



**HAL**  
open science

## NSGA-II for Solving a Multi-objective, Sustainable and Flexible Job Shop Scheduling Problem

Candice Destouet, Houda Tlahig, Belgacem Bettayeb, Bélahcène Mazari

► **To cite this version:**

Candice Destouet, Houda Tlahig, Belgacem Bettayeb, Bélahcène Mazari. NSGA-II for Solving a Multi-objective, Sustainable and Flexible Job Shop Scheduling Problem. APMS 2023: Production Management Systems for Responsible Manufacturing, Service, and Logistics Futures, IFIP Working Group 5.7 on Advances in Production Management Systems, Sep 2023, Trondheim ( Norvège), Norway. pp.548-562, 10.1007/978-3-031-43670-3\_38 . hal-04208130

**HAL Id: hal-04208130**

**<https://hal.science/hal-04208130>**

Submitted on 15 Sep 2023

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

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

# NSGA-II for solving a multi-objective, sustainable and flexible job shop scheduling problem

Candice Destouet<sup>1</sup> [0000-0002-7856-0011], Houda Tlahig<sup>1</sup> [0000-0002-6761-8265],  
Belgacem Bettayeb<sup>2</sup> [0000-0003-0997-9529], and Bélahcène Mazari<sup>3</sup>  
[0000-0003-1049-2654]

<sup>1</sup> CESI LINEACT, 15C Av. Albert Einstein, Villeurbanne, France,  
cdestouet@cesi.fr, htlahig@cesi.fr,

<sup>2</sup> CESI LINEACT, 8 Bd. Louis XIV, Lille, France,  
bbettayeb@cesi.fr,

<sup>3</sup> CESI LINEACT, 80 Av. Edmund Halley Rouen, Saint-Etienne-du-Rouvray, France,  
bmazari@cesi.fr

**Abstract.** Recently, sustainable scheduling has emerged to make the trade-off between economic, environmental and social factors. While classical scheduling research focuses on economic and environmental indicators such as makespan and energy consumption, modern human-centred manufacturing systems consider additional important factors such as workers' well-being and ergonomic risks. This study proposes a multi-objective mathematical model that jointly optimizes the makespan, the total energy consumption, and the OCRA (Occupational Repetitive Actions) index (a measure of ergonomic risk) in the case of a flexible job shop. The model has the originality of also taking into account the travel times of operators and products between machines. We use the NSGA-II method to solve it. The first results allow us to analyze the mutual influences of the criteria and demonstrate the advantage of such a method to obtain a good compromise between the three objectives in a reasonable resolution time.

**Keywords:** Industry 5.0, sustainability, flexible job shop, multi-objective scheduling, NSGA-II

## 1 Introduction

Nowadays, the technological advances generated by Industry 4.0 in addition to the recent emergence of Industry 5.0 have induced significant changes in production systems. Industry 4.0 has enhanced the digitization of information, the data sharing, and the automation of production lines, utilizing real-time production approaches, simulation tools, and communication technologies. This has further improved the performance of manufacturing systems. On the other hand, and because they have so far not been sufficiently considered, the Industry 5.0 paradigm has come to integrate environmental and social factors alongside technology, emphasizing the triptych: human, environment and resilience [1]. These new paradigms affect all a company's functions and have implications for

decision-making at different levels. At the operational level, production scheduling must consider additional constraints such as human and environmental factors, new decision-making strategies, and objectives aligned with “production lines 5.0” [2]. Such objectives may include minimizing energy consumption and maximizing operator well-being. However, the production scheduling problem is known to be NP-hard and finding a solution can be challenging. Various mathematical optimization models such as single-objective and multi-objective have been suggested, and different methods proposed to solve the problem, including Branch-and-Bound, constraint programming, genetic algorithm, tabu search, and ant colony optimization. They are discussed in the scientific literature, highlighting the complexity of the problem and the challenges that researchers face.

