activities represent a substantial part of global energy consumption and the environmental impact it induces (Dai et al., 2019), taking into account energy consumption as an objective function to be minimized in scheduling problems can significantly contribute to reduce pollution. For instance, using one machine instead of another may increase processing time but reduce energy consumption and gas emissions, which can be decisive for achieving frugal and environmentally friendly scheduling. For many years, industries have predominantly prioritized economic imperatives. However, as societal aspirations and regulatory frameworks evolve, a significant paradigm shift is ongoing. Thus, ensuring a judicious trade-off between economic prosperity and environmental preservation is becoming an imperative for long-term sustainability.

Over the past decade, with a notable emphasis in the last three years, many research articles have focused on the concept of green production. In a comprehensive review by Xiong et al. (2022), 297 articles on scheduling problems published between 2016 and 2021 were analyzed. Notably, 54 of these articles specifically addressed environmental factors, encompassing considerations such as energy consumption and carbon emissions. The most prevalent environmental concern in the literature is the reduction of energy consumption (Destouet et al., 2023a; Li and Wang, 2022), specifically during operational processing (Gong et al., 2018). However, it is important to note that energy consumption can arise from various stages of the production process, including set-up and maintenance operations (Song et al., 2014; Li et al., 2021), loading and unloading products, transportation between machines (Zhang et al., 2018), turning on and off machines (Zhang et al., 2018), operating auxiliary facilities and equipment (Li et al., 2022d), and even during machine idle periods (Zhang et al., 2019).

Beyond energy consumption, a limited number of studies incorporate carbon emissions into their models (Yin et al., 2017a; Xu et al., 2021a), a metric directly linked to energy consumption. In other approaches, Hajibabaei and Behnamian (2022) endeavored to mitigate noise pollution by minimizing machine speed increments, while Gong et al. (2018) introduced a recycling tool ship, both relatively rare in the literature. Typically, environmental models are multi-objective, aiming to minimize energy consumption alongside economic factors such as total cost or makespan. In some studies (Coca et al., 2019; Hongyu and Xiuli, 2021), a third criterion focusing on societal considerations is also minimized, thus achieving a sustainable schedule.

2.1.2. Human factors considerations

While Industry 4.0 introduced automation and intelligent systems, Industry 5.0 aims to shift towards a human-centered approach, fostering a synergy between machines and workers to enhance social prosperity (Nahavandi, 2019). Many studies on production scheduling have traditionally focused on machine resources, overlooking the pivotal role of operators within the model. However, these machines are typically operated

by one or more workers, each possessing distinct skills that influence their speed, work quality, and capability to perform specific operations. Unlike machines, workers require structured work schedules, adequate rest periods, and a safe working environment, rendering them more challenging to model and manage properly. Given their significant impact on system efficiency and resilience, it is imperative to incorporate their considerations, as underscored in recent research (Sindhwani et al., 2022).

The integration of the human factor in the SFJSSP has been relatively underexplored (Gong et al., 2018). However, some studies have addressed worker considerations in scheduling in various ways. Firstly, economic aspects, such as worker costs and skills, have been incorporated. For instance, Gong et al. (2018) propose a model aimed at minimizing labor costs, where the processing cost for each worker is dependent on their skills. Zheng and Wang (2016) and Lei and Tan (2016) present models where workers' varying skill levels influence machine operation and consequently affect processing times. Wu et al. (2018) examine the learning effect on workers, suggesting that increased time spent on a machine leads to improved skills.

Furthermore, worker safety and well-being are critical considerations. Acknowledging the positive impact of worker happiness on reducing turnover, minimizing absenteeism, and enhancing overall system efficiency (Taris and Schreurs, 2009), several studies have focused on these aspects. Berti et al. (2021), Tan et al. (2021), and Sun et al. (2019) address worker fatigue by incorporating rest periods to ensure safety and efficiency. Specifically, Jaber and Neumann (2010) demonstrate that short breaks after each task improve system performance.

For ergonomic risks, various indices are suggested in the literature for different static and dynamic scheduling problems, extensively detailed by Stanton et al. (2004). The most well-known indices include RULA (Rapid Upper Limb Assessment), REBA (Rapid Entire Body Assessment), OCRA (OCcupational Repetitive Action), OWAS (Ovako Working Posture Analysis System), NIOSH (National Institute for Occupational Safety and Health), and EAWS (European Assembly Worksheet). RULA and OCRA focus on upper limbs, while REBA also includes lower limbs. OWAS and EAWS consider the full body. These indices differ in the parameters they assess, such as posture, frequency, or lifted weight. A more detailed comparison of their features can be found in Destouet et al. (2023a). Rinaldi et al. (2022) utilize three of these indices—OCRA, OWAS, and NIOSH—in a single model to comprehensively address ergonomic risks.

In our study, we use the OCRA index, which is considered the most comprehensive for assessing upper limb risks. According to the European standard EN 1005 and the international standard ISO 11228-3, OCRA index can be used to analyze and predict the onset of musculoskeletal disorders. Introduced by Occhipinti (1998), the OCRA index compiles a range of risk factors associated with technical actions, including posture, frequency, strength, lack of rest, total duration, and additional risks like vibration or temperature. Technical actions refer to basic manual tasks performed during the work cycle, such as holding, turning, and pushing (Akyol and Baykasoğlu, 2019). During one shift, a worker's OCRA index is calculated as the ratio between the Actual number of performed Technical Actions (ATA) and the Recommended number of Technical Actions (RTA): $OCRA{=}ATA/RTA$.

The recommended number of technical actions is formulated as $RTA{=}\ CF \times PM \times FM \times RM \times ARF \times (RcM \times DuM)$. This calculation incorporates several multipliers, each ranging between 0 and 1 based on the specifics of each task: PM (posture multiplier), FM (force multiplier), RM (repetitiveness multiplier), and ARF (additional risks multiplier). Additionally, it includes multipliers independent of specific tasks but dependent on the work environment: RcM (lack of recovery multiplier) and DuM (overall duration multiplier). The constant of frequency of technical actions per minute, CF, is set at 30.

Then, the OCRA index for each worker is a real, positive and unitless numerical value, providing a measure of his ergonomic risk exposure. Şenyiğit et al. (2022) and Stanton et al. (2004) define six levels of risks depending on OCRA index, presented in Figure 1.

From another perspective, some studies incorporate workers' preferences into their models. Ruiz-Torres et al. (2015) maximize the on-time delivery jointly to worker satisfaction, which is represented by job prefer-

Figure 1: Risks of musculoskeletal disorders depending on OCRA Index

ences. In a subsequent publication, Ruiz-Torres et al. (2019) strike a trade-off between makespan and worker satisfaction, demonstrating the feasibility of finding schedules that effectively address both objectives. In this study, workers select a preference level, indicating their liking or disliking for each machine along with a desired level of variety in an identical parallel machine scheduling problem. Mohan (2008) maximize employee satisfaction by taking into account their seniority and shift preferences. Akbari et al. (2013) maximize workers' satisfaction considering workers' availability, productivity, priority preference, seniority level and required number. In the study of Bezoui et al. (2023), the decision-maker preferences are integrated when addressing a multi-objective flexible job shop scheduling problem. The outcomes reveal a discernible improvement in the quality of solutions achieved through this integration. All these studies highlight the significance of workers' well-being at work, underlining that it should be taken into account beforehand when establishing production schedules, as "happy workers are also productive workers" (Taris and Schreurs, 2009).

2.2. Scheduling problems with internal transport constraints

While investigating scheduling problems, many research papers neglect the internal transportation times and their induced constraints (Zhang et al., 2021; Luo et al., 2022). However, it is important to recognize that these product movements are time-consuming and that neglecting them can lead to delays and altered overall process times (Homayouni and Fontes, 2021). Jiang et al. (2022) analyze the impact of transportation on completion time and energy consumption, establishing a strong connection between these variables. Sanogo et al. (2023) underscored the significance of considering internal transportation times as equally crucial as task scheduling within a workshop. They focused on scheduling transportation tasks carried out by Autonomous Guided Vehicles (AGVs), aiming primarily to prevent collisions and deadlocks. Ebrahimi et al. (2020) prove that considering transportation time can reduce the energy consumption and tardiness penalty by 24% and may avoid idle times of machines. Li et al. (2020), Zhang et al. (2019) and Zhang et al. (2021) addressed the flexible job shop scheduling problem while accounting for transportation time constraints. They aimed to minimize completion time and reduce energy consumption in a flexible workshop. In their model, the authors considered energy consumption during task execution as well as during transport within the workshop. Li et al. (2022d) study the same problem with insufficient transportation resources in a dynamic context, allowing new job insertions and machine breakdowns.

On the other hand, incorporating multi-skilled operators into the workshop cannot be effective without introducing additional constraints related to travel times. Given their flexibility, operators may be assigned to different machines, which means that the schedule has to take into account the time needed for a worker to move from one machine to another. These constraints can be viewed as workforce routing problems. Workforce scheduling and routing problems are discussed in the review by Castillo-Salazar et al. (2016). Operators' travel time is often overlooked in the design and operation of manufacturing processes. However, Tan et al. (2021) study the fatigue levels of workers during the manufacturing process, assuming that travel time does not count as working time or rest time and that it has a neutral effect on the level of fatigue.

2.3. Multi-objective scheduling problems

Multi-objective problems (MOPs) are thoroughly documented in the literature, and various methods have been proposed to address them. Chiandussi et al. (2012) explain four approaches: the global criterion method, the linear combination with weights, the ϵ-constrained method, and the multi-objective genetic algorithms.

- The global criterion method: This approach has the advantage of being simple and efficient (Sadeghian et al., 2019). Its principle is based on minimizing the distance of the solutions from an ideal solution.

Nevertheless, it is computationally expensive and does not always obtain non-dominated solutions in the Pareto sense.

- The linear weight combination method: In this approach, the objective function is written as a weighted sum of all the considered objectives. We thus pass from a multi-objective model to a single-objective one. The difficulty with this method is to find the weights for each objective and it only allows to obtain a small number of solutions (Ding et al., 2021).

- The ϵ-constraint method: This approach also keeps only one objective and transforms the others into constraints. It is easy to implement and enables a rich representation of the efficient set (Dou et al., 2020). The main disadvantages of this approach is its computational time and the difficulty in certain cases to encode the objectives.

- The multi-objective evolutionary algorithms: This approach is widely used in the literature. In these algorithms, the procedure starts from an initial population of solutions, and thanks to certain operators, generally bio-inspired, the population evolves over the iterations. The expected ultimate goal is that the population converges to the Pareto-front representing the set of non-dominated solutions (Zhang et al., 2020).

Among the evolutionary algorithms, two variants of the Non-dominated Sorting Genetic Algorithm (NSGA) caught our attention to solve the SFJSSP problem: (1) NSGA-II (Deb et al., 2002) which is used many times and is often improved when combined with other methods (local search, meta-heuristics, *etc.*) (2) NSGA-III which was proposed by Deb and Jain (2014), and differs from NSGA-II in its selection mechanisms. As mentioned by Campos Ciro et al. (2016), these two algorithms have shown good results and seem to be efficient for problems like ours. In this study, the authors tested both algorithms on different instance sizes and demonstrated that NSGA-III consistently outperforms NSGA-II in terms of the hypervolume metric for each instance size. Coca et al. (2019) compared the performances of NSGA-II and NSGA-III using a case study with three different metrics: the number of non-dominated solutions, the uniform distribution of the non-dominated solutions, and the level of non-dominated solutions. The authors found that NSGA-III outperforms NSGA-II across all three metrics. Both this study and the one by Hongyu and Xiuli (2021) include in their objective functions the three aspects considered in this paper, namely: the economic aspect (makespan), the environmental aspect (energy consumption or carbon emissions), and the societal aspect (ergonomic risks). Hongyu and Xiuli (2021) employed an improved NSGA-III, noting that while NSGA-III inherits the framework and advantages of NSGA-II, it also integrates a selection strategy based on reference points, ensuring the convergence and diversity of the next generation of individuals. Gong et al. (2018) and Zhu et al. (2020) use other evolutionary algorithms, namely a new hybrid genetic algorithm and the memetic algorithm, to find a solution ensuring the best trade-off between human, environmental and economic aspects.

With the advent of machine learning and artificial intelligence, new methodologies have emerged that greatly enhance traditional computational techniques. Evolutionary algorithms can be enhanced with reinforcement learning, a field of machine learning that allows for adaptive parameter adjustment. While Han et al. (2023) demonstrate that reinforcement learning has been more intensively studied in single-objective evolutionary algorithms, recent research has increasingly focused on integrating these methods into multi-objective evolutionary algorithms. For instance, Li et al. (2022a) introduce a multi-objective evolutionary algorithm based on decomposition, which incorporates a Q-Learning-based parameter adaptation strategy to improve diversity. Similarly, Chen et al. (2023) employ a multi-objective immune algorithm guided by Q-Learning to optimize neighborhood selection. Both studies emphasize that integrating Q-Learning enhances both exploitation and exploration capabilities, leading to superior results across all evaluated metrics.

More recently, reinforcement learning models have been also developed to select decision rules for multi-objective problems (Wang et al., 2022). These kinds of Artificial-Intelligence-based algorithms can be particularly efficient for dynamic scheduling problems.

3. Problem description

3.1. Assumptions

In this study, we propose two non-linear integer programs for solving the SFJSSP which will be referred to as 'First model' (or MOP1) and 'Second model' (or MOP2) throughout the remainder of this paper. MOP1 aims to minimize, simultaneously, the makespan (f_1), the energy consumption (f_2) and the maximum OCRA index among all workers (f_3). MOP2 seeks to minimize the makespan (f_1) and the energy consumption (f_2), while maximizing the minimal satisfaction of all workers (f_4). In our study, sustainability encompasses both environmental and human considerations. For the environmental aspect, this involves accounting for energy consumption. For the human aspect, it includes knowledge, safety, and workers' preferences.

We consider three types of energy-consuming activities: i) execution of operations on machines, ii) internal transportation of products, and iii) operation of auxiliary installations (lighting, heating, *etc.*). We presume that, for a given schedule, the energies consumed by auxiliary installations are directly correlated with the makespan, because longer processing time require more energy for lighting, heating, and other related purposes. The energy consumption during processing operations depends on the specific operation types and on the machines and workers involved in each operation. Additionally, we assume that energy consumption during transportation is solely determined by the distance between machines and remains consistent across all products.

Regarding the human consideration, we focus not only on workers' skills, but also on their safety and well-being through: i) their rest time and OCRA index in the first model (MOP1) and ii) their satisfaction in the second model (MOP2). For each worker and each time period, the OCRA index is computed based on the number of technical actions and multiplying factors related to posture, force, repetitiveness, additional risks (such as vibration or temperature), lack of rest, and overall process duration. According to the European standard EN 1005 (CEN/TC122, 2008) and the international standard ISO 11228-3 (ISO/TC159/SC3, 2007), the OCRA method is recommended to analyze, predict and prevent the onset of musculoskeletal disorders.

In the formulation of MOP1, we aim to minimize the OCRA index for each worker among other objectives. In the MOP2 formulation, we impose the OCRA index to be less than 2.2, as recommended by ISO 11228-3, to ensure an acceptable level of risk exposure for workers. The MOP2 aims to maximize workers' satisfaction by accounting for three dimensions of preferences regarding: i) machines, which is a positive real ranging from 0 to 1, reflecting the degree of liking to work on a given machine, ii) tasks variety, which is also a positive real between 0 and 1 representing the proportion of different machines to which a worker likes to be assigned, and iii) working shift, which varies in the same range as previous dimensions, with values close to 0 indicating a preference for morning periods, values close to 1 indicating a preference for afternoon periods, and a number near 0.5 indicating a neutral preference.

The FJSSP is basically defined by a set of jobs $\mathcal{J} = \{j_1, j_2, ..., j_J\}$, each one divided in a set of operations $\mathcal{O}_j = \{O_{j1}, O_{j2}, ..., O_{jK_j}\}$ to be executed on a set of machines $\mathcal{M} = \{m_1, m_2, ..., m_M\}$ in a given sequence. In the FJSSP we considered in this paper, operations must also be assigned to a worker. We assume that every worker is capable of performing any task. However, the time required to complete a task is contingent on the individual worker assigned to it. Indeed, a more skilled worker will be faster. Then, we have to handle three sub-problems: optimizing the assignment of both machine and worker for each operation, and the optimal sequence for each machine and each worker.

To develop our mathematical model, we made the following assumptions. We note that these assumptions can be customized to align with the specific rules and regulations of the factory or company.

- each operation requires exactly one operator and one machine to be processed;
- machines and operators can perform one operation at a time;
- an operator rests 8 % of his working time after each operation;
- the time horizon is composed of two shifts of 7 hours per day.

3.2. Mathematical model

3.2.1. Notations

The notations used to formalizing our mathematical models are detailed below.