The aim of this paper is to address the Sustainable Flexible Job Shop Scheduling Problem (SFJSSP), a scheduling problem for a flexible job shop that integrates human and environmental factors. We propose a multi-objective optimization model that simultaneously minimizes the total execution time, total energy consumption, and exposure of operators to musculoskeletal risks, evaluated by the OCRA index presented by Occhipinti [3]. This index assesses the exposure of operators to tasks that involve various musculoskeletal risk factors of the upper limbs. Our research aims to analyze the trade-offs between the economic, environmental, and social aspects of the problem. To our knowledge, only a few studies have considered these three sustainability indicators simultaneously in a single multi-objective model [4–6], while constraints related to moving operators and products between machines are often simplified or ignored. The transportation times of products between machines introduce delays that directly affect the launch dates of subsequent tasks and the overall makespan [7]. Internal transport tasks in today’s automated factories are performed by robotic vehicles, which involves challenges and interesting issues for their optimization [8]. Moreover, our model considers operator flexibility, enabling them to switch between machines, which may introduce additional time delays and affect their fatigue levels [9]. As a result, our scheduling approach effectively accounts for these transport constraints. To solve this problem, we propose using the NSGA-II algorithm (Non-dominated Sorting Genetic Algorithm II), introduced by Deb et al. [10] and analyzing the results obtained by testing our model on instances from the literature. We structure the rest of this paper as follows: section 2 presents a brief overview of the SFJSSP problem and methods for solving multi-objective models. In Sections 3 and 4, we present our optimization model and describe the algorithm used to solve it. Section 5 presents our results and analysis, while Section 6 concludes our paper and presents some research perspectives.

## 2 Literature review

### 2.1 Sustainable scheduling

In the context of Industry 5.0, sustainability has become a growing concern, leading to a renewed interest in energy efficiency and operator well-being [11]. While Industry 4.0 has introduced new tools to enhance the efficiency and robustness of scheduling solutions, Industry 5.0 presents new challenges to pro-

duction systems that align with the changing economic, the social and the environmental ones. As a result, sustainable scheduling has emerged as a critical aspect of production systems. It is now essential to study the impact of environmental and social factors on scheduling with the same level of attention given to economic and technological considerations. In recent years, there has been a significant amount of research focused on studying the impact of environmental factors in scheduling problems, as highlighted by the work of Xiong et al. [12]. This study reviewed 297 papers published between 2016 and 2021, with 54 of them addressing environmental indicators such as energy consumption, carbon emission, or noise emission. More recently, we reviewed about 100 articles published over the past decade on the SFJSSP, and found that energy consumption has received the most attention among the various environmental factors [13], while carbon emission was studied in 22 papers. In most cases, the proposed models incorporated these factors into the objective function, 98% maximizing green production.

The consideration of the human factor in SFJSSP has received less attention in the literature [4]. According to our state-of-the-art [13], only 28% of the articles that incorporate operators consider them as part of the objective function, and even then, it's usually done through their cost. However, some articles aim to minimize ergonomic risks [5,6], worker fatigue [14], or operator workload [15,16]. In particular, Jaber and Neumann [17] and Sun et al. [18] consider rest time for workers to ensure their safety and efficiency. Their studies demonstrate that incorporating short breaks after each task can enhance the overall performance of the system. Some studies enforce fixed-length and predetermined breaks, while others propose a more proactive approach by optimizing them based on fatigue calculations [19]. This problem, known as work-rest scheduling, entails determining the optimal number, placement, and duration of rest times for workers. While minimizing the total execution time, production cost, or total delay are the most common objectives, Gong et al. [4] formulated the flexible and sustainable scheduling problem by considering the total execution time, ecological cost, and operator cost. In another study, Coca et al. [6] considered completion time as an economic indicator, and carbon dioxide emission, water consumption, and metal waste as environmental indicators. Additionally, they took labor intensity, noise, ambient temperature, and vibration amplitude as social indicators. Hongyu and Xiuli [5] proposed simultaneous consideration of makespan, energy consumption, and ergonomic risk as objectives. Like Gong et al. [20], they introduced an improved NSGA-III algorithm to solve the SFJSSP problem.

## 2.2 Resolution method for multi-objective problems

Handling multi-objective problems can be done in various ways. Four commonly used methods are discussed by Chiandussi et al. [21]: the global criterion method, the linear combination with weights, the *epsilon*-constrained method, and multi-objective genetic algorithms. The global criterion method is simple and efficient, aiming to minimize the distance of the solutions from an ideal solution. However, it can be computationally expensive and may not always provide non-dominated solutions in the Pareto sense. In the linear weight combination method, the ob-

jective function is a weighted sum of all considered objectives, converting a multi-objective problem to a mono-objective one. Nonetheless, it can be difficult to weigh each objective. The *epsilon*-constraint method selects a single objective and turns the others into constraints. Genetic algorithms are widely used and start with an initial population of solutions that evolve through iterations, aiming to converge towards the Pareto front [22]. In particular, NSGA-II [10] and NSGA-III [23] are two evolutionary algorithms that have shown promising results for the SFJSSP. NSGA-II is often combined with other methods like local search and meta-heuristics to improve it [24, 25]. NSGA-III, which differs from NSGA-II by its selection mechanisms, has also shown good results [5]. Additionally, reinforcement learning models have been developed more recently to select decision rules for multi-objective problems [26].