Sets, indices and parameters

$\mathcal{W}$	set of workers, $w = 1..W$.
$\mathcal{M}$	set of machines, $m = 1..M$.
$\mathcal{J}$	set of jobs, $j = 1..J$.
$\mathcal{K}_j$	set of operations of j noted O_{kj}, with $k = 1..K_j \ \forall j \in \mathcal{J}$.
$\mathcal{T}$	set of periods, $t = 1..T$.
$start_t, end_t$	starting and ending time of period t .
p_{kjwm}	processing time of operation O_{kj} on machine m with worker w.
$s_{kjmk'j'}$	setup time of machine m between operations o_{kj} and o_{kj}.
$tp_{mm'}$	transport time of a product between machines m and m'.
$t_{mm'}$	travel time for a worker between machines m and m'.
e_{jkwm}	energy consumption to execute operation O_{kj} on machine m with worker w.
$e^{trans}_{mm'}$	energy consumption to transport a product from machine m to machine m'.
e^{aux}	energy consumption for auxiliary installations by time unit.
ata_{kjm}	number of technical actions to execute operation O_kj on machine m.
pm_{kjm}	posture multiplier when executing O_{kj} on machine m.
rm_{kjm}	repetitiveness multiplier for executing O_{kj} on machine m.
fm_{kjm}	strength multiplier for executing O_{kj} on machine m.
arf_{kji}	additional risks multiplier for executing O_{kj} on machine m.
rcm	recovery period multiplier.
dum	duration of repetitive task multiplier.
cf	constant frequency of technical actions per minute.
p^{mac}_{wi}	preference level of worker w for machine i.
p^{var}_{w}	preference level of variety for worker w.
p^{shift}_{w}	shift preference for worker w.

Decision variables

S_{kj}	starting time of operation O_{kj} ;
C_{kj}	ending time of operation O_{kj} ;
$Cmax$	total processing time, makespan.
R_{kjwt}	rest time of worker w after operation O_{kj}.
V_{tw}	binary variable equal to 1 if worker w is assigned to period t.
X_{kjwmt}	binary variable equal to 1 if O_{kj} is executed on machine m by worker w in period t.
$Y_{kjwk'j'}$	binary variable equal to 1 if O_{kj} is executed by worker w before $O_{k'j'}$.
$Z_{kjmk'j'}$	binary variable equal to 1 if O_{kj} is executed on machine m before $O_{k'j'}$.
$N_{kjmk'j'm'}$	binary variable equal to 1 if O_{kj} is executed on machine m and $O_{k'j'}$ on machine m'.
ET	total energy consumption for transport.
EM	total energy consumption for operations.
EC	total energy consumption for auxiliaries installation.
ATA_{wt}	real number of technical actions for worker w in period t.
RTA_{wt}	recommended number of technical actions for worker w on period t.
$OCRA^{max}$	OCRA maximum for all workers and all periods.
A_{wi}	number of times worker w is assigned on machine i.

B_{wi}	binary variable equals to 1 if worker w is at least once on machine i.
Sat_w^{mac}	level of satisfaction of worker w for machine assignment.
Sat_w^{var}	level of satisfaction of worker w for variety.
Sat_w^{shift}	level of satisfaction of worker w for shift assignment.
Sat_w^{total}	level of satisfaction total of worker w.
Sat^{min}	minimum level of satisfaction of all workers.

3.2.2. Objective functions

As previously mentioned, we are focusing on four objectives and have developed two models to address them. By minimizing these objectives simultaneously, we aim to achieve an optimal solution that balances production efficiency, energy efficiency, and operator safety and well-being. The specific objective functions considered are as follows:

$$f_1 : Cmax \tag{1}$$

$$f_2 : ET + EM + EC \tag{2}$$

$$f_3 : OCRA^{max} \tag{3}$$

$$f_4 : -Sat^{min} \tag{4}$$

In the first model (5), we minimize f_1, f_2 and f_3 simultaneously. In the second model (6), we replace f_3 by f_4, while always considering f_1 and f_2.

$$MOP1 : \min \ \{f_1, f_2, f_3\} \tag{5}$$

$$MOP2 : \min \ \{f_1, f_2, f_4\} \tag{6}$$

3.2.3. Constraints

The constraint (7) defines the makespan, as the completion time of the last operation in the schedule. Constraints (8), (9) and (10) compute energy consumption linked, respectively to internal transport operations, processing operations, and the functioning of auxiliary installations.

$$Cmax \geq C_{kj} \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \tag{7}$$

$$ET = \sum_j \sum_k \sum_m \sum_{m'} N_{kjm(k+1)jm'} \times e_{mm'}^{trans} \tag{8}$$

$$EM = \sum_j \sum_k \sum_m \sum_w \sum_t X_{kjwmt} \times e_{kjwm} \tag{9}$$

$$EC = Cmax \times e^{aux} \tag{10}$$

Equations (11) to (15) represent the constraints applicable to workers, shared across both models. Equation (11) guarantees that each worker is allocated once in two successive time periods, i.e. the same working shift, Equation (12) ensures their rest time after each operation performed. Equations (13) and (14) compute the real and recommended number of technical actions for each worker at each period, with the ratio between these counts representing the OCRA index, and Equation (15) defines the $OCRA^{max}$ as the maximum OCRA

index reached among all workers during all time periods.

$$V_{tw} + V_{(t+1)w} = 1 \qquad \forall t \in \mathcal{T}, w \in \mathcal{W} \qquad (11)$$

$$R_{kjwt} = 0.08 \times \sum_{m} X_{kjwmt} \times p_{kjwm} \qquad \forall t \in \mathcal{T}, w \in \mathcal{W}, j \in \mathcal{J}, k \in \mathcal{K}_j \qquad (12)$$

$$ATA_{wt} = \sum_{j}\sum_{k}\sum_{m} ata_{kjm} X_{kjwmt} \qquad \forall w \in \mathcal{W}, t \in \mathcal{T} \qquad (13)$$

$$RTA_{wt} = rcm \times dum \times \sum_{j}\sum_{k}\sum_{m} cf \times pm_{kjm} rm_{kjm} fm_{kjm} arf_{kjm} \times X_{kjwmt} p_{kjwm} \qquad \forall t \in \mathcal{T}, w \in \mathcal{W} \qquad (14)$$

$$OCRA^{max} \geq \frac{ATA_{wt}}{RTA_{wt}} \qquad \forall t \in \mathcal{T}, w \in \mathcal{W} \qquad (15)$$

Equations (16), (17) and (18) are precedence-related constraints linked to, respectively, two operations of the same job, two operations assigned to the same machine, and two operations assigned to the same worker.

$$S_{(k+1)j} \geq C_{kj} + \sum_{m}\sum_{m'} N_{kjm(k+1)jm'} \times tp_{mm'} \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \qquad (16)$$

$$S_{k'j'} + (1 - \sum_{m} Z_{kjmk'j'})M \geq C_{kj} + s_{kjmk'j'} \qquad \forall j, j' \in \mathcal{J}, k, k' \in \mathcal{K}_j, \mathcal{K}_{j'} \qquad (17)$$

$$S_{k'j'} + (1 - \sum_{w} Y_{kjwk'j'})M \geq C_{kj} + \sum_{w}\sum_{t} R_{kjwt} + \sum_{m}\sum_{m'} N_{kjm(k+1)jm'} \times t_{mm'} \qquad \forall j, j' \in \mathcal{J}, k, k' \in \mathcal{K}_j, \mathcal{K}_{j'} \qquad (18)$$

Finally, constraint (19) states that an operation is executed exactly by one worker and one machine, and constraints (20) and (21) ensure that an operation starts and ends at the same period.

$$\sum_{w}\sum_{m}\sum_{t} X_{kjwmt} = 1 \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \qquad (19)$$

$$C_{kj} - (1 - \sum_{w}\sum_{m} X_{kjwmt})M \leq end_t \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_j, t \in \mathcal{T} \qquad (20)$$

$$S_{kj} + (1 - \sum_{w}\sum_{m} X_{kjwmt})M \geq start_t \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_j, t \in \mathcal{T} \qquad (21)$$

In the second model, we add the constraints (22) to (27). First, we ensure that $OCRA$ index does not exceed 2.2 for each worker and period with constraint (22). The constraints (23), (24) and (25) compute for each worker the level of satisfaction regarding, respectively, machine affectation, variety and shift assignment, while the constraints (26) compute the total level of satisfaction for each worker and constraints (27) define the minimum level of satisfaction among all workers.

$$OCRA^{max} \leq 2.2 \qquad (22)$$

$$Sat_w^{mac} = \frac{\sum_i p_{wi}^{mac} \times A_{wi}}{\sum_i A_{wi}} \qquad \forall w \in \mathcal{W} \qquad (23)$$

$$Sat_w^{var} = 1 - \frac{|\sum_i B_{wi} - p_w^{var} \times |M||}{|M|} \qquad \forall w \in \mathcal{W} \qquad (24)$$

$$Sat_w^{shift} = (1 - p_w^{shift}) \times V_{0w} + p_w^{shift} \times V_{1w} \qquad \forall w \in \mathcal{W} \qquad (25)$$

$$Sat_w^{total} = \frac{Sat_w^{mac} + Sat_w^{var} + Sat_w^{shift}}{3} \qquad \forall w \in \mathcal{W} \qquad (26)$$

$$Sat^{min} \leq Sat_w^{total} \qquad \forall w \in \mathcal{W} \qquad (27)$$

Table 3 summarizes the two multi-objective optimization models we consider.

Regarding the problem's complexity, it has been demonstrated that FJSSP is NP-hard, and thus, it can be deduced that SFJSSP shares this NP-hardness (Hongyu and Xiuli, 2021). Solving such a complex problem with an exact method is feasible only for small-scale instances. To obtain a high-quality solution close to the optimal one in a reasonable time, we have implemented two types of non-dominated sorting-based

	First model (MOP1)	Second model (MOP2)
Objective functions	f_1, f_2, f_3	f_1, f_2, f_4
Constraints	(7) to (21)	(7) to (27)

Table 3: Summary of the two models

evolutionary algorithms: NSGA-II and NSGA-III. These algorithms are described in the next section.

4. Algorithms description

The non-dominated sorting genetic algorithms have been widely used to solve multi-objective problems, in particular for scheduling problems (Amjad et al., 2018; Yang et al., 2016; Song et al., 2014). These algorithms are based on the Pareto-dominance principle, explained in section 4.5.1.

The basic framework of these algorithms is as follows: we start with a population P_0 composed of N solutions (see section 4.2). At each iteration, we create an offspring population Q_t of size N thanks to crossover and mutation operators (see section 4.3). We gather P_t and Q_t and obtain a population R_t that contain 2N solutions. Then, we keep the N best solutions to composed the next population P_{t+1} of size N. These solutions are chosen thanks to their dominance level (see section 4.5.1) to ensure elitism, added to other methods presented in sections 4.5.2 and 4.5.3 to ensure diversity. The next sub-sections describe in details how the algorithms work.

4.1. Chromosome coding

A solution is described by a three-layer chromosome, representing our three sub-problems, as shown in Figure 2. The first one represents the operations sequence (OS), which is a repetition of job indices. The k^{th} occurrence of a job index j in the sequence means o_{kj}. The second and third layers represent the solutions of machine allocation and operator allocation problems, respectively (MA and WA). The code of MA comprises numbers ranging from 1 to M (indices of machines), designating the specific machine assigned to execute the operation. Similarly, WA encompasses numbers between 1 and W (indices of workers), indicating the worker assigned to carry out the operation. For the solution coding of MOP2, an additional information is introduced in WA, specifically regarding shift assignments. For instance, the notation 20 indicates that worker 2 is assigned to the respective operation during the morning shift 0, whereas 21 designates worker 2 during the afternoon shift 1. This additional information is necessary, as all tasks that are assigned to the same worker has to be assigned to the same shift. In Figure 2, O_{11} is executed on machine 2 by worker 2 on an afternoon shift, O_{12} on machine 1 by worker 1 on a morning shift, etc. Note that all operations assigned to the same workers are assigned to the same shift.

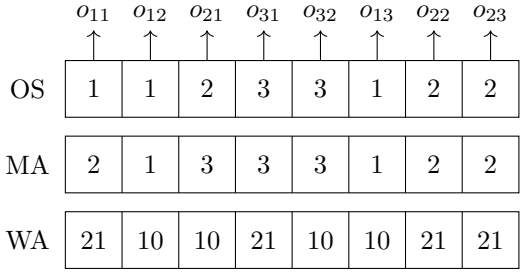


Figure 2: Genes representing our solutions (the sequence of operations OS, the machines allocation MA, and the workers allocation WA)

4.2. Initialization

At the beginning of our algorithm, we initialize the initial population using heuristics adapted to each mathematical model, as shown in table 4. In both models, the population is divided into quarters: one

quarter is initialized randomly, another quarter uses a processing time heuristic, another quarter uses an energy consumption heuristic, and the final quarter relies on a human factors heuristic. This diversified initialization strategy aims to produce high-quality solutions through heuristics while introducing variability through random generation. In addition, we implement a multi-start approach, in which several initial solutions are generated throughout the execution of the algorithm, guaranteeing resistance to local optima and allowing new solution paths to be explored.

<table>
<tr><td colspan="2"></td><td colspan="4">N individuals</td></tr>
<tr><td colspan="2"></td><td>25%</td><td>50%</td><td>75%</td><td>100%</td></tr>
<tr><td rowspan="3">First model</td><td>OS</td><td colspan="4">randomly generated</td></tr>
<tr><td>MA</td><td rowspan="3">randomly generated</td><td rowspan="2">for each task, select the pair machine/worker that minimize processing time</td><td rowspan="2">for each task, select the pair machine/worker that minimize energy consumption</td><td rowspan="2">for each task, select the machine that minimize OCRA and assign it to the operator with the lowest OCRA</td></tr>
<tr><td>WA</td></tr>
<tr><td rowspan="3">Second model</td><td>OS</td><td colspan="4">randomly generated</td></tr>
<tr><td>MA</td><td rowspan="3">for each task, select the pair machine/worker that minimize processing time</td><td rowspan="3">for each task, select the pair machine/worker that minimize energy consumption</td><td rowspan="3">for each task, select the worker with the lowest preference level and the machine he prefers</td></tr>
<tr><td>WA</td></tr>
<tr><td>SA</td><td>Assign workers that make the first operations to the morning periods</td></tr>
</table>

Table 4: Population initialization method

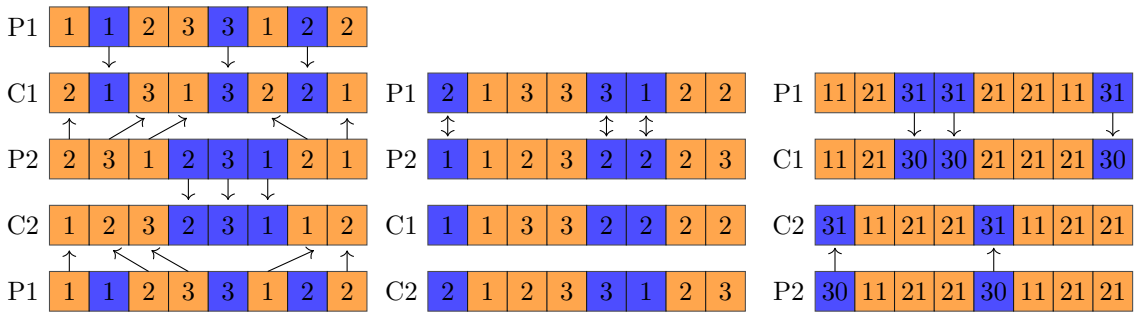
4.3. Offspring generation

At the commencement of each iteration, we create an offspring population, denoted as Q_t, consisting of N individuals derived from the existing parent population P_t. The generation of this offspring population is accomplished through the application of crossover and mutation operators, which are detailed in subsections 4.3.1 and 4.3.2. The offspring generation aims to explore new solutions, trying to improve the parent population through two distinct approaches. Firstly, by merging promising solutions within the parent population, thereby drawing on the strengths of several alternatives. Secondly, by moving away from the parent population, where genes are randomly exchanged to encourage exploration and discovery of new solutions. In the next iteration, we will retain the best solutions from the population of parents and offspring.

4.3.1. Crossover operators

Crossover operators are used to explore the neighborhood of good solutions by exchanging genes from two different solutions. To choose which solutions to crossover, we use a tournament selection. It randomly selects k solutions, compares them with the crowded comparison operator (4.5.2), and keeps the two best chromosomes to cross them. In our algorithm, we first sequentially apply a crossover on OS, then a crossover on MA, and finally on WA, not necessarily on the same operation for each of them. For the operation sequence, we apply the crossover operators as described in Figure 3a. We randomly select operations, here o_{12} ,o_{32} and o_{22}, and we keep their positions on the parent 1 for the child 1, and their positions on the parent 2 for the child 2. We complete the child 1 with the missing operations sequenced as on the parent 2 and the opposite for the child 1. For the crossover on the machines allocation, we randomly select operations and exchange their machine allocation on the parent 1 with their machine allocation on the parent 2 (see Figure 3b). When conducting crossover on the chromosome layer WA, we first apply a crossover similar to MA, exchanging worker allocation for different operations. Then, we add another crossover for shift assignment. We

randomly designate operators and interchange their shifts between two chromosomes, thereby automatically altering the shift assignment for their respective tasks. For a visual presentation, Figure 3c demonstrates the shift exchange involving worker 3.

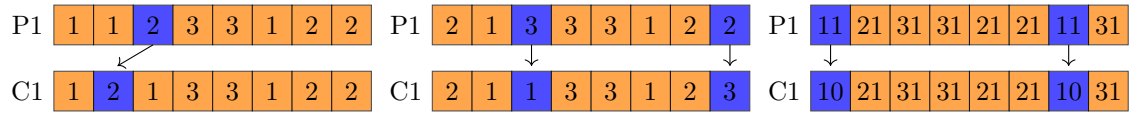


(a) Crossover operator for OS **(b)** Crossover operator for MA/WA **(c)** Second crossover operator for WA

Figure 3: Crossover operators

4.3.2. Mutation operators

Mutation is a technique used to diversify solutions and avoid local optimum. The mutation operations are applied after the crossover operations with a probability p. For OS, we randomly select an operation, and move it to a random other place, as shown in Figure 4a. For MA, we randomly select an operation and replace its machine by another randomly selected one (Figure 4b). For WA, we proceed a first mutation similar to MA for worker mutation, and a second one for shift mutation. For this last one, we randomly select workers and change their shift, and so the shift assignment for their respective task (4c).



(a) Mutation operator for OS **(b)** Mutation operator for MA/WA **(c)** Second mutation operator for WA

Figure 4: Mutation operators

4.3.3. Q-learning method for parameters setting

In the sections 4.3.1 and 4.3.2, we described the crossover and mutation operators to create the offspring solution R_t. These operators depend on different parameters:

- First, in the process of selecting solutions for crossover, we employ a tournament selection method, introducing a parameter denoted as k that specifies the number of solutions to be compared. A larger k entails running the tournament on a larger pool of solutions, which tends to favor the selection of high-quality solutions. Conversely, a smaller k results in a comparison among a randomly chosen subset of solutions from the parent population P_t, potentially allowing for the selection of solutions that are less efficient but contributing to diversity. In the former scenario, we emphasize elitism, while in the latter our focus is on promoting diversity and exploring uncharted solution spaces.
- Secondly, the mutation process is applied to a chromosome with a specified probability denoted as p. When p is set close to 0, the probability of retaining our already favorable solution is heightened. Conversely, when p approaches 1 the probability of inducing a mutation in the chromosome increases, thereby facilitating the exploration of novel solution domains. As before, the p parameter plays a pivotal role in either reinforcing elitism or encouraging diversity within the population.

To efficiently set the parameters k and p, we implement a Q-learning method, which possesses the capability to dynamically adjust these parameters in response to the specific state or context at hand. The Q-learning method operates as follows: within a given state, an agent selects a particular action. Following the outcome of this action, a corresponding reward is assigned to the state/action pair, and the agent continuously updates its knowledge throughout the course of the process, as described in (Watkins and Dayan, 1992). The states, actions, and rewards employed in our algorithms are detailed in the rest of this section.

Actions: the various potential actions are distinct settings for our parameters. Specifically, we establish four possible values for each parameter: k can assume values from the set $\{0.125, 0.17, 0.25, 0.5\}$ (this parameter will be multiplied to the population size), and p can be chosen from the set $\{0.2, 0.4, 0.6, 0.8\}$. During each iteration, the algorithm selects a single value for each parameter.

States: We define four distinct states based on the notions of convergence and diversity. In each iteration t, the convergence metric CV_t is computed using the function *SetCoverage*(P_1,P_2) which quantifies the percentage of solutions in population P_1 that are dominated by at least one solution in population P_2 (see section 5.1). Simultaneously, the diversity metric DV_t is calculated in accordance with the methods outlined in section 5.1, relying on the inter-solution distances within a population. Then, $\Delta CV = \frac{SetCoverage(P_t,P_{t-1})}{SetCoverage(P_{t-1},P_t)}$ and $\Delta DV = \frac{DV_t}{DV_{t-1}}$. Then, we define our four states in a manner similar to the approach described by Li et al. (2022b):

- S_1: Diversity and convergence are improved, $\Delta CV < 1$ and $\Delta DV > 1$.
- S_2: Convergence is improved but not diversity, $\Delta CV < 1$ and $\Delta DV \leq 1$.
- S_3: Diversity is improved but not convergence, $\Delta CV \geq 1$ and $\Delta DV > 1$.
- S_4: Convergence and diversity are not improved, $\Delta CV \geq 1$ and $\Delta DV \leq 1$.

Rewards: The reward function prioritizes the enhancement of solution diversity. When improvements in both diversity and convergence occur, the reward is maximized. If only diversity shows improvement, the reward is slightly lower, and a similar reduction applies when only convergence improves. Ultimately, the lowest reward is assigned when neither diversity nor convergence witness improvement. The reward function is defined as follows:

$$Reward = \begin{cases} 10 & \text{if } \Delta CV < 1 \text{ and } \Delta DV > 1 \\ 5 & \text{if } \Delta CV \geq 1 \text{ and } \Delta DV > 1 \\ 0 & \text{if } \Delta CV < 1 \text{ and } \Delta DV \leq 1 \\ -5 & \text{if } \Delta CV \geq 1 \text{ and } \Delta DV \leq 1 \end{cases} \tag{28}$$

During each iteration, we are situated in a state denoted as s_t. We make a decision by taking an action labeled as a_t, and as a result, we receive a reward, indicated as r_t. Subsequently, the Q-value is updated according to the following formula:

$$Q(s_t, a_t) = (1 - \alpha) \times Q(s_t, a_t) + \alpha(r_t + \lambda \times max_a Q(s_{t+1}, a)), \tag{29}$$

where α represents the learning rate and λ represents the discount factor. The learning rate α typically ranges between 0 and 1. A value of α close to 1 can render the Q-value highly volatile, making it extremely responsive to fluctuations. Conversely, a value of α approaching 0 tends to favor historical information, resulting in a conservative Q-value. To strike a balance between adaptability and stability, we introduce a dynamic α coefficient that changes with each iteration. This coefficient is defined as $\alpha(t) = 1 - 0.9\left(\frac{t}{T}\right)$, where T signifies the total number of iterations. Consequently, in the early iterations, the Q-value is heavily influenced, while as the process progresses, the Q-value becomes increasingly stable.
The discount factor λ plays a critical role in temporal reward considerations. A value of λ near 0 primarily emphasizes immediate rewards, whereas a value closer to 1 places greater significance on rewards in the

distant future. In this context, we opt to establish λ at a value of 0.8.

The Q-learning algorithm is represented by the flowchart in Figure 5. The algorithm initiates with all Q-values set to zero. In each iteration, a crucial decision arises between exploration and exploitation. During exploration, the algorithm randomly selects an action and observes the associated outcome, thereby enhancing its learning process. In contrast, exploitation entails evaluating the current system state and opting for the action that maximizes the reward function. In the early stages of the algorithm, exploration takes precedence to facilitate learning, while as the iterations progress, the system tends more towards exploiting its acquired knowledge. To facilitate this transition, we employ the following epsilon-greedy exploration strategy: i) Set ϵ to 1, ii) At each iteration, choose a random number between 0 and 1, iii) If the generated number is less than ϵ, then choose a random action (exploration), otherwise take the best action Q-value (exploitation), and iv) Update epsilon: $\epsilon \leftarrow \epsilon \times 0.99$. Under this strategy, the value of ϵ decreases gradually as the process is running. Consequently, in the early stages, exploration takes precedence, while as the algorithm progresses, it focuses more and more on exploiting the acquired knowledge. It's important to note that these steps are integral components of the overall algorithm framework.

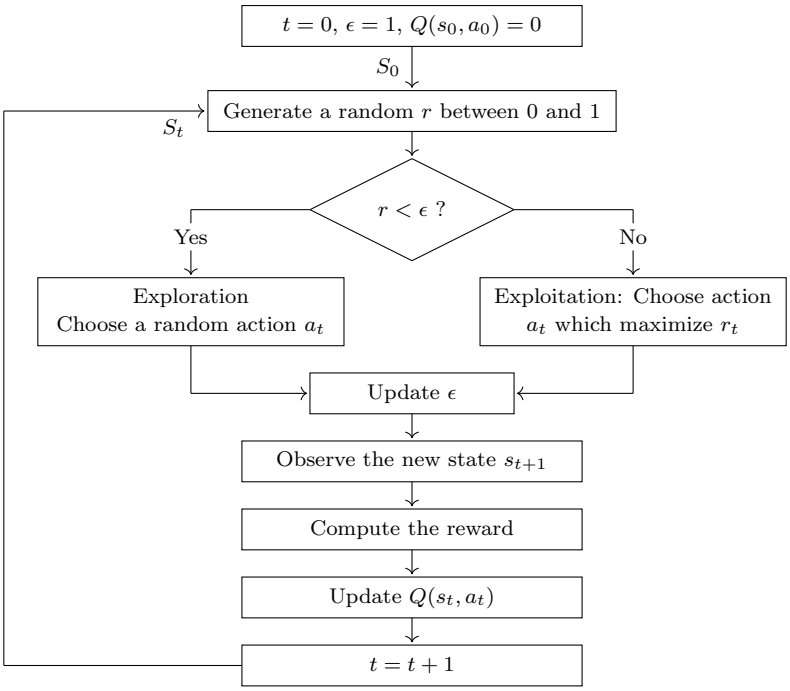


Figure 5: Q-learning framework

4.4. Fitness calculation

Each solution within the population is evaluated through a fitness function comprising three components, aligned with the three distinct objective functions: the makespan, the total energy consumption, and either the maximum OCRA index (in the first model MOP1) or the minimum preference level (in the second model MOP2).

To calculate these components in MOP1, we employ a decoding process for the three genes associated with each solution. Initially, we consider the first operation from the OS layer of the chromosome. Utilizing information from the MA and WA layers, we can determine the machine and worker to which the considered operation is assigned, allowing us to deduce its start and end dates. We then proceed to the next operation on the OS layer, where the assigned machine and worker are known. To establish the start date for this operation, we consider the maximum date among the following cases: i) The end date of the preceding task of the same job, taking into account the transport time between the respective machines; ii) The end date of

the preceding task assigned to the machine, incorporating the setup time between the two operations; and iii) The end date of the previous task assigned to the worker, accounting for the time required to transition from the prior machine to the current one, along with the worker's rest time. This approach ensures that all precedence constraints are meticulously respected. In the case where the end time of the operation exceeds the end time of the period (t), we shift the operation to the beginning of period $t+2$ to accommodate the selected operator's availability. This process is repeated until all operations in the OS layer are scheduled, enabling us to calculate the makespan, the total energy consumption (inclusive of transport, auxiliary, and task execution), and the maximum OCRA index among all operators across all periods.

For MOP2, the procedure is analogous, but with the inclusion of an additional information that specifies the shift in which workers and tasks are scheduled. This introduces an extra set of constraints impacting the start and end times of the task.

4.5. Solutions selection

4.5.1. Dominance

In the context of a multi-objective optimization problem aimed at minimization, Pareto dominance is defined as follows (Zhu et al., 2022; Chiandussi et al., 2012):

Definition 1: We define $\mathcal{O} = \{1, 2, ..O\}$ the set of objective functions (in our case, $\mathcal{O} = \{1, 2, 3\}$). A solution $\mathbf{x}$ is said to dominate a solution $\mathbf{y}$, denoted as $\mathbf{x} \prec \mathbf{y}$, if and only if:

$$\begin{cases} \forall o \in \{1, 2, .., O\} : f_o(\mathbf{x}) \leq f_o(\mathbf{y}) \\ \exists o' \in \{1, 2, .., O\} : f_o(\mathbf{x}) < f_o(\mathbf{y}) \end{cases}$$

Definition 2: The Pareto-Front is the set of solutions whose objectives cannot all be improved simultaneously. It means that we cannot improve one of the objectives without damaging another one. These solutions are non-dominated by any other solution.

At the end of each iteration t, we need to select N solutions from the set R_t to form the forthcoming population P_{t+1}. Our primary objectives are to maintain elitism and retain solutions that closely approximate the Pareto Front. To achieve this, we assess the level of dominance for each solution within R_t. Solutions that are not dominated by any other are grouped into a first front denoted by $\mathcal{F}_0$ and assigned the rank 0. Solutions dominated by only one other solution are categorized into a second front denoted by $\mathcal{F}_1$ and assigned the rank 1, and so forth. Subsequently, we traverse the fronts in order, starting from front $\mathcal{F}0$, and select solutions from these fronts until the size of P_{t+1} reaches the specified limit of N. In most cases, we reach a front, denoted as front l, where it is not possible to retain all of its solutions, as this would cause P_{t+1} to exceed the size constraint of N. In such situations, we must decide which solutions from this front to select in order to reach the desired size of N for P_{t+1}. The specific method employed for this selection varies depending on the algorithm used. For instance, NSGA-II utilizes the crowded-comparison operator, as detailed in section 4.5.2. In NSGA-III, a three-step procedure is employed, encompassing normalization, association, and niche preservation, as outlined in section 4.5.3. These techniques help to promote the exploration of new areas of research, thus guaranteeing the diversity of the solutions obtained and mitigating the risk of falling into local optima.

4.5.2. Crowded-comparison operator

The crowded-comparison operator plays a crucial role in NSGA-II, particularly in the selection of solutions that will make up the subsequent population, denoted as P_{t+1}. As described in section 4.5.1, we systematically explore the initial fronts, commencing from the front $\mathcal{F}_0$, retaining solutions from these fronts as long as the

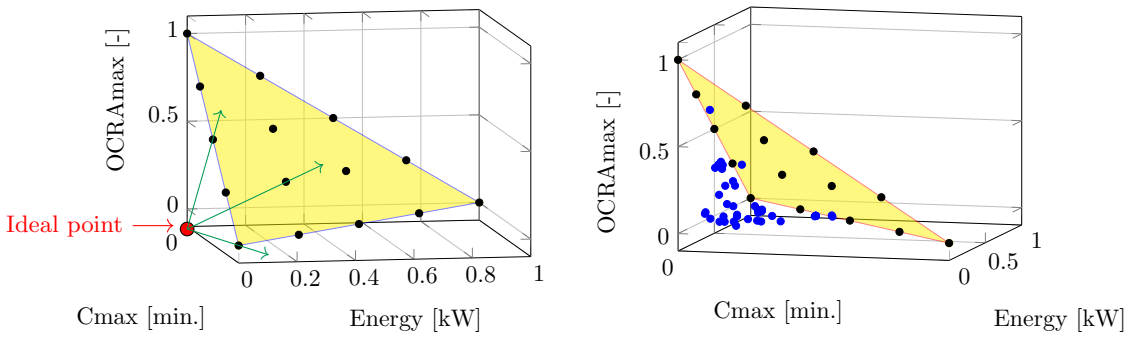
cumulative count does not surpass N. However, if the inclusion of solutions from the front $\mathcal{F}_l$ would lead to a population size exceeding N, we employ the crowded distance criterion to select only a specific number of solutions from the front $\mathcal{F}_l$ in order to complete P_{t+1} to its designated size of N.

To maximize the coverage of solutions and steer clear of local optima, we prioritize diversity within the set of explored solutions by leveraging the concept of crowded distance. This entails calculating, for each solution, its distance to all other solutions as a summation across the three objective functions. Smaller crowded distances indicate that a solution is closely surrounded by other solutions, signifying its proximity within the solution space. In contrast, solutions with larger crowded distances are more isolated from their counterparts. Priority is afforded to solutions with larger crowded distances, as they offer the potential to explore novel solution neighborhoods and expand the scope of our research area.

In summary, we employ the crowded comparison operator to compare two solutions. Initially, we examine their ranks: if they differ, we favor the solution with the lower rank (i.e. closer to the Pareto front) to uphold elitism, otherwise we opt for the solution with the largest crowded distance, thereby promoting diversity.

4.5.3. Normalization, association, and niche-preservation

NSGA-III shares a similar framework with NSGA-II, with the key distinction lying in the method of solution selection. While NSGA-II relies on the crowded-comparison operator, NSGA-III employs an alternative technique as elaborated in Deb and Jain (2014). Briefly, this method comprises three distinct steps. The initial step involves the determination of reference points. Following the approach outlined in Deb and Jain (2014), these reference points are strategically positioned on a normalized hyperplane that is evenly oriented across all objective axes and intersects each axis at a value of 1. The number of reference points (H) corresponds to the binomial coefficient computed by $H = C_p^{M+p-1}$, with p the number of divisions along each objective and M the number of objectives. In our scenario, we are dealing with three objective functions, and we opt to partition each of them into four segments, resulting in a total of 15 reference points. The placement of these reference points is illustrated in Figure 6. In Figure 6b, we provide an example of our normalized solutions, with the majority positioned below the hyperplane, as we are addressing a minimization problem. The ideal point, emphasized in Figure 6a, is defined as $(0, 0, 0)$. Subsequently, we connect this ideal point to each of the reference points using vectors, which we will refer to as the 'reference lines.' A subset of these lines is depicted in green in Figure 6a.



(a) With examples of reference lines **(b)** With examples of solutions

Figure 6: Reference points and normalized hyperplane

The second step involves the assignment of each solution to a reference line and, consequently, to a specific reference point. To achieve this, we calculate the perpendicular distance of each solution from every reference line. A solution is then paired with the reference point corresponding to the reference line with the shortest distance to that solution.

The third step involves niche preservation. In this phase, we compute the number of solutions n_p associated with each reference point p within P_{t+1}. Until the size of P_{t+1} reaches N, we execute the following procedure.

First, we identify the reference point with the minimum n_p. Then, if there are multiple reference points with the same minimal value, we randomly select one. If several solutions within $\mathcal{F}l$ are associated with this reference point, we randomly choose one and increment n_p. However, if there are no associated solutions from $\mathcal{F}l$, the reference point is excluded from the selection process.

4.6. Local search

At the end of each iteration, we perform a local search to enhance the exploitation capabilities of the algorithm, aiming to potentially improve the quality of solutions (Li et al., 2022c). This local search is based on variable neighborhood search algorithm and is applied to a subset of τ randomly selected solutions from the new population. In this study, we employed 5 different local search techniques based on our problem. However, conducting all of them on each solution would be overly time-consuming, and selecting one randomly may reduce the likelihood of success. Hence, we adopt a reinforcement learning approach, inspired by the methodology outlined in Li et al. (2022c), to strategically choose the most promising local search for each solution. The 5 employed local search techniques are as follows:

- LS_1: Randomly select two operations and exchange their position in the chromosome OS.
- LS_2: Randomly choose an operation and adjust its machine and worker allocation to the machine/worker pair that minimizes its processing time.
- LS_3: Randomly choose an operation and adjust its machine and worker allocation to the machine/worker pair that minimizes its energy consumption.
- LS_4: Randomly choose a worker and change its shift assignment.
- LS_5 for MOP1: Randomly choose an operation and assign it to the machine that minimizes the $OCRA$ index.
- LS_5 for MOP2: Randomly choose an operation and assign it to the machine preferred by the worker assigned to it.