### 3 Problem description

#### 3.1 Assumptions

In this study, we propose a nonlinear integer program for the SFJSSP that considers both environmental and human factors. Specifically, we model three types of energy-consuming activities during the process: the execution of operations on machines, the internal transport of products, and the auxiliary installations such as lighting and heating. Our approach acknowledges the importance of the human factor by considering various key aspects. Firstly, we consider the operators' skills, recognizing that the duration of tasks depends on the operator assigned to them. Secondly, we account for the travel time between machines, assuming it to be consistent for all workers. Lastly, we prioritize operator safety by assessing their rest times, which include breaks following each operation, and monitoring their OCRA indexes. The OCRA index is calculated based on factors such as posture, force, repetitiveness, additional risks (e.g., vibration, temperature), lack of rest, and overall process duration. According to the European standard EN 1005 [27] and ISO 11228-3 [28], this index is recommended for analyzing and predicting the occurrence of musculoskeletal disorders (TMS). An OCRA index below 2.2 indicates no risk for the worker, while an index between 2.2 and 3.5 indicates a low risk of TMS, and an index above 3.5 is considered hazardous [29]. The SFJSSP involves a set of jobs  $\mathcal{J} = j_1, j_2, \dots, j_J$  divided into operations  $O_j = o_{j1}, o_{j2}, \dots, o_{jk}$  that need to be processed in a predefined given sequence on a set of machines  $\mathcal{M} = m_1, m_2, \dots, m_M$ . Our model includes the assignment of each operation to an operator and aims to find the optimal assignment of machines and operators for each operation, as well as an optimal sequence of operations on each machine and operator. While an operation can be performed by any operator on any machine, its duration depends on the operator and the machine performing it. Our model minimizes the total execution time, the total energy consumption, and the maximum OCRA. We assume that (1) each operation requires one operator and one machine, (2) machines and operators can perform one operation at a time, (3) operators rest for 8% of the time, (4) transport time between machines is the same for all products but depend on the machines, and (5) the time horizon consists of two 7-hour periods per day. It is worth noting that these assumptions may vary depending on the company.

### 3.2 Mathematical model

#### 3.2.1. Notations

The sets, indices and parameters used are :

$\mathcal{W}$	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 tasks in $j$ noted $O_{kj}$ , $k = 1..K_j$
$\mathcal{T}$	set of periods, $t = 1..T$ .
$d_t, f_t$	starting and ending time of period $t$ .
$p_{k_j w m}$	processing time of $O_{k_j}$ on machine $m$ with worker $w$ .
$s_{k_j m k'_{j'}}$	setup time on machine $m$ between operations $O_{k_j}$ and $O_{k'_{j'}}$ .
$tp_{mm'}$	transport time for a product between machines $m$ and $m'$ .
$t_{mm'}$	travel time for a worker between machines $m$ and $m'$ .
$e_{mm'}^{trans}$	energy consumption for transporting a product from $m$ to $m'$ .
$e^{aux}$	energy consumption of auxiliary equipment per unit time.
$ata_{k_j m}$	number of technical actions to execute $O_{k_j}$ on machine $m$ .
$pm_{k_j m}$	posture multiplication factor for executing $O_{k_j}$ on machine $m$
$rm_{k_j m}$	repetitiveness multiplication factor for executing $O_{k_j}$ on machine $m$
$fm_{k_j m}$	strength multiplication factor for executing $O_{k_j}$ on machine $m$
$arf_{k_j i}$	additional risks multiplication factor for executing $O_{k_j}$ on $m$
$rcm$	multiplication factor for lack of rest.
$dum$	multiplication factor for the overall duration of repetitive tasks.
$cf$	constant frequency of technical actions by minute.