Two memory matrices, namely Success Memory (SM) and Failure Memory (FM), are used in the algorithm as depicted in Figure 7. Initially, both SM and FM are empty and are updated at each iteration t. The probability associated with each local search is initially set to 0.2. Determining the maximum size of SM and FM ($smax$) and the number of solutions τ to explore per iteration are crucial parameters. We have opted to set them at 20% of the number of iterations and 20% of the population size, respectively. At each iteration t, the counters for success and failure solutions for each local search LS_i, respectively denoted as ns_i and nf_i, are reset to 0. τ solutions are randomly selected, and a roulette algorithm is executed to select the local search to be used for each solution based on their associated probabilities. Subsequently, a new solution is generated using the chosen local search LS_i. If the new solution dominates the previous one, ns_i is incremented; otherwise, nf_i is incremented. After performing the τ local searches, the counts ns and nf are appended to the end of SM and FM, respectively. If the size of SM and FM exceeds $smax$, the oldest records are removed. Then, the probability of each local search is updated using the formula $prob_i = \frac{SR_i}{SR_i+FR_i}$, where $SR_i = \sum_{j=1}^{smax} ns_{ij}$ and $FR_i = \sum_{j=1}^{smax} nf_{ij}$. Finally, each probability is normalized by dividing by the sum of all probabilities: $PR_i = \frac{prob_i}{\sum_{k=1}^{5} prob_k}$.

4.7. NSGA-II and NSGA-III

In this subsection, we provide a brief overview of the fundamental framework of the NSGA-II and NSGA-III algorithms that we have implemented. The distinct steps involved in both algorithms are outlined in Figure 8, and the improvements made in this work are highlighted in bold. The primary divergence between these two algorithms pertains to their respective solution selection mechanisms. In each iteration, solutions are chosen from the l top-ranked fronts to form the population P_{t+1}. To fill the remaining slots and reach a total of N individuals, NSGA-II employs the crowded comparison operator, whereas NSGA-III employs a combination of normalization, association, and niche-preservation techniques.

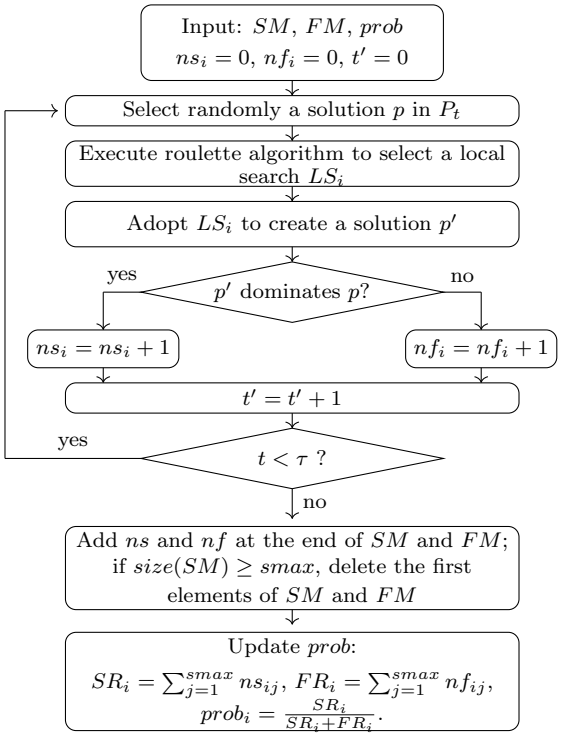


Figure 7: Variable neighborhood search based on reinforcement learning

Regarding the stopping criteria, we impose a minimum number of iterations and a condition based on the ongoing improvement of solutions (iterations continue as long as significant improvements are observed). Furthermore, we implement a multi-start strategy, starting with several multiple initial solutions to mitigate the risk of falling into local optima.

5. Metrics and used instances

5.1. Metrics

Comparing two solutions in a single-objective problem is straightforward. In the case of minimization, the smallest solution is deemed superior, while for maximization, it's the largest. However, in multi-objective problems, a solution can outperform another in one objective while underperforming in another. To evaluate the performance of different algorithms in such scenarios, we rely on various metrics. In this study, we will employ four key metrics: the hypervolume (Hv), the set coverage (Sc), the diversity (Dv), and the inverted generational distance (IGD) to assess the quality and distribution of the solutions.

Hyper volume: this metric quantifies the volume enclosed by all solutions within the ensemble. It serves as one of the most effective metrics for assessing the quality of algorithms in the context of multi-objective problems. The hyper volume takes into account both the convergence of solutions toward the Pareto front and the diversity among these solutions. To calculate the hyper volume, we utilize the 'LebMeasure' algorithm, as introduced by Fleischer (2003).

Figure 9 illustrates how the hyper volume operates in a 2D scenario with two objectives. The process involves calculating the area of the rectangles, as depicted in the figure. In cases with 3 objectives, calculating the hyper volume is similar to measuring the volume of cubes. A larger hyper volume value corresponds to a better set of solutions. In the figure, the red set in subfigure (a) outperforms the blue set in subfigure (b). The orange point serves as the reference point for hyper volume calculation and represents the worst value in both objectives.

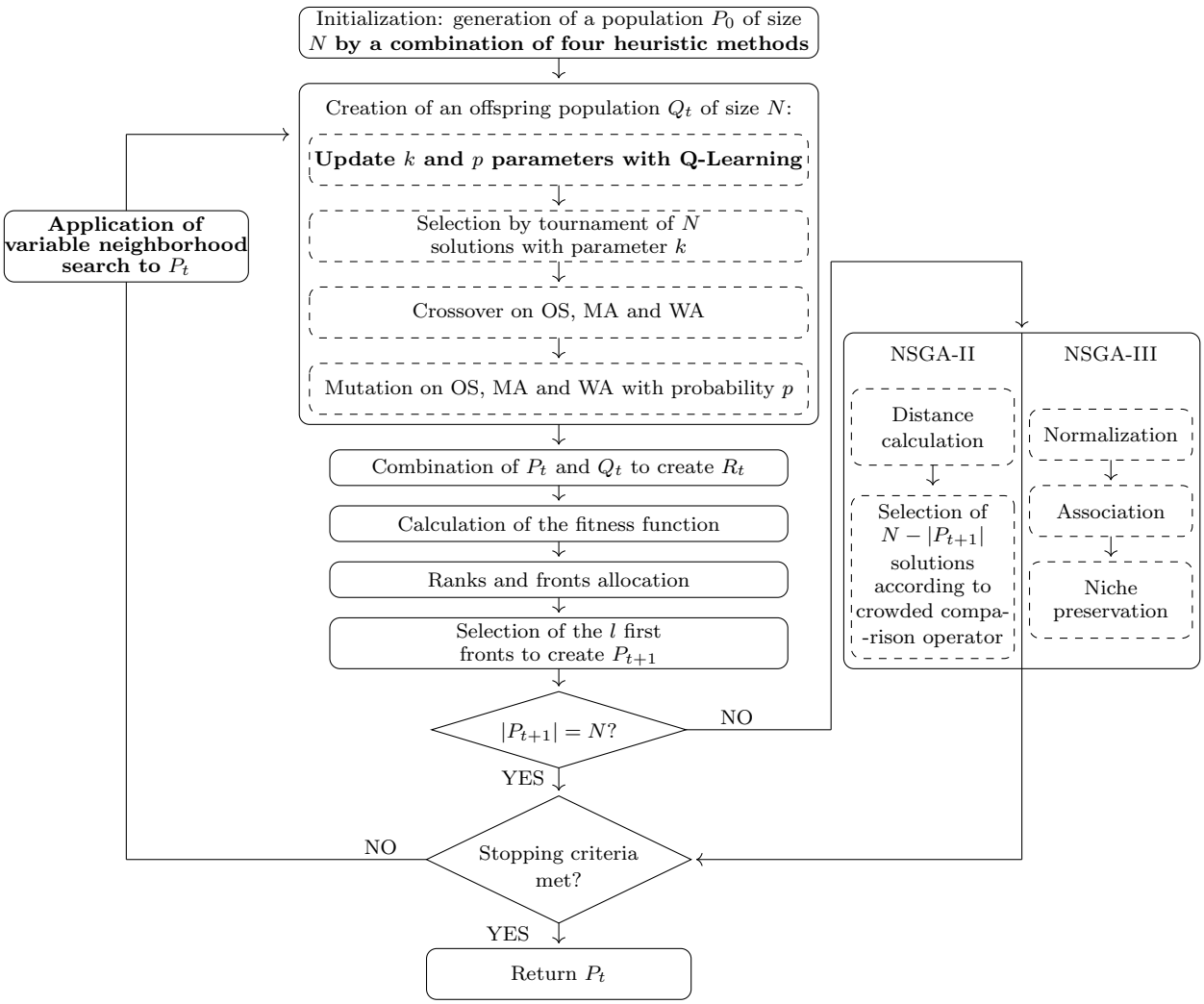
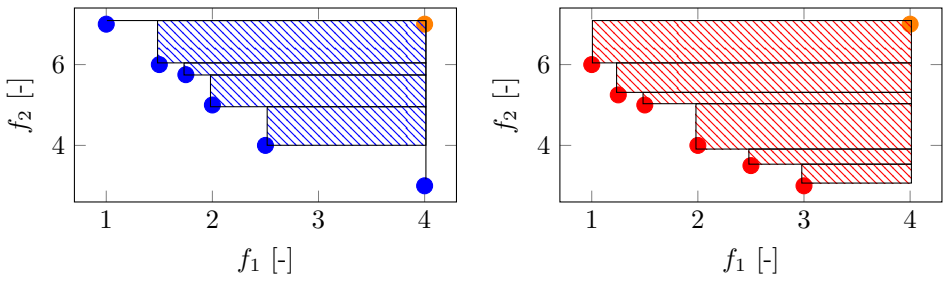


Figure 8: NSGA-II and NSGA-III description, with improvements highlighted in bold



(a) First set **(b)** Second set

Figure 9: Comparison of two solution sets using the hyper volume metric for two objective functions

Set coverage: We employ the set coverage metric, as defined by Caldeira and Gnanavelbabu (2021). Let P_1 and P_2 represent two sets of solutions. $Sc(P_1, P_2)$ quantifies the proportion of solutions in P_1 that are dominated by at least one solution in P_2. If $Sc(P_1, P_2)$ yields a higher value than $Sc(P_2, P_1)$, it implies that set P_2 is superior to set P_1. This metric is instrumental in ascertaining which set of solutions closely approximates the Pareto front. Moreover, we integrate this metric into our algorithm to evaluate the convergence (CV) of our solutions.

Diversity: We incorporate a diversity metric, as outlined in Li et al. (2022b), which is denoted by $Dv(P_1) = \frac{\sum_{i=1}^{N-1} |d_i - \bar{d}|}{(N-1)\bar{d}}$. Here, $N = |P_1|$, d_i represents the Euclidean distance between two adjacent points in P_1, and

$\bar{d}$ signifies the average value of d_i. A higher value of Dv signifies a greater diversity among the solutions. This metric serves the purpose of evaluating whether the algorithm has successfully escaped local minima and explored a diverse range of solutions.

Inverted generational distance: This metric is used to evaluate the distance of a set of solutions to the Pareto front. In most cases, the Pareto front is difficult to calculate due to the complexity of the problem. The metric is calculated as follows (Tang et al., 2024): $IGD(P, P^*) = \frac{1}{|P|} * \sum_{i=1}^{|P|} d_i$, where P is the set of solutions evaluated, P^* the set Pareto front, and d_i is the Euclidean distance of a solution i to the closest solution in the Pareto front.

5.2. Instances

To evaluate and validate our algorithms, we conducted tests using diverse instance types, inspired from the works of Kacem et al. (2002) for small instances, and Caldeira et al. (2020) for bigger instances. These two data sets encompass jobs, operations, machines, and associated processing times. We enriched them with supplementary data encompassing energy consumption, worker characteristics, and transport/setup times, as documented in existing literature. For instance, energy data was sourced from Yin et al. (2017a) and Caldeira et al. (2020), OCRA index data from Akyol and Baykasoğlu (2019), and transportation/setup time data from Zhang et al. (2019). For clarity, we denote instances influenced by Kacem et al. (2002) as kcm_i and instances influenced by Caldeira et al. (2020) as FJ_i, as outlined in Table 5a. Table 5b presents additional data used for all instances. For workers' skills, we take into account a coefficient factor at the time of task execution. A smaller value of the factor indicates a higher level of worker's skills.

To provide a practical demonstration of the two models introduced in this study, we tested them using an instance described in Yin et al. (2017b), representing an engineering case study at an engine plant operated by DF Motor Corporation. The plant operates as a partially flexible job shop, specializing in the production of four distinct product types: plate brackets, fan brackets, fan hubs, and adapter flanges. The manufacturing floor is equipped with eight machines, including two millers, two drilling machines, one machining center, one manual lathe, and two computer numerical controlled lathes. In our study, we adapt the instance to our model by adding 8 workers, along with all information concerning their activities. We consider a production planning composed of ten units of each product type, resulting in instance comprising 40 jobs, 570 operations, and 8 machines. We will refer to this instance as $DFMC$ throughout the remainder of the paper.

6. Algorithms performance

This section showcases performance study of the proposed algorithms through several comparisons. First, we compare the NSGA-II and NSGA-III performances. Then, we present the NSGA-III combined with Q-Learning (QNSGA-III) results and compare them to the NSGA-III results with fixed parameters. We also evaluate QNSGA-III versus another evolutionary algorithm, namely the Strength Pareto EvolutionaryAlgorithm 2 (SPEA2), improved with Q-learning (QSPEA2). Subsequently, we analyze the results of QNSGA-III with and without heuristic initialization and variable neighborhood search.

To ensure a robust comparison of the algorithms, we conducted multiple runs on each considered instance using identical parameters. We opted to base our analysis on 10 runs, under the assumption that the results maintain a consistent level of reliability. For each run, we generate populations consisting of 100 chromosomes, and we carry out 6 sets of 100 iterations, amounting to a total of 600 iterations.

6.1. NSGA-II vs NSGA-III performance comparison

To address our problem, we use the NSGA-II and NSGA-III. We begin by comparing the performance of these algorithms. The results for instances kcm_1, kcm_2, kcm_3, kcm_4 are summarized in Table 6. In this table, a result written in bold signifies that NSGA-III outperforms NSGA-II. Notably, NSGA-III exhibits

Inst. ID	# jobs	# tasks	# machines	# workers	proc. time (min)	energy (kW)
kcm_1	4	12	5	3	$[1-54]$	$[4-15]$
kcm_2	10	29	7	5	$[1-56]$	$[4-15]$
kcm_3	10	30	10	7	$[1-17]$	$[4-15]$
kcm_4	15	56	10	4	$[1-85]$	$[4-15]$
FJ_{05}	18	78	6	4	$[1-20]$	$[15-20]$
FJ_{10}	31	125	15	8	$[1-20]$	$[15-20]$
FJ_{15}	46	182	15	8	$[1-20]$	$[15-20]$
FJ_{20}	60	234	30	20	$[1-20]$	$[15-20]$
$DFMC$	40	570	8	8	$[1-12]$	$[0.15-1.36]$

(a) Instances

energy for transport (by time unit)	3.9
energy auxiliaries (by time unit)	1
transport time (min)	$[1-3]$
setup time (min)	$[0-1]$
travel time (min)	$[0.5-2]$
workers' skills	$[0.9-1.3]$
nb of technical actions	$[10-70]$
posture multiplier	1-07-0.6-0.5
fatigue multiplier	1-0.85-0.65-0.35-0.2-0.01
repetitiveness multiplier	1-0.85-0.7
additional factors multiplier	1-0.95-0.9-0.8
preferences machines, variety, shift	$[0-1]$

(b) Additional data

Table 5: Instances composition

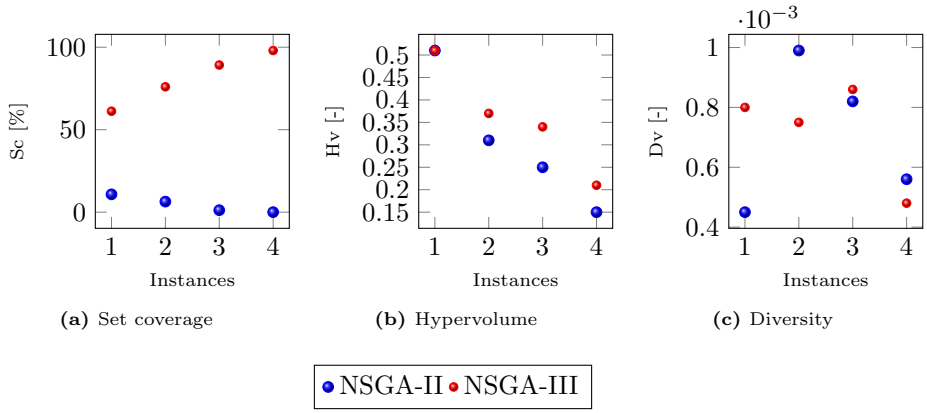
superior performance across all metrics. For a more detailed analysis of numerical values across all instances and metrics, refer to Figure 10. Our results align with the conclusion of Campos Ciro et al. (2016), which mention that NSGA-III is more adapted than NSGA-II for solving problems with more than two objective. Figure 11 compares two sets of solutions obtained for the instance kcm_1 using NSGA-II and NSGA-III, with the Pareto front derived from the ϵ-constraint approach using Gurobi solver (denoted as GUROBI). Specifically, the mean IGD for the solutions obtained with NSGA-II and NSGA-III are 0.163 and 0.147 respectively for MOP1, and 0.1732 and 0.1476 respectively for MOP2. This indicates that the solutions obtained with NSGA-III are closer to the Pareto front compared to those obtained with NSGA-II.