The decision variables are noted as follows :

$S_{k_j}, C_{k_j}$	starting and ending time of $O_{k_j}$ ;
$Cmax$	makespan;
$R_{k_j w t}$	resting time for worker $w$ after $O_{k_j}$ .
$V_{tw}$	binary variable equal to 1 if worker $w$ is assigned to period $t$ .
$X_{k_j w m t}$	binary variable equal to 1 if $O_{k_j}$ is executed on machine $m$ by worker $w$ on period $t$ .
$Y_{k_j w k'_{j'}}$	binary variable equal to 1 if $O_{k_j}$ is done by worker $w$ before $O_{k'_{j'}}$ .
$Z_{k_j m k'_{j'}}$	binary variable equal to 1 if $O_{k_j}$ is done on machine $m$ before $O_{k'_{j'}}$ .
$N_{k_j m k'_{j' m'}}$	binary variable equal to 1 if $O_{k_j}$ 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 auxiliaries energy consumption.
$ATA_{wt}$	real number of technical actions for worker $w$ on period $t$ .
$RTA_{wt}$	recommended number of technical actions for worker $w$ on period $t$ .
$OCRA^{max}$	OCRA maximum for all workers and periods.

#### 3.2.2. Objective functions

Our objective functions aim to minimize the total execution time of the opera-

tions (2), the total energy consumption required for the overall process (3), and the maximum OCRA value among all operators (4). By minimizing these three factors simultaneously, we can achieve an optimal solution that balances the trade-off between production efficiency, energy efficiency, and operator safety.

$$\min \{f_1, f_2, f_3\} \quad (1)$$

$$f_1 : Cmax \quad (2)$$

$$f_2 : ET + EM + EC \quad (3)$$

$$f_3 : OCRA^{max} \quad (4)$$

### 3.2.3. Constraints

The constraints (5) defines the total execution time, i.e. the end date of the last operation. The constraints (6), (7) and (8) define the energy consumption related to transport, task execution, and auxiliary energy respectively.

$$Cmax \geq C_{kj} \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \quad (5)$$

$$ET = \sum_j \sum_k \sum_m \sum_{m'} N_{kjm(k+1)jm'} \times e_{mm'}^{trans} \quad (6)$$

$$EM = \sum_j \sum_k \sum_m \sum_w \sum_t X_{kjwt} \times e_{kjwt} \quad (7)$$

$$EC = Cmax \times e^{aux} \quad (8)$$

The constraints (9) to (13) are those related to the operators. (9) state that an operator is assigned to one period on two, (10) ensure sufficient rest time. (11) and (12) compute the actual and recommended number of technical actions for each operator in each period, and (13) compute  $OCRA^{max}$ .

$$V_{tw} + V_{(t+1)w} = 1 \quad \forall t \in \mathcal{T}, w \in \mathcal{W} \quad (9)$$

$$R_{kjwt} = 0.08 \times \sum_m X_{kjwt} \times p_{kjwt} \quad \forall t \in \mathcal{T}, w \in \mathcal{W}, j \in \mathcal{J}, k \in \mathcal{K}_j \quad (10)$$

$$ATA_{wt} = \sum_j \sum_k \sum_m ata_{kjm} X_{kjwt} \quad \forall w \in \mathcal{W}, t \in \mathcal{T} \quad (11)$$

$$RTA_{wt} = (rcm \times dum) \sum_j \sum_k \sum_m cf \times pm_{kjm} \times rm_{kjm} \times fm_{kjm} \times arf_{kjm} \\ \times X_{kjwt} p_{kjwt} \quad \forall t \in \mathcal{T}, w \in \mathcal{W} \quad (12)$$

$$OCRA^{max} \geq \frac{ATA_{wt}}{RTA_{wt}} \quad \forall t \in \mathcal{T}, w \in \mathcal{W} \quad (13)$$

The constraints (14), (15) and (16) are precedence constraints for two tasks of the same job, two tasks assigned to the same machine, and two tasks assigned

to the same operator, respectively.

$$S_{(k+1)j} \geq C_{kj} + \sum_m \sum_{m'} N_{kjm(k+1)jm'} \times tp_{mm'} \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \quad (14)$$

$$S_{k'j'} + (1 - \sum_m Z_{kjm k'j'})M \geq C_{kj} + s_{kjm k'j'} \quad \forall j, j' \in \mathcal{J}, k, k' \in \mathcal{K}_j, \mathcal{K}_{j'} \quad (15)$$

$$S_{k'j'} + (1 - \sum_w Y_{k j w k'j'})M \geq C_{kj} + \sum_w \sum_t R_{k j w t} + \sum_m \sum_{m'} N_{kjm(k+1)jm'} \times t_{mm'} \quad \forall j, j' \in \mathcal{J}, k, k' \in \mathcal{K}_j, \mathcal{K}_{j'} \quad (16)$$