Inst. ID	kcm_1	kcm_2	kcm_3	kcm_4
Sc	**50%**	**69%**	**88%**	**98%**
Hv	**0%**	**14%**	**35%**	**39%**
Dv	-16%	**5%**	-32%	**44%**

Table 6: Comparison between NSGA-II and NSGA-III performance

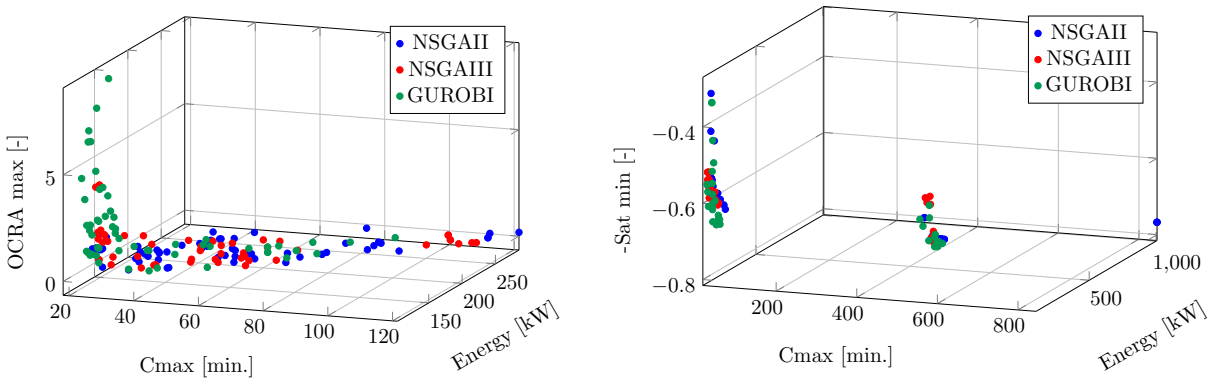
6.2. Q-learning vs Fixed parameters based comparison

In this section, we compare NSGA-III with fixed parameters (k and p) and with Q-learning (QNSGA-III), while excluding the heuristic for initialization. To simplify, we present only the MOP1 results in this section since those obtained with MOP2 led to the same trends and conclusions. We denote the populations obtained with Q-learning as P_Q and those obtained with fixed parameters as P_f. We experimented with four parameter pairs (p,k): (0.2,0.125), (0.2,0.5), (0.8,0.125), and (0.8,0.5). The results are summarized in Table 7. In this table, percentages written in bold indicates that the results obtained with Q-learning are better. The set coverage gap represents the difference between the two set coverages, calculated as $Sc(P_f, P_Q) - Sc(P_Q, P_f)$. The hyper volume gap is determined using the formula $\frac{Hv(P_Q)-Hv(P_f)}{Hb(P_f)} \times 100$. Similarly, the diversity gap



(a) Set coverage **(b)** Hypervolume **(c)** Diversity

Figure 10: Comparison between NSGA-II and NSGA-III performance



(a) MOP1 **(b)** MOP2

Figure 11: Comparison of NSGA-II, NSGA-III and GUROBI results through the kcm_1 instance for MOP1 and MOP2

is computed with the formula $\frac{Dv(P_Q)-Dv(P_f)}{Dv(P_f)} \times 100$. Note that the hyper volume and diversity percentages may appear relatively high due to the use of small numerical values. We notice the fact that the Q-learning method significantly enhances the majority of the solutions. With only a few exceptions, there is a notable enhancement in diversity for each instance.

		0,2-0,125	0,2-0,5	0,8-0,125	0,8-0,5
kcm_1	Sc	-14%	**56.8%**	**5.6%**	**65%**
	Hv	-10.1%	**8.29%**	**1.87%**	**8.29%**
	Dv	**17.28%**	**65.16%**	**46.54%**	**31.12%**
kcm_2	Sc	-42.8%	**48.4%**	-0.4%	**64.2%**
	Hv	-5.23%	**6.49%**	**1.46%**	**9.2%**
	Dv	**38.29%**	**64.91%**	**27.07%**	**25.45%**
kcm_3	Sc	**18%**	**72%**	**40%**	**90.6%**
	Hv	**1.54%**	**3.78%**	**1.08%**	**10.45%**
	Dv	**42.47%**	-6.43%	-13.76%	**74.10%**
kcm_4	Sc	-5%	**25.8%**	**31.6%**	**83.4%**
	Hv	**11.69%**	**13.45%**	**14.91%**	**23.68%**
	Dv	**23.98%**	**13.87%**	-13.5%	**31.21%**

Table 7: Comparison between NSGA-III with Q-learning and NSGA-III with fixed parameters

Table 8 presents the computation times for each instance and configuration. The provided values represent the normalized time (by the best time identified for each instance). A bold '1' signifies the configuration with the shortest computational time. Other values serve as multipliers of the best time, providing a relative comparison across configurations. For all instances, when using Q-learning or a fixed parameter of $p = 0.2$, both algorithms exhibit faster performance compared to a parameter value of $p = 0.8$.

	Q	0,2-0,125	0,2-0,5	0,8-0,125	0,8-0,5
kcm_1	1.08	1.05	**1**	1.19	1.11
kcm_2	1.02	1.02	**1**	1.18	1.13
kcm_3	1.08	1.08	**1**	1.18	1.28
kcm_4	1.03	1.01	**1**	1.16	1.08

Table 8: Normalized calculation times across instances relative to the optimal performance

In a recent study, Pătrăușanu et al. (2024) identified NSGA-II and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) as two of the most successful evolutionary algorithms. The SPEA2 is characterised by the parameters k and p, which can either be fixed or determined using the same Q-Learning approach applied in QNSGA-III. To evaluate the differences and perform a deeper analysis in the integration of Q-Learning, we implemented both SPEA2 with fixed parameters, as proposed by (Zitzler, E. et al., 2001), and enhanced version incorporating Q-Learning, referred to as QSPEA2. Figure 12 gives a comparison of the non-dominated solutions obtained with NSGA-III, QNSGA-III, SPEA2 and QSPEA2 through the kcm_1 instance. The mean IGD values obtained for SPEA2 and QSPEA2 are 0.2 and 0.19 respectively, indicating that the solutions obtained with Q-Learning are closer to the Pareto front.

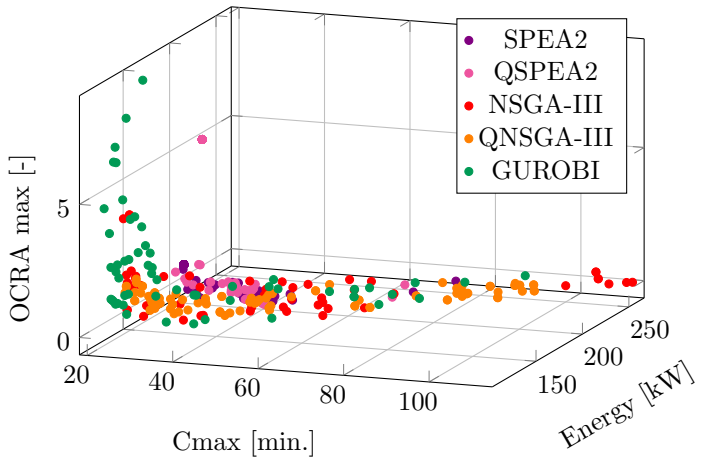


Figure 12: Comparison of SPEA2, QSPEA2, NSGAIII, QNSGAIII and GUROBI using the kcm_1 instance for MOP1

Table 9 shows the gap between the hypervolume of the solutions obtained with GUROBI and those obtained with SPEA2, QSPEA2, NSGA-III, and QNSGA-III. It is noted that NSGA-III outperforms SPEA2, and the Q-Learning approach yields better results than fixed parameters in both algorithms.

Algorithms	SPEA2	QSPEA2	NSGA-III	QNSGA-III
Hv gap	4.35%	4.24%	2.39%	2.18%

Table 9: Hypervolume gaps of SPEA2, QSPEA2, NSGA-III, QNSGA-III in comparison to GUROBI using the kcm_1 instance for MOP1

For a deeper analysis, we tested the original NSGA-III, as proposed and implemented by Blank and Deb (2020) in the Pymoo library for Python. We compared our QNSGA-III with the original NSGA-III using the DAS-CMOP7, DAS-CMOP8, and DAS-CMOP9 instances from Fan et al. (2020), each featuring 12 different levels of difficulty. Specifically, difficulty levels 1, 5, and 9 are diversity-hard problems; levels 2, 6, and 10 are feasibility-hard problems; levels 3, 7, and 11 are convergence-hard problems; and levels 4, 8, and 12 exhibit all three types of difficulties. The comparison results are presented in Table 10, using the Inverse Generational Distance (IGD) and the Hypervolume as metrics. Results highlighted in bold indicate that our QNSGA-III outperformed the original NSGA-III. Notably, the QNSGA-III demonstrated better performance on diversity-hard problems and on problems that exhibit all three types of difficulties. This outcome makes

sens, as the reward mechanism in Q-Learning prioritizes diversity over other metrics. For feasibility-hard and convergence-hard problems, the original NSGA-III achieved better results, suggesting areas for future improvement of our algorithm. However, it is important to note that our QNSGA-III was designed for a specific problem, and its convergence on instances of this problem may be higher than on more general problems.

	Diversity hardness			Feasibility hardness			Convergence hardness			The 3 difficulties		
		IGD	Hv		IGD	Hv		IGD	Hv		IGD	Hv
DAS-CMOP7	1	**66%**	**37%**	2	-21%	-2%	3	**51%**	**25%**	4	**14%**	**6%**
	5	**58%**	**25%**	6	-24%	-2%	7	-40%	-15%	8	**29%**	**3%**
	9	**69%**	**37%**	10	-29%	-2%	11	-111%	**52%**	12	-458%	-29%
DAS-CMOP8	1	**62%**	**62%**	2	-15%	-5%	3	**5%**	**11%**	4	-9%	**2%**
	5	**62%**	**53%**	6	-14%	-3%	7	-3%	**20%**	8	**9%**	**7%**
	9	**54%**	**39%**	10	-21%	-5%	11	-142%	-341%	12	**1%**	**6%**
DAS-CMOP9	1	**4%**	-7%	2	**29%**	**24%**	3	**1%**	-7%	4	**49%**	**23%**
	5	**8%**	-5%	6	-14%	**2%**	7	-41%	-6%	8	**42%**	**14%**
	9	-9%	-3%	10	**73%**	**62%**	11	-20%	-95%	12	**84%**	**81%**

Table 10: Comparison between the original NSGA-III and the QNSGA-III

6.3. Random vs Heuristic initialization based comparison

The preceding results were generated utilizing random initialization without the application of local search. Subsequently, we conducted a comparative analysis with results obtained through heuristic initialization coupled with variable neighborhood search, as presented in Table 11. To establish a baseline for comparison, we reference the results generated by QNSGA-III. Entries highlighted in bold signify instances where solutions derived from heuristic initialization and local search application outperform those obtained from random initialization without local search. The analysis indicates that incorporating heuristic initialization and local search enhances both diversity and hypervolume across all instances. Nevertheless, it is noteworthy that in certain cases, set coverage is superior with random initialization. This phenomenon may be attributed to the potential for encountering local optima with our heuristic approach.

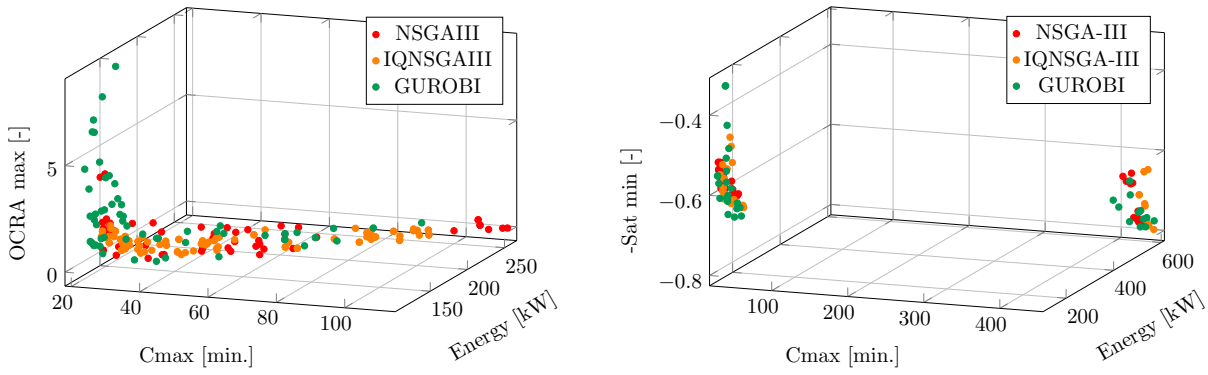
Inst. ID	kcm_1	kcm_2	kcm_3	kcm_4
Sc	-5%	**1%**	**11%**	**1%**
Hv	**32%**	**50%**	**50%**	**9%**
Dv	**71%**	**77%**	**94%**	**75%**

Table 11: Comparison between QNSGA-III and IQNSGA-III

Figure 13 illustrates the comparison between solutions obtained using NSGA-III alone and those obtained using the enhanced version of NSGA-III with Q-Learning, the heuristic method, and the local search application (IQNSGA-III). The IGD for the sets of solutions obtained by NSGA-III and IQNSGA-III are respectively 0.147 and 0.119 for MOP1, and 0.1476 and 0.0794 for MOP2, underscoring the superiority of the enhanced NSGA-III in terms of proximity to the Pareto front. The hypervolume gap between the solutions obtained with GUROBI and those obtained with IQNSGA-III is 1.31%, which outperforms the hypervolume gap of 2.18% achieved by QNSGA-III.

7. Results and sensitivity analysis

This section presents the obtained results of both models on kcm and FJ instances first, and then on the engineering case $DFMC$. Section 7.1 and 7.2 present the results and sensitivity analysis of MOP1 and MOP2 respectively, obtained using IQNSGA-III. As a reminder, MOP1 aims to minimize the makespan, the energy consumption and the OCRA index and MOP2 aims to minimize the makespan, the energy consumption and



(a) MOP1 **(b)** MOP2

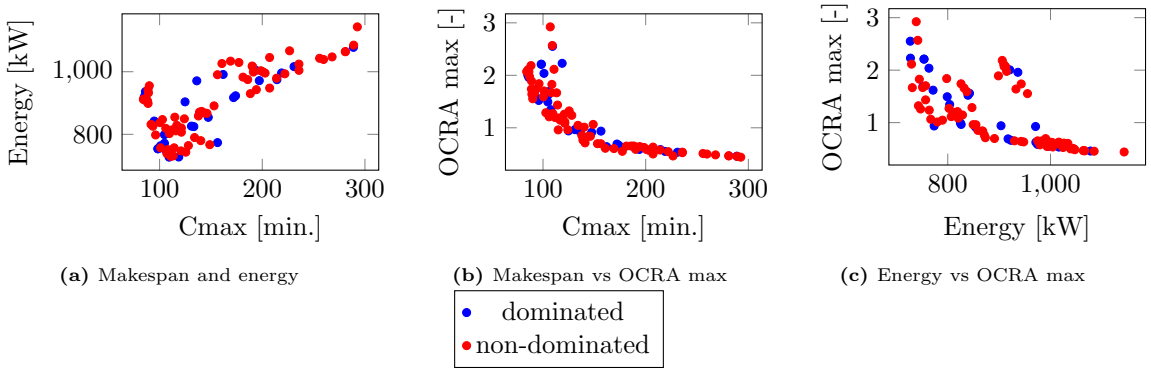
Figure 13: NSGA-III, IQNSGA-III, and GUROBI results comparison for kcm_1

the workers preferences. In 7.1 and 7.2, we respectively evaluate the sensitivity of MOP1 to the workers number variation and the sensitivity of MOP2 to the workers preferences profiles variation.

7.1. First model

7.1.1. Results analysis

At the end of the algorithm, we obtain a final population consisting of 100 chromosomes, each representing a distinct solution. Among these solutions, certain ones are non-dominated by any others. Figure 14 shows the set of solutions obtained for the instance kcm_4. The solutions depicted in red represent the non-dominated solutions that are not outperformed by any others, while the remaining solutions are shown in blue. We observe that, in this instance, there exists a positive correlation between the makespan and energy consumption, where longer processing times result in higher energy consumption. This is due to the link between the energy consumed by the auxiliary equipments and the energy consumed overall the process duration. Conversely, $OCRA^{max}$ decreases as the makespan and the energy consumption increase. This can be explained by the fact that reducing the makespan often necessitates reducing the frequency and/or duration of workers' rest periods, which systematically increase their OCRA indices.