The constraints (17) state that an operation is executed by exactly one operator and one machine. Constraints (18) and (19) ensure that a task starts and ends in the same period.

$$\sum_w \sum_m \sum_t X_{k j w m t} = 1 \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_j \quad (17)$$

$$C_{kj} - (1 - \sum_w \sum_m X_{k j w m t})M \leq f_t \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_j, t \in \mathcal{T} \quad (18)$$

$$S_{kj} + (1 - \sum_w \sum_m X_{k j w m t})M \geq d_t \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_j, t \in \mathcal{T} \quad (19)$$

We also consider constraints to link variables (e.g., an operator cannot perform tasks on a period where he is not assigned) and non-negativity constraints, which we do not describe here.

It has been shown that FJSSP is an NP-hard problem, so SFJSSP is also NP-hard [5]. Solving the problem with a solver is only possible on small instances. To obtain a solution close to the optimal solution in a reasonable time, we have implemented the NSGA-II algorithm, described in the next section

## 4 Algorithm description

The NSGA-II algorithm has been widely used to solve multi-objective scheduling problems [30–32]. It is based on the principle of Pareto dominance, which is explained in Section 4.2. The algorithm starts with a population  $P_0$  consisting of  $N$  solutions. At each iteration, we create a child population  $Q_t$  of size  $N$  using crossover and mutation operators. We combine  $P_t$  and  $Q_t$  to obtain a population  $R_t$  that contains  $2N$  solutions. We keep the top  $N$  solutions to compose the next population  $P_{t+1}$ . These solutions are selected based on their level of dominance to ensure elitism, added with a “crowded distance” method that selects distant solutions to ensure diversity. The following subsections describe in detail the operation of the algorithm.

### 4.1 Coding and decoding

#### 4.1.1. Solutions representation

A solution is described by an individual composed of three different chromosomes, representing our three sub-problems (Figure 1). The first one describes

the Operations Sequence (OS). As this is fixed for each job, the  $k^{th}$  appearance of job  $j$  in the sequence represents  $o_{kj}$ . The second and third chromosomes represent the Machine Allocation and Worker Allocation problems respectively (MA and WA). In this example,  $O_{11}$  is executed on machine 2 by operator 2,  $O_{12}$  on machine 1 by operator 1, etc.

	$o_{11}$	$o_{12}$	$o_{21}$	$o_{31}$	$o_{32}$	$o_{13}$	$o_{22}$	$o_{23}$
OS	1	1	2	3	3	1	2	2
MA	2	1	3	3	3	1	2	2
WA	2	1	1	2	1	1	2	2

Fig. 1: Chromosomes representing our solutions

#### 4.1.2. Crossover operations

Crossover operations are an essential part of this algorithm for exploring the neighborhood of good solutions. By exchanging genes from two different solutions, we can create new individuals that inherit some of the beneficial traits of their parents. In our approach, we use three separate crossover operations for each of the three chromosomes representing our three sub-problems: OS, MA, and WA. These operations do not necessarily target the same operation across all three chromosomes. On OS, we employ the crossover method shown in Figure 2a. We randomly select a set of operations from both parents, e.g.,  $O_{12}$ ,  $O_{32}$ , and  $O_{22}$ . We then create two children: one by copying these operations in the same order from one parent while adding the missing operations from the other parent, and the contrary for the other child. To perform a crossover on MA and WA, we select a set of random operations from both parents and swap their machine and operator allocations. Figure 2b illustrates this process.

#### 4.1.3. Mutation operations

Mutation is an important technique to promote diversity in solutions and avoid getting stuck in local optima. After performing crossover operations, we apply mutation operations with a certain probability. For OS, we randomly select an operation and move it to another position in the sequence, as shown in Figure 3a. For MA and WA, we randomly select an operation and replace its assigned machine/operator with another, randomly selected one (Figure 3b).

#### 4.1.4. Calculation of the fitness function

Each solution is evaluated by a fitness function that includes three elements, corresponding to the three objective functions: total execution time, total energy consumption, and maximum OCRA. To calculate these elements, we decode the three chromosomes for each solution. Firstly, we take the first operation from the OS chromosome. Using MA and WA chromosomes, we determine the machine and operator on which to assign the operation, and thus calculate its start and

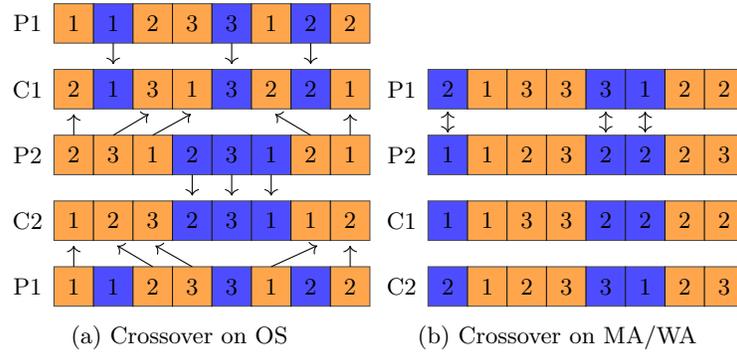


Fig. 2: Crossover operations.

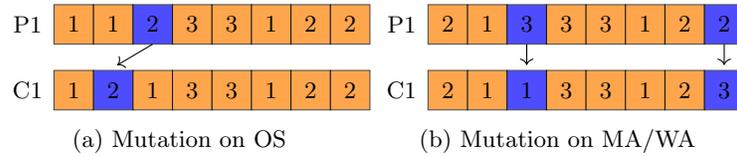


Fig. 3: Mutation operations.

end date. Next, we select the next operation from the OS chromosome and repeat the process until we have a complete schedule. To ensure that all precedence constraints are respected, we calculate the start date of each operation as the maximum date between: (1) the end date of the previous task of the same job added to the transport time between the two respective machines, (2) the end date of the previous job assigned to the machine plus the setup time between the two operations, and (3) the end date of the previous task assigned to the operator added to the travel time between the two machines and his rest time. If the end date of a task exceeds the end date of the period, the start of the task is postponed to the beginning of the next period to which the worker is assigned. Finally, we use the obtained schedule to calculate the total execution time, the total energy consumption, and the maximum OCRA between all operators and periods.

#### 4.2 Dominance

To compare the different solutions and converge towards the best ones, we employ the concept of Pareto dominance. We define the set of non-dominated solutions as the ones that cannot be improved in any objective without worsening another. Let  $nb$  be the number of objectives, which is three in this case. We have the following definition:

$v$  dominate  $u$  ( $u \prec v$ ) if and only if :

$$\begin{cases} \forall i \in 1, \dots, nb & f_i(v) \leq f_i(u) \\ \exists i \in 1, \dots, nb & f_i(v) < f_i(u) \end{cases}$$

To approach the Pareto front, we adopt an iterative approach that involves selecting solutions that are closest to the front at each step. We start by identifying the solutions that are not dominated by any other solutions in the set, and we assign them the rank of 0, which constitutes the first front  $\mathcal{F}_0$ . Among the remaining solutions, we classify those which are not dominated by any other in the front  $\mathcal{F}_1$ , we give them the rank 1. This process continues until all solutions have been assigned a rank, and we obtain a set of fronts  $\mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_n$  with  $n$  being the number of fronts. At each step of the algorithm, we keep the solutions of the first  $l$  fronts as long as the number of solutions does not exceed the desired population size. If we have fewer solutions than the desired population size, we consider the next front  $\mathcal{F}_{l+1}$  and select the solutions that are most diverse using the “crowded distance” method, which is described in the next subsection.

### 4.3 Crowded distance method

To explore a diverse set of solutions, we use the “crowded distance” method to ensure that the solutions we select are spread out across the search space. For each solution, we calculate its distance to all other solutions as the sum of the distances according to the three objective functions. A solution with a low distance value is surrounded by other solutions, while a solution with a high distance value is far away from any other solution. We prioritize solutions with high distance values, as they represent unexplored areas of the search space. To compare two solutions, we use the “crowded comparison operator”. First, we compare their ranks; if they are different, we select the solution with the lowest rank to ensure elitism. If their ranks are the same, we select the solution with the largest distance value to ensure diversity.