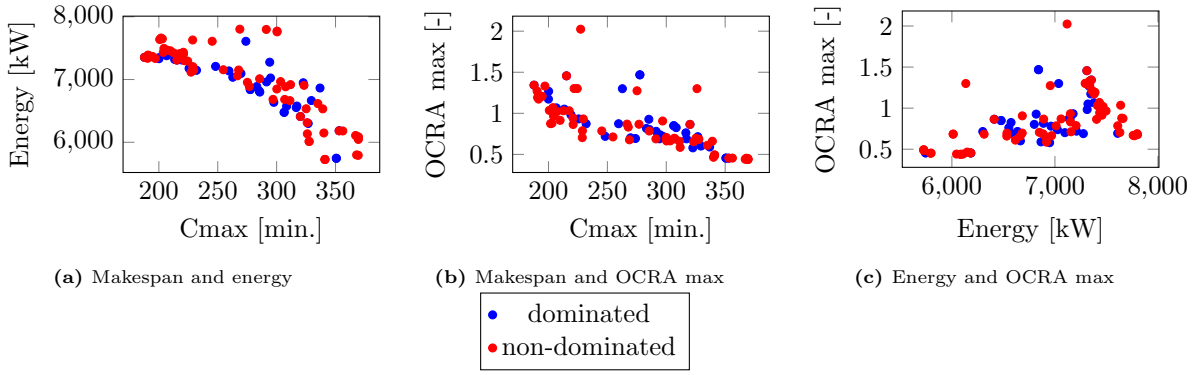


(a) Makespan and energy **(b)** Makespan vs OCRA max **(c)** Energy vs OCRA max

Figure 14: Display of the 100 solutions of kcm_4 on two-by-two axes

Divergent trends emerge when examining the curves for larger instances, e.g. FJ_{20} as showed by Figure 15. Notably, the makespan remains in contradiction with $OCRA^{max}$, whereby an increase in makespan results in a decrease in $OCRA^{max}$. However, the positive correlation between total energy consumption and makespan is no longer revealed due to the large number of tasks, extensive machine usage leading to energy consumption for transport, and heightened energy expenditure in task execution. Consequently, the auxiliary energy factor has a comparatively smaller impact on overall energy consumption. Conversely, the opposite tends to hold true, i.e. small makespan values are associated with higher energy consumption, while

larger makespan values exhibit small energy consumption. Nevertheless, energy consumption and $OCRA^{max}$ exhibit a stronger positive correlation, with both metrics having relatively low values concurrently.



(a) Makespan and energy **(b)** Makespan and OCRA max **(c)** Energy and OCRA max

Figure 15: Display of the 100 solutions of FJ_{20} on two-by-two axes

Our algorithm provides a range of solutions based on various criteria. Subsequently, the final decision involves selecting a schedule for implementation. The key consideration is determining the most suitable schedule based on the specific situations we may encounter. As an example, we will take the kcm_4 instance. Figure 16 illustrates the set of the non-dominated solutions obtained. We normalized each objective value by dividing it by the worst value found. The blue curve represents the normalized $OCRA^{max}$ values, the green curve represents the normalized energy consumption values, and the pink curve represents the normalized makespan values. Additionally, the orange curve represents the average value of these three criteria for each solution. The minimum value for each curve is denoted by a round circle. Specifically, solutions 0, 2, 10, and 33 have the minimum values of $OCRA^{max}$, makespan, average value, and energy consumption, respectively. The corresponding schedules can be found in Figure 17 in the following order: solutions 2, 33, 0, and 10.

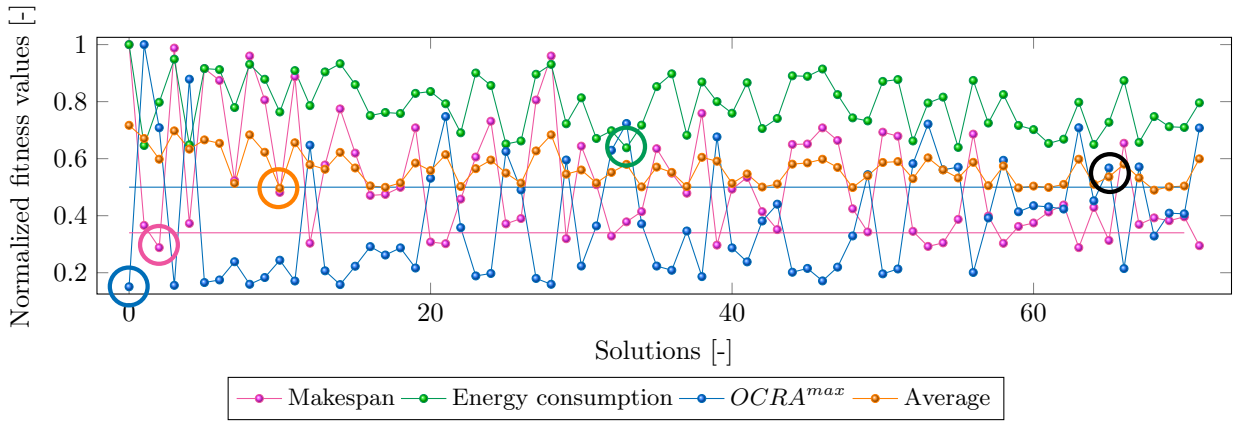


Figure 16: Normalized fitness values for the non-dominated solutions

The first schedule minimizes the makespan by efficiently assigning tasks to the most efficient workers and machines to execute them. The energy consumption remains relatively low due to the correlation between the makespan and auxiliary equipments energy consumption, while the $OCRA$ values remain below 2.2 thanks to the short duration of the process. The second schedule focuses on minimizing total energy consumption by selecting machines with low energy usage for each task. Additionally, efforts are made to keep tasks of the same job on the same machine to minimize energy expenditure on internal transport. Although the makespan remains favorable due to the correlation between makespan and auxiliary energies, only one $OCRA$ value slightly exceeds 2.2 (for worker 2) while remaining close to targeted threshold. The third schedule achieves very low $OCRA$ values, but is the longest in duration and consumes the highest amount of energy. It prioritizes worker safety, allocating tasks to machines and workers that offer maximum security, regardless of

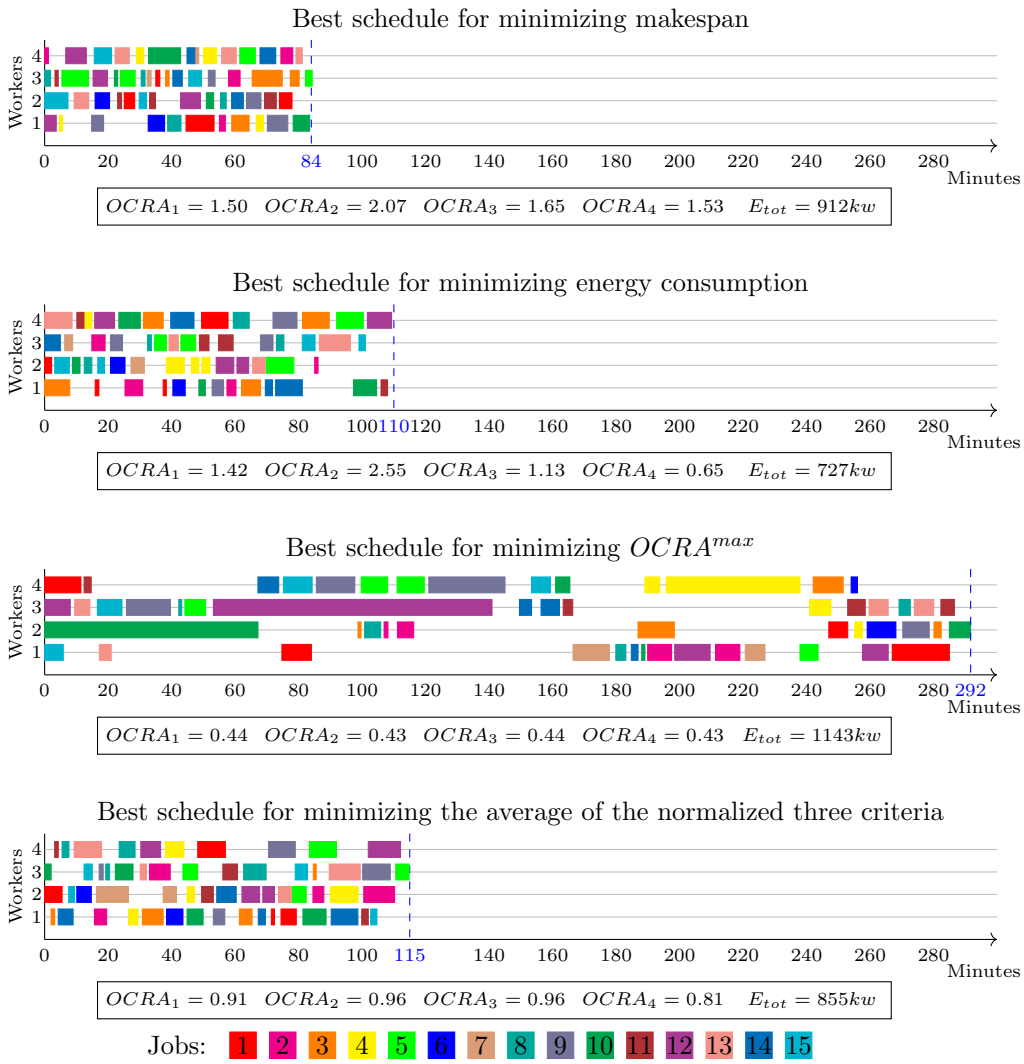


Figure 17: Comparison of different schedules

the time required. However, this schedule is not practical for industrial purposes as the makespan is 3.5 times longer than the optimal duration. To strike a balance among these three schedules, we select the schedule with the minimal average normalized fitness values. This fourth schedule, depicted in the figure, does not achieve optimal values for any objective but comes close to them. It represents a good compromise between economic, environmental, and human factors.

Then, if we are in a normal situation, the fourth schedule offers a balanced approach that considers economic efficiency, environmental impact, and worker well-being. Nevertheless, when confronted with a critical aspect, there is flexibility to temporarily adjust the schedule. For instance, situations may arise where workers frequently experience stress, fatigue, or injuries such as musculoskeletal disorders, and then the schedule can be adapted to favor workers' well-being among others aspects. Additionally, circumstances such as a peak in pollution or economic challenges might prompt us to adapt the schedule, opting for a less polluted schedule or a more concise timeframe. To emphasize a particular aspect without compromising others significantly, we can establish an upper bound for that specific aspect and seek the optimal schedule that minimizes the other two criteria. For instance, if our emphasis is on human factors, we might set $OCRA^{max}$ under 1.5 (the upper level of the optimal range, as shown in Figure 1). This corresponds, in Figure 16, to solutions where the normalized $OCRA^{max}$ is under 0.5 (indicated by the straight blue line). We then examine the average solution within this range and select the one with the minimum values, while adhering to the $OCRA^{max}$ constraint; in this case, it aligns with the fourth schedule (solution 10 in the Figure

16). Similarly, for a focus on economic considerations with a set makespan under 100 minutes, corresponding solutions would have a normalized makespan under 0.34 (indicated by the straight pink line in Figure 16). Applying the same methodology, we identify the schedule that minimizes the average of all criteria while respecting the makespan constraints. This solution is marked with a black circle on Figure 16, corresponding to the solution 65. The same reasoning applies to environmental factors, not presented here.

We have demonstrated that our set of solutions provides a broad range of scheduling possibilities. Depending on the specific situation, we can formulate a flexible schedule that simultaneously addresses all three aspects or gives priority to a particular one, while ensuring consideration for the others. The methodology outlined below serves as a valuable aid in the scheduling problem decision-making process. To test our model in a scenario that closely mirrors real-world conditions, we use the $DFMC$ instance. Results obtained are depicted in Figures 18 and 19. The makespan and energy consumption exhibit a positive correlation, with the exception of solutions featuring the smallest values of makespan and energy consumption. However, we observe a consistent increase in energy consumption alongside makespan. Conversely, the value of OCRAmax demonstrates an inverse correlation with both energy consumption and makespan. Once again, achieving a satisfactory trade-off between the three criteria can be facilitated through the normalization of the objectives, as illustrated in Figure 16. The minimum makespan value found is 1165, the minimum energy consumption is recorded as 2172, and the minimum OCRAmax index is 0.91. The solutions derived from the normalization process yield a makespan of 1201, energy consumption of 2350, and OCRA index of 1.26. It's noteworthy that this solution is very closely aligned in terms of makespan and energy consumption. A good balance is achieved between the three criteria.

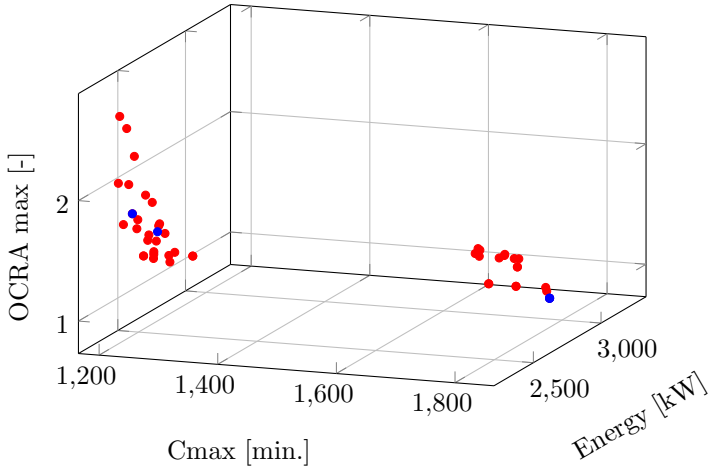
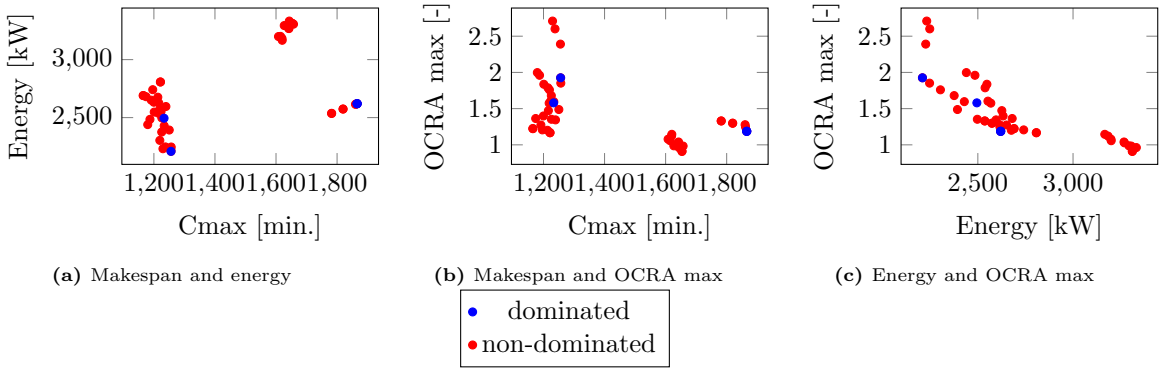


Figure 18: Display of the 100 solutions of $DFMC$



(a) Makespan and energy **(b)** Makespan and OCRA max **(c)** Energy and OCRA max

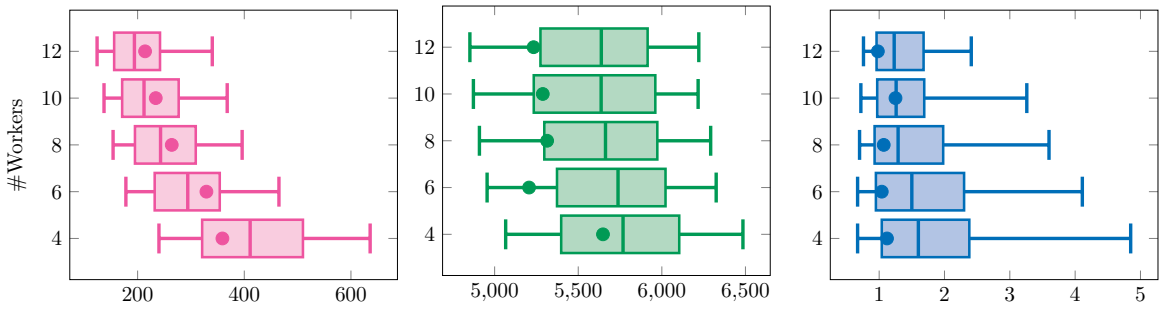
Figure 19: Display of the 100 solutions of $DFMC$ on two by two axes

7.1.2. Sensitivity analysis

We examine the influence of human-related parameters on the three specified criteria. Analyses were conducted for varying numbers of workers involved in the process. This involved the generation of 10 instances for each of the following scenarios: 25% additional workers, 50% additional workers, 25% fewer workers, and 50% fewer workers. Subsequently, we conducted some statistical analysis on the results obtained from these instances. The outcomes of the various instances tests are succinctly presented in Figure 20 for FJ_{15} and in Figure 21 for $DFMC$. In both figures, the box plots represent the dispersion of the results found for the three criteria. Then, the median, the best value, the worst value, and the quartiles are highlighted. The point indicates the value of the corresponding criterion with the minimization of the average of the three normalized criteria (refer to Figure 16).

The findings based on the results of the FJ_{15} instance show that increasing the workforce has a notable impact on all three criteria, particularly on makespan and $OCRA^{max}$. Analysis of the results reveals that a 25% and 50% increase in the number of workers correlates with a respective reduction of median makespan values by 12.5% and 20%. Conversely, a reduction in the workforce by 25% and 50% results in an increase of 21% and 69% in the median makespan values. Regarding $OCRA^{max}$, notable fluctuations in values are observed with the reduction and addition of workers. Specifically, the worst values exhibit an increase of 14% and 35% when the workforce is reduced by 25% and 50%, respectively. Conversely, when workers are added, there is a corresponding decrease. Similarly, the median values of $OCRA^{max}$ demonstrate variations ranging from 2% to 24% when adjusting the number of workers upward or downward. The energy consumption is less sensitive to the variations of the number of workers, as the best average value vary from 1% to 6%, and the median value from $0,5\%$ to $1,5\%$.

Regarding $DFMC$ instance, a notable difference is observed in the three criteria when the number of workers is adjusted. As the number of workers decreases, both the makespan and energy consumption increase across all statistical measures: median, lower bound, upper bound, lower quartile, and upper quartile. This trend is also evident in the normalized values of the three criteria. Specifically, for $OCRA^{max}$, the lowest value is found in the scenario with 12 workers, while the highest value is observed in the scenario with 4 workers.



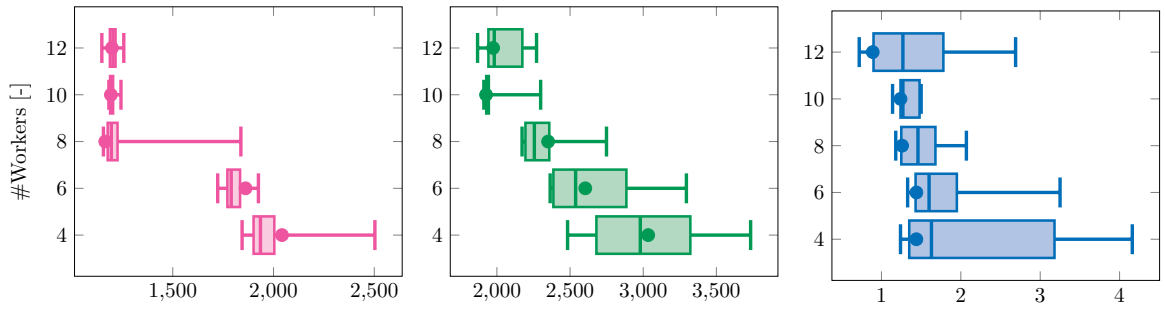
(a) Makespan **(b)** Energy consumption **(c)** $OCRA^{max}$

Figure 20: Box plot analysis: impact of workforce variations on FJ_{15}

7.2. Second model results

7.2.1. Results analysis

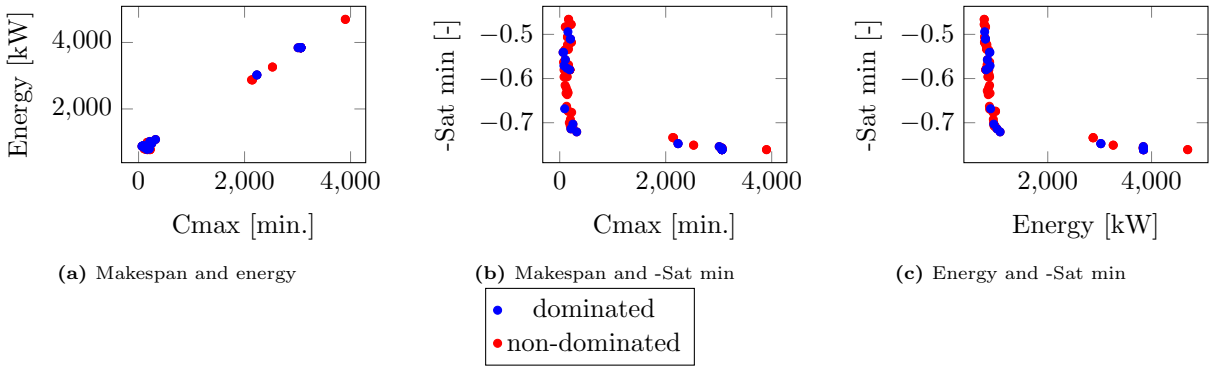
For the MOP2 model, we introduce the notion of worker preferences and define profiles for these workers based on their preferences. Figure 22 illustrates the results obtained with a short instance (kcm_4) using random generated preferences. First, we can observe notably elevated values for both makespan and energy consumption. This is attributed to the preferences among workers for different shifts. Although it is feasible to complete all operations within a single shift in our instances, the consideration of individual worker preferences necessitates the assignment of several workers to different shifts. Consequently, operations assigned to these



(a) Makespan **(b)** Energy consumption **(c)** $OCRA^{max}$

Figure 21: Box plot analysis: impact of workforce variations

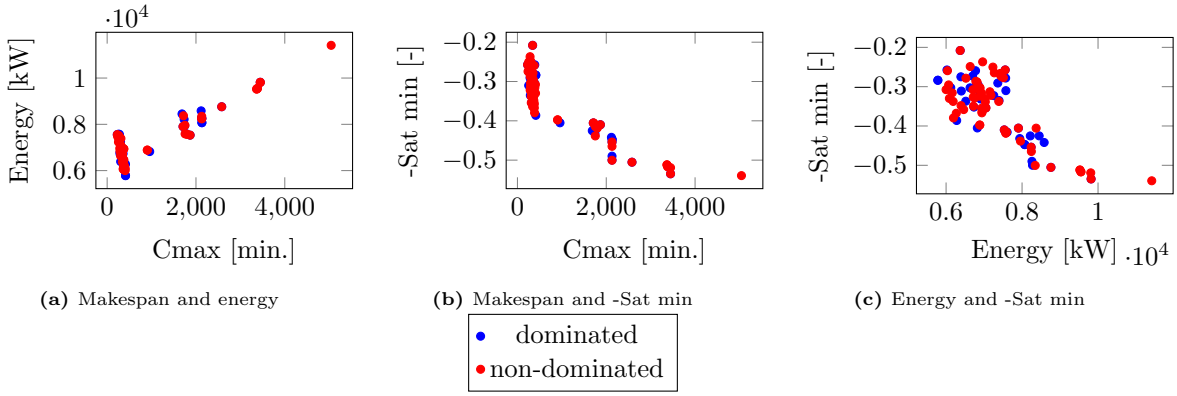
workers must wait until their designated shift to commence. This problem should not arise with larger instances. Besides, we note a discernible positive correlation between makespan and energy consumption for kcm_4, with both criteria conflicting with the minimum value of satisfaction, denoted as $-Sat^{min}$.



(a) Makespan and energy **(b)** Makespan and -Sat min **(c)** Energy and -Sat min

Figure 22: Display of the 100 solutions of kcm_4 on two-by-two axes

In the case of the FJ instances, the minimum satisfaction conflicts with both energy consumption and makespan, as shown in Figure 23. Regarding makespan and energy consumption, an increase in makespan values correlates with a corresponding rise in energy consumption when the makespan attains really high values. Conversely, when makespan values fall below 1000, a contrasting trend emerges. For a visual representation of this significant correlation, a zoom on the area of interest is given in Figure 24. We analyze similar trends to those observed in the first model. Specifically, the three criteria are found to be conflicting for the FJ instances, while for kcm_4, there exists a positive correlation between makespan and energy consumption. This correlation can be attributed to the relatively small number of tasks, which places greater emphasis on auxiliary energy considerations.



(a) Makespan and energy **(b)** Makespan and -Sat min **(c)** Energy and -Sat min

Figure 23: Display of the 100 solutions of FJ_{20} on two-by-two axes

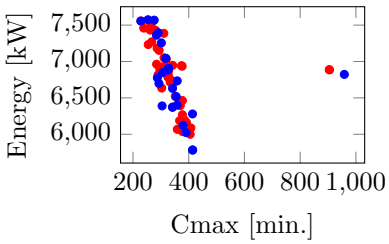


Figure 24: Zoom on Figure 23a

Figures 25 and 26 showcase the outcomes of $DFMC$ instances. The makespan and energy consumption exhibit a positive correlation, while the satisfaction minimum demonstrate an inverse correlation with both criteria. The minimum makespan, energy consumption, and maximum satisfaction found are 1237, 1947 and 0.58, respectively. The solutions derived from the normalization process yield a makespan of 1253, an energy consumption of 2024, and a maximum satisfaction of 0.53. Once again, the normalization of the objectives can achieve a good trade-off between the three criteria.

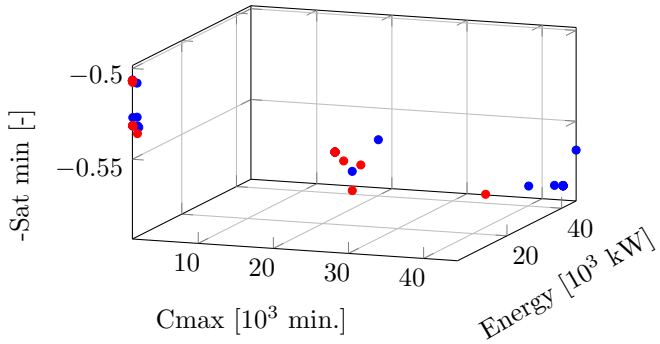
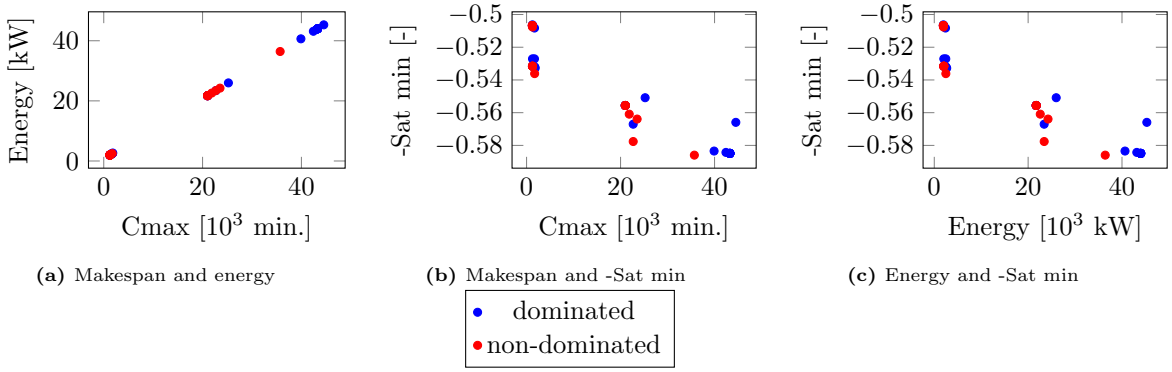


Figure 25: Display of the MOP2 solutions



(a) Makespan and energy

(b) Makespan and -Sat min

(c) Energy and -Sat min

Figure 26: Display of MOP2 solutions on two by two axes

7.2.2. Sensitivity analysis

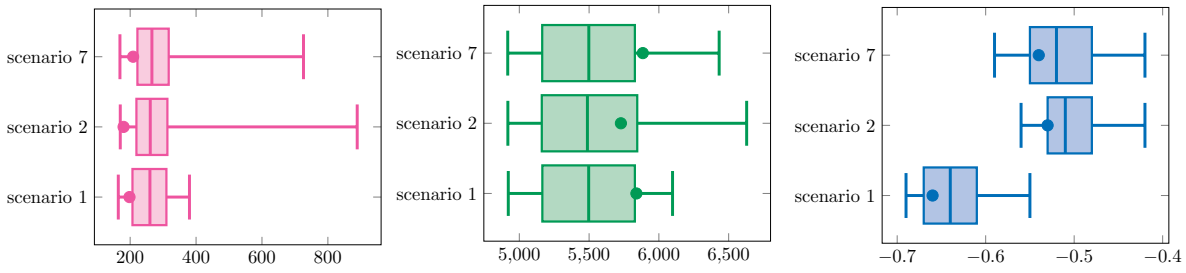
Next, our objective is to examine the fluctuations in the three criteria based on different workers' profiles. We defined four distinct profiles:

- profile1: exhibits a preference for morning shifts and a diverse range of machine assignments;
- profile2: exhibits a preference for morning shifts but without variations in the types of machines used;
- profile3: exhibits a preference for afternoon shifts and a diverse range of machine assignments;
- profile4: exhibits a preference for afternoon shifts but without variations in the types of machines used;

Then, we have different scenarios, based on these profiles:

- scenario 1: the workforce comprises individuals exclusively aligned with profile 1;
- scenario 2: the workforce comprises individuals exclusively aligned with profile 2;
- scenario 3: the workforce comprises individuals exclusively aligned with profile 3;
- scenario 4: the workforce comprises individuals exclusively aligned with profile 4;
- scenario 5: the workforce comprises 50% of individuals aligned with profile 1 and 50% of individuals aligned with profile 3 ;
- scenario 6: the workforce comprises 50% of individuals aligned with profile 2 and 50% of individuals aligned with profile 4;
- scenario 7: the workforce comprises 50% of individuals aligned with profile 1 and 50% of individuals aligned with profile 2;
- scenario 8: the workforce comprises 50% of individuals aligned with profile 3 and 50% of individuals aligned with profile 4;
- scenario 9: the workforce comprises 25% of individuals aligned with each profile.

Given the small size of our instances and the ability to execute all tasks within a single shift, it is not meaningful to analyze results in the context of scenario 5, 6, and 9 for FJ_{15}. This is because such analyses would yield high makespan values, which, in reality, may not be reflective of practical scenarios. Then, we choose to compare results for scenario 1, 2, and 7, analyzing the impact of the preferences for variability or regularity on each metric. A series of 10 tests for each scenario have been conducted on distinct instances. The findings are visually represented in Figure 27. Figure 27a, 27b, and 27c specifically illustrate the distribution and trends associated with makespan, energy consumption, and preferences, providing a comprehensive overview of the model's performance across these key metrics. Firstly, in terms of minimum satisfaction, it is evident that a preference for variability ensures a better minimum satisfaction. This indicates that workers are more likely to change tasks, which is conducive to achieving favorable results in the other two criteria. Specifically, the median value of scenario 1 is 20% and 18% better than the median values of scenario 2 and scenario 7, respectively. Regarding makespan and energy consumption, the median values are not significantly affected by worker preferences. However, the worst values show a substantial increase with scenario 2 (133% for makespan and 8% for energy consumption). Across all three criteria, scenario 1 consistently yields the best results, scenario 2 the worst, and scenario 7 falls in between. Therefore, it can be concluded that workers' flexibility contributes to achieving favorable makespan and energy consumption values. Consequently, having workers who favor variability over regularity is preferable.

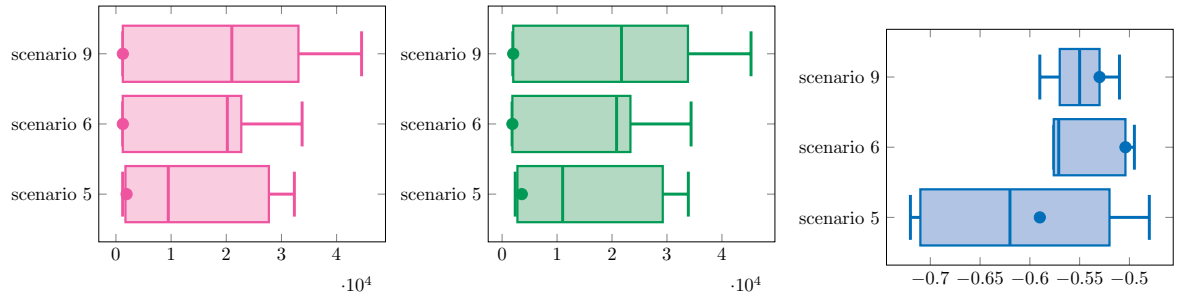


(a) Makespan **(b)** Energy consumption **(c)** $-Sat^{min}$

Figure 27: Box plot analysis: solutions variation in FJ_{15} based on different workers' profiles

For the $DFMC$ instance, we utilize scenario 5 to assess preferences for diversity, scenario 6 to examine preferences for regularity, and scenario 9 to evaluate the impact of a homogeneous distribution of workers across the four distinct profiles. The results of these analyses are presented in Figure 28. Compared to scenario 5, the median makespan increases by 112% in scenario 6 and by 121% in scenario 9. Similarly, the median energy consumption increases by 85% in scenario 6 and by 97% in scenario 9. The fluctuations in

the lower bound and the best mean of the normalized criteria are negligible. However, for the satisfaction criterion, scenario 5 shows significant improvements in the lower bound, median, and best mean of the normalized criteria compared to the other two scenarios.



(a) Makespan **(b)** Energy consumption **(c)** $-Sat^{min}$

Figure 28: Box plot analysis: solutions variation based on different workers' profiles

8. Conclusion

In this paper, we addressed a sustainable flexible job shop scheduling problem considering energy consumption and worker's safety and preferences. We proposed two multi-objective Mixed Integer Non-Linear Programs (MINLP) to model this problem. Our ultimate goal is to achieve a balance between economic, environmental and social aspects. The social criterion was addressed differently in the two models. While the first model minimizes the $OCRA^{max}$ index which assess ergonomic risks related to task execution and subsequently addresses the risk of musculoskeletal disorders, the second model maximizes Sat^{min}, which represents the minimum satisfaction level among all workers and integrates their preferences regarding working shift, machines and tasks variety. Both models are solved using Non-dominated Sorting Genetic Algorithms, NSGA-II and NSGA-III, enhanced particularly through a Q-learning method for genetic operators' parameters tuning and a local search based on reinforcement learning. We tested our models on data sets inspired from the literature and analyze their sensitivity regarding the workers number and preferences variations.

Our findings underscore the interconnected impact of each criterion across diverse instance sets. The global conclusion is that environmental and social considerations are integral to the scheduling process and must not be overlooked. Integrating worker safety and preferences introduces a conflict with the other two criteria but enhances the system resilience, mitigating risks such as absenteeism, injuries, and fatigue. Consequently, we propose a method for selecting an optimal final schedule tailored to specific contextual requirements, demonstrating favorable outcomes across all three criteria. These models serve as a foundation for decision-makers to tackle upcoming challenges associated with the evolving social landscape and the revolutions in advanced manufacturing and service systems.