### 4.4 Initialization

To initialize our population at the start of the algorithm, we follow a few different rules. First, we generate the sequence of operations (OS) randomly. Then, we generate the assignment of tasks to machines and workers (MA and WA) using four different rules. For 25% of the solutions, we randomly generate the assignments. For another 25%, we select the machine/operator pair that minimizes the processing time for each task. Another 25% of the solutions are generated by selecting the machine/operator pair that minimizes the energy consumption for each task. Finally, for the remaining 25% of the solutions, we select the machine that minimizes the OCRA index and assign it to the operator with the lowest OCRA. By using a mix of random and optimized assignment rules, we hope to generate a diverse initial population to start our search for the best solutions.

### 4.5 Algorithm steps

The algorithm is presented in Figure 4. It begins by generating an initial population using a combination of random and heuristic methods. In each iteration, the non-dominated solutions are sorted into fronts, and the solutions closest to the Pareto front are selected to form the next generation. To ensure diversity in the population, the crowded distance operator is used. The algorithm stops either when a certain number of iterations is reached, or when no improvement is made in the current generation.

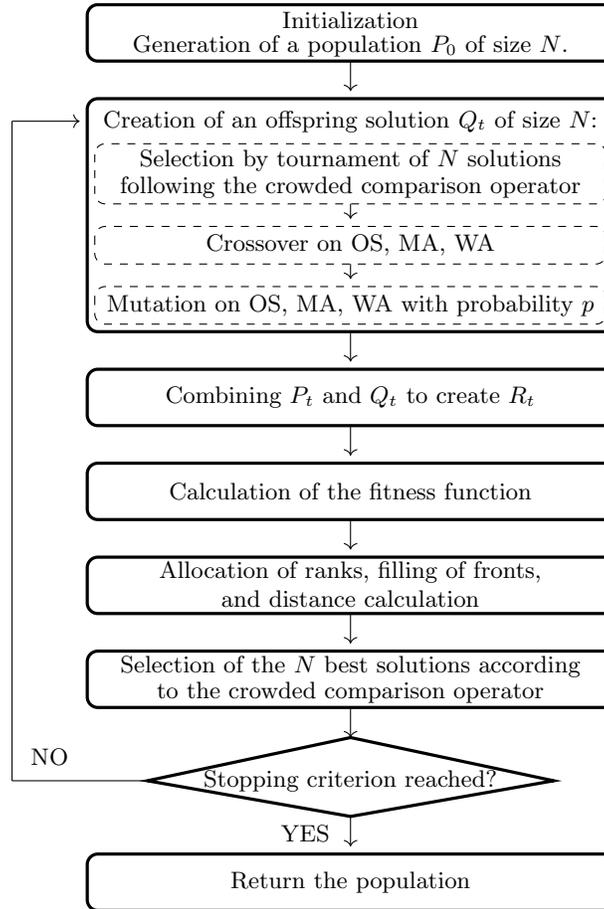


Fig. 4: Algorithm description

## 5 Results and analysis

To evaluate the effectiveness of our algorithm, we conducted experiments on a small instance inspired by Kacem et al. [33]. To generate transportation, energy consumption, and human factors data, we used random data. The instance consists of 5 machines, 3 operators, and 4 jobs, which are divided into 12 tasks. To obtain the optimal solution for each objective function, we implemented this instance in Python and solved the problem using the Gurobi solver for two criteria: total execution time and energy consumption. The resulting solutions represent the optimal solutions for minimizing total execution time and for minimizing energy consumption independently. These two solutions are marked in yellow on the graphs in Figures 5, 6a, 6b, and 6c. The displayed solutions are the last population obtained. In red, we show the solutions that are on the  $\mathcal{F}_0$

front, i.e., the solutions that are not dominated by any other solutions. In blue, we show the solutions that are on the upper fronts and thus are dominated at least by those in red.

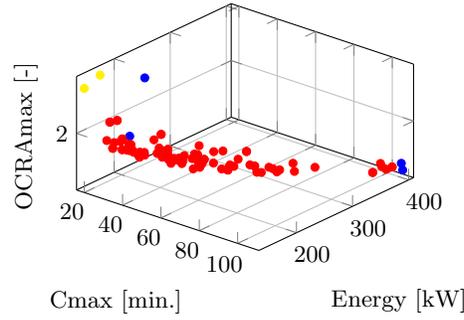


Fig. 5: Display of our solutions according to the three objectives

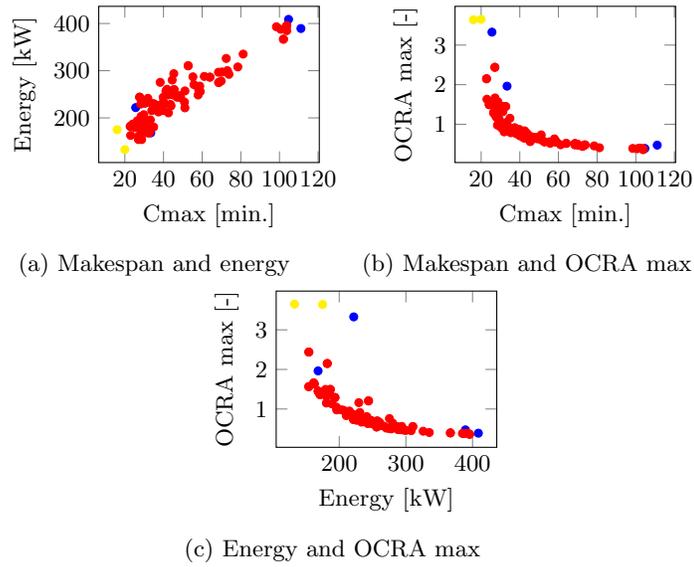


Fig. 6: Display of solutions on two-by-two axes

The figures displayed represent the results obtained from our algorithm, using a population of 100 individuals, with a minimum of 200 iterations completed within a resolution time of about 180 seconds. They are comparable to the optimal solutions for single-objective models obtained with the GUROBI solver. Fig-

ure 6a illustrates the correlation between total execution time and total energy consumption, where longer process times lead to higher energy consumption. This relationship is partly due to the integration of auxiliary energies that are directly related to the makespan. However, the OCRA max objective is in contradiction with the other two objectives. Despite this, it is still possible to find a scheduling that achieves favorable outcomes in both energy consumption and makespan while maintaining a reasonable OCRA max to ensure worker safety. The results demonstrate that our algorithm is capable of achieving a favorable trade-off between the three objectives in a reasonable time. Further analysis are necessary to determine which scheduling solution to apply. One possible approach is to keep solutions where the OCRA max is lower than 2.2, and select the one with the best execution time and/or energy consumption.

## 6 Conclusion and perspectives

This paper presented a novel multi-objective model that accounts for economic, environmental, and human factor considerations in a flexible job shop. To find Pareto-optimal solutions, we implemented the NSGA-II algorithm, which is renowned for its ability to solve multi-objective models. Our results suggest that the algorithm can produce effective trade-offs between the three objectives, while maintaining reasonable resolution times for the tested instances. In future research, we aim to explore several perspectives that could further enhance the proposed approach by: (1) evaluating our model on larger instances, ideally based on real-world data, (2) optimizing the exploration parameters of the algorithm to improve its performance, (3) evaluating the quality of the solutions obtained using metrics such as hypervolume, (4) refining the objectives related to the human factor, such as minimizing the gap between OCRA max and the optimal working conditions threshold of 2.2, and considering fatigue curves that vary according to the workload and operator's breaks, (5) integrating different criteria to choose a single solution among the set of obtained solutions and determine the optimal scheduling, (6) making the model dynamic and able to react to hazards and (7) testing the NSGA-III algorithm (which differ from NSGA-II by its selection mechanisms) to solve our problem and compare the two algorithms obtained results.

## References

1. X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0—inception, conception and perception," *Journal of Manufacturing Systems*, 2021.
2. M. Parente, G. Figueira, P. Amorim, and A. Marques, "Production scheduling in the context of industry 4.0: review and trends," *International Journal of Production Research*, 2020.
3. E. Occhipinti, "Ocra: a concise index for the assessment of exposure to repetitive movements of the upper limbs," *Ergonomics*, 1998.
4. G. Gong, Q. Deng, X. Gong, W. Liu, and Q. Ren, "A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators," *Journal of Cleaner Production*, 2018.

5. L. Hongyu and W. Xiuli, "A survival duration-guided nsga-iii for sustainable flexible job shop scheduling problem considering dual resources," *IET Collaborative Intelligent Manufacturing*, 2021.
6. G. Coca, O. Castrillón, S. Ruiz, J. Mateo-Sanz, and L. Jiménez, "Sustainable evaluation of environmental and occupational risks scheduling flexible job shop manufacturing systems," *Journal of Cleaner Production*, 2019.
7. S. Homayouni and D. B. M. M. Fontes, "Production and transport