We acknowledge some limitations of this study. Firstly, the models are static and are not designed to handle disruptions. That is why, in our future work, we aim to develop a dynamic model capable of responding promptly and in real-time to process-related hazards, particularly those involving workers, such as absenteeism and diligence at work. A dynamic model would enhance the resilience of the system, allowing it to recover quickly and maintain optimal performance. Secondly, regarding the workers, this study is still in its early stages. Currently, the rest time for workers is set at 8% of their working time. Conducting a more detailed study on worker fatigue and optimizing break times could significantly enhance the model's effectiveness. Additionally, other ergonomic risk index should be studied and compared, depending on the types of manufacturing processes, and more worker profiles could be studied, with a deeper analysis of their behavior and preferences. Finally, our algorithm could be applied to other scheduling models, particularly those with different objective functions, such as minimizing tardiness or cost.

References

Akbari, M., Zandieh, M., Dorri, B., 2013. Scheduling part-time and mixed-skilled workers to maximize employee satisfaction. The International Journal of Advanced Manufacturing Technology 64, 1017–1027. doi:10.1007/s00170-012-4032-4.

Akyol, S., Baykasoğlu, A., 2019. ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors. Journal of Intelligent Manufacturing 30, 291–302. doi:10.1007/s10845-016-1246-6.

Amjad, M., Butt, S., Kousar, R., Ahmad, R., Agha, M., Faping, Z., Anjum, N., Asgher, U., 2018. Recent research trends in genetic algorithm based flexible job shop scheduling problems. Mathematical Problems in Engineering 2018, 1–32.

Berti, N., Finco, S., Battaïa, O., Delorme, X., 2021. Ageing workforce effects in Dual-Resource Constrained job-shop scheduling. International Journal of Production Economics 237, 108151. doi:10.1016/j.ijpe.2021.108151.

Bezoui, M., Olteanu, A.L., Sevaux, M., 2023. Integrating preferences within multiobjective flexible job shop scheduling. European Journal of Operational Research 305, 1079–1086. doi:10.1016/j.ejor.2022.07.002.

Blank, J., Deb, K., 2020. pymoo: Multi-objective optimization in python. IEEE Access 8, 89497–89509.

Caldeira, R.H., Gnanavelbabu, A., 2021. A Pareto based discrete Jaya algorithm for multi-objective flexible job shop scheduling problem. Expert Systems with Applications doi:10.1016/j.eswa.2021.114567.

Caldeira, R.H., Gnanavelbabu, A., Vaidyanathan, T., 2020. An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. Computers & Industrial Engineering 149. doi:10.1016/j.cie.2020.106863.

Campos Ciro, G., Dugardin, F., Yalaoui, F., Kelly, R., 2016. A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016 .

Castillo-Salazar, J.A., Landa-Silva, D., Qu, R., 2016. Workforce scheduling and routing problems: literature survey and computational study. Annals of Operations Research 239, 39–67. doi:10.1007/s10479-014-1687-2.

CEN/TC122, 2008. Nf en 1005-1+a1. Ergonomie .

Chen, J., Shi, Y., 2021. Stochastic model predictive control framework for resilient cyber-physical systems: review and perspectives. Philosophical Transactions of the Royal Society A 379.

Chen, X.l., Li, J.q., Xu, Y., 2023. Q-learning based multi-objective immune algorithm for fuzzy flexible job shop scheduling problem considering dynamic disruptions. Swarm and Evolutionary Computation doi:10.1016/j.swevo.2023.101414.

Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F., 2012. Comparison of multi-objective optimization methodologies for engineering applications. Computers & Mathematics with Applications .

Coca, G., Castrillón, O., Ruiz, S., Mateo-Sanz, J., Jiménez, L., 2019. Sustainable evaluation of environmental and occupational risks scheduling flexible job shop manufacturing systems. Journal of Cleaner Production 209, 146–168. doi:10.1016/j.jclepro.2018.10.193.

Dai, M., Tang, D., Giret, A., Salido, M.A., 2019. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer-Integrated Manufacturing 59, 143–157. doi:10.1016/j.rcim.2019.04.006.

Deb, K., Jain, H., 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Transactions on Evolutionary Computation .

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation .

Destouet, C., Tlahig, H., Bettayeb, B., Mazari, B., 2023a. Flexible job shop scheduling problem under Industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement. Journal of Manufacturing Systems 67, 155–173. doi:10.1016/j.jmsy.2023.01.004.

Destouet, C., Tlahig, H., Bettayeb, B., Mazari, B., 2023b. NSGA-II for Solving a Multi-objective, Sustainable and Flexible Job Shop Scheduling Problem. doi:10.1007/978-3-031-43670-3_38.

Ding, Z., Zhao, Z., Liu, D., Cao, Y., 2021. Multi-objective scheduling of relief logistics based on swarm intelligence algorithms and spatio-temporal traffic flow. Journal of Safety Science and Resilience 2, 222–229. doi:10.1016/j.jnlssr.2021.07.003.

Dou, J., Su, C., Zhao, X., 2020. Mixed integer programming models for concurrent configuration design and scheduling in a reconfigurable manufacturing system. Concurrent Engineering 28, 32–46. doi:10.1177/1063293X19898727.

Ebrahimi, A., Jeon, H., Lee, S., Wang, C., 2020. Minimizing total energy cost and tardiness penalty for a scheduling-layout problem in a flexible job shop system: A comparison of four metaheuristic algorithms. Computers & Industrial Engineering 141, 106295. doi:10.1016/j.cie.2020.106295.

Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., Goodman, E., 2020. Difficulty adjustable and scalable constrained multiobjective test problem toolkit. Evolutionary Computation , 339–378doi:10.1162/evco_a_00259.

Fleischer, M., 2003. The Measure of Pareto Optima Applications to Multi-objective Metaheuristics, in: Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., Deb, K. (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg.

Gong, G., Deng, Q., Gong, X., Liu, W., Ren, Q., 2018. A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. Journal of Cleaner Production 174, 560–576. doi:10.1016/j.jclepro.2017.10.188.

Grabowska, S., Saniuk, S., Gajdzik, B., 2022. Industry 5.0: improving humanization and sustainability of industry 4.0. Scientometrics , 1–28.

Hajibabaei, M., Behnamian, J., 2022. Reducing noise pollution by flexible job-shop scheduling with worker flexibility: Multi-subpopulation evolutionary algorithm. Scientia Iranica doi:10.24200/sci.2022.57813.5431.

Han, Y., Peng, H., Mei, C., Cao, L., Deng, C., Wang, H., Wu, Z., 2023. Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning. Knowledge-Based Systems doi:10.1016/j.knosys.2023.110801.

Homayouni, S.M., Fontes, D.B.M.M., 2021. Production and transport scheduling in flexible job shop manufacturing systems. Journal of Global Optimization 79, 463–502. doi:10.1007/s10898-021-00992-6.

Hongyu, L., Xiuli, W., 2021. A survival duration-guided NSGA-III for sustainable flexible job shop scheduling problem considering dual resources. IET Collaborative Intelligent Manufacturing 3, 119–130. doi:10.1049/cim2.12003.

ISO/TC159/SC3, 2007. Iso 11228-3:2006. Ergonomie .

Jaber, M., Neumann, W., 2010. Modelling worker fatigue and recovery in dual-resource constrained systems. Computers & Industrial Engineering 59. doi:10.1016/j.cie.2010.03.001.

Jiang, T., Zhu, H., Liu, L., Gong, Q., 2022. Energy-conscious flexible job shop scheduling problem considering transportation time and deterioration effect simultaneously. Sustainable Computing: Informatics and Systems 35, 100680. doi:10.1016/j.suscom.2022.100680.

Kacem, I., Hammadi, S., Borne, P., 2002. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Intelligent Forecasting, Fault Diagnosis, Scheduling, and Control .

Lei, D., Tan, X., 2016. Local search with controlled deterioration for multi-objective scheduling in dual-resource constrained flexible job shop, in: 2016 Chinese Control and Decision Conference (CCDC), pp. 4921–4926. doi:10.1109/CCDC.2016.7531874. journal Abbreviation: 2016 Chinese Control and Decision Conference (CCDC).

Li, J.q., Deng, J.w., Li, C.y., Han, Y.y., Tian, J., Zhang, B., Wang, C.g., 2020. An improved Jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times. Knowledge-Based Systems 200, 106032. doi:10.1016/j.knosys.2020.106032.

Li, M., Wang, G.G., 2022. A review of green shop scheduling problem. Information Sciences 589, 478–496. doi:10.1016/j.ins.2021.12.122.

Li, M., Wang, G.G., Yu, H., 2021. Sorting-Based Discrete Artificial Bee Colony Algorithm for Solving Fuzzy Hybrid Flow Shop Green Scheduling Problem. Mathematics 9. doi:10.3390/math9182250.

Li, R., Gong, W., Lu, C., 2022a. A reinforcement learning based rmoea/d for bi-objective fuzzy flexible job shop scheduling. Expert Systems with Applications doi:10.1016/j.eswa.2022.117380.

Li, R., Gong, W., Lu, C., 2022b. A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. Expert Systems with Applications 203, 117380. doi:10.1016/j.eswa.2022.117380.

Li, R., Gong, W., Lu, C., 2022c. A reinforcement learning based rmoea/d for bi-objective fuzzy flexible job shop scheduling. Expert Systems with Applications doi:10.1016/j.eswa.2022.117380.

Li, Y., Gu, W., Yuan, M., Tang, Y., 2022d. Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network. Robotics and Computer-Integrated Manufacturing 74, 102283. doi:10.1016/j.rcim.2021.102283.

Luo, Q., Deng, Q., Gong, G., Guo, X., Liu, X., 2022. A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm. Expert Systems with Applications 207, 117984. doi:10.1016/j.eswa.2022.117984.

Mohan, S., 2008. Scheduling part-time personnel with availability restrictions and preferences to maximize employee satisfaction. Mathematical and Computer Modelling 48, 1806–1813. doi:10.1016/j.mcm.2007.12.027.

Nahavandi, S., 2019. Industry 5.0—a human-centric solution. Sustainability doi:10.3390/su11164371.

Occhipinti, E., 1998. OCRA: a concise index for the assessment of exposure to repetitive movements of the upper limbs. Ergonomics 41, 1290–1311. doi:10.1080/001401398186315.

Parente, M., Figueira, G., Amorim, P., Marques, A., 2020. Production scheduling in the context of Industry 4.0: review and trends. International Journal of Production Research 58, 5401–5431. doi:10.1080/00207543.2020.1718794. publisher: Taylor & Francis.

Pătrăușanu, A., Florea, A., Neghină, M., Dicoiu, A., Chiș, R., 2024. A systematic review of multi-objective evolutionary algorithms optimization frameworks. Processes doi:10.3390/pr12050869.

Rinaldi, M., Fera, M., Bottani, E., Grosse, E.H., 2022. Workforce scheduling incorporating worker skills and ergonomic constraints. Computers & Industrial Engineering 168, 108107. doi:https://doi.org/10.1016/j.cie.2022.108107.

Ruiz-Torres, A., Ablanedo-Rosas, J., Mukhopadhyay, S., Paletta, G., 2019. Scheduling workers: A multi-criteria model considering their satisfaction. Computers & Industrial Engineering 128, 747–754. doi:10.1016/j.cie.2018.12.070.

Ruiz-Torres, A., Alomoto, N., Paletta, G., Pérez, E., 2015. Scheduling to maximise worker satisfaction and on-time orders. International Journal of Production Research 53, 2836–2852.

Sadeghian, O., Oshnoei, A., Nikkhah, S., Mohammadi-Ivatloo, B., 2019. Multi-objective optimisation of generation maintenance scheduling in restructured power systems based on global criterion method. IET Smart Grid 2, 203–213. doi:10.1049/iet-stg.2018.0140.

Sanogo, K., Mekhalef Benhafssa, A., Sahnoun, M., Bettayeb, B., Abderrahim, M., Bekrar, A., 2023. A multi-agent system simulation based approach for collision avoidance in integrated job-shop scheduling problem with transportation tasks. Journal of Manufacturing Systems doi:10.1016/j.jmsy.2023.03.011.

Sindhwani, R., Afridi, S., Kumar, A., Banaitis, A., Luthra, S., Singh, P., 2022. Can industry 5.0 revolutionize the wave of resilience and social value creation? a multi-criteria framework to analyze enablers. Technology in Society doi:https://doi.org/10.1016/j.techsoc.2022.101887.

Song, W., Zhang, C., Lin, W., Shao, X., 2014. Flexible Job-Shop Scheduling Problem with Maintenance Activities Considering Energy Consumption. Applied Mechanics and Materials 521, 707–713. doi:10.4028/www.scientific.net/AMM.521.707.

Stanton, N., Hedge, A., Brookhuis, K., Salas, E., Hendrick, H. (Eds.), 2004. Handbook of Human Factors and Ergonomics Methods. 0 ed., CRC Press. doi:10.1201/9780203489925.

Sun, X., Guo, S., Guo, J., Du, B., 2019. A Hybrid Multi-Objective Evolutionary Algorithm With Heuristic Adjustment Strategies and Variable Neighbor-Hood Search for Flexible Job-Shop Scheduling Problem Considering Flexible Rest Time. IEEE Access 7, 157003–157018. doi:10.1109/ACCESS.2019.2948057.

Tan, W., Yuan, X., Wang, J., Zhang, X., 2021. A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced NSGA-II: An application from casting workshop. Computers & Industrial Engineering 160, 107557. doi:10.1016/j.cie.2021.107557.

Tang, H., Xiao, Y., Zhang, W., Lei, D., Wang, J., Xu, T., 2024. A dql-nsga-iii algorithm for solving the flexible job shop dynamic scheduling problem. Expert Systems with Applications doi:10.1016/j.eswa.2023.121723.

Taris, T., Schreurs, P., 2009. Well-being and organizational performance: An organizational-level test of the happy-productive worker hypothesis. Work & Stress 23, 120–136. doi:10.1080/02678370903072555. publisher: Routledge.

Wang, H., Cheng, J., Liu, C., Zhang, Y., Hu, S., Chen, L., 2022. Multi-objective reinforcement learning framework for dynamic flexible job shop scheduling problem with uncertain events. Applied Soft Computing 131.

Wang, J., Zheng, P., Qin, W., Li, T., Zhang, J., 2018. A novel resilient scheduling paradigm integrating operation and design for manufacturing systems with uncertainties. Enterprise Information Systems doi:10.1080/17517575.2018.1526322.

Watkins, C.C.H., Dayan, P., 1992. Q-learning. Machine Learning 8, 279–292. doi:10.1007/BF00992698.

Wu, R., Li, Y., Guo, S., Xu, W., 2018. Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. Advances in Mechanical Engineering 10, 168781401880409. doi:10.1177/1687814018804096.

Xiong, H., Shi, S., Ren, D., Hu, J., 2022. A survey of job shop scheduling problem: The types and models. Computers & Operations Research 142, 105731. doi:10.1016/j.cor.2022.105731.

Xu, W., Hu, Y., Luo, W., Wang, L., Wu, R., 2021a. A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. Computers & Industrial Engineering 157, 107318. doi:10.1016/j.cie.2021.107318.

Xu, X., Lu, Y., Vogel-Heuser, B., Wang, L., 2021b. Industry 4.0 and Industry 5.0—Inception, conception and perception. Journal of Manufacturing Systems 61, 530–535. doi:10.1016/j.jmsy.2021.10.006.

Yang, X., Zeng, Z., Wang, R., Sun, X., 2016. Bi-Objective Flexible Job-Shop Scheduling Problem Considering Energy Consumption under Stochastic Processing Times. PLOS ONE 11, e0167427.

Yin, L., Li, X., Gao, L., Lu, C., Zhang, Z., 2017a. A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem. Sustainable Computing: Informatics and Systems 13, 15–30. doi:10.1016/j.suscom.2016.11.002.

Yin, L., Li, X., Gao, L., Lu, C., Zhang, Z., 2017b. A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem. Sustainable Computing: Informatics and Systems doi:10.1016/j.suscom.2016.11.002.

Zhang, G., Hu, Y., Sun, J., Zhang, W., 2020. An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. Swarm and Evolutionary Computation .

Zhang, H., Xu, G., Pan, R., Ge, H., 2021. A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. Engineering Optimization , 1–22doi:10.1080/0305215X.2021.1949007. publisher: Taylor & Francis.

Zhang, X., Ji, Z., Wang, Y., 2018. An improved SFLA for flexible job shop scheduling problem considering energy consumption. Modern Physics Letters B 32, 1840112. doi:10.1142/S0217984918401127.

Zhang, Z., Wu, L., Peng, T., Jia, S., 2019. An Improved Scheduling Approach for Minimizing Total Energy Consumption and Makespan in a Flexible Job Shop Environment. Sustainability 11. doi:10.3390/su11010179.

Zheng, X.l., Wang, L., 2016. A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. International Journal of Production Research 54, 5554–5566. doi:10.1080/00207543.2016.1170226. publisher: Taylor & Francis.

Zhu, H., Deng, Q., Zhang, L., Hu, X., Lin, W., 2020. Low carbon flexible job shop scheduling problem considering worker learning using a memetic algorithm. Optimization and Engineering 21, 1691–1716. doi:10.1007/s11081-020-09494-y.

Zhu, S., Xu, L., Goodman, E., Deb, K., Lu, Z., 2022. A general framework for enhancing relaxed Pareto dominance methods in evolutionary many-objective optimization. Natural Computing doi:10.1007/s11047-022-09889-z.

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report. doi:10.3929/ETHZ-A-004284029.

Şenyiğit, E., Atici, U., Şenol, M., 2022. Effects of OCRA parameters and learning rate on machine scheduling. Central European Journal of Operations Research 30, 941–959. doi:10.1007/s10100-020-00708-3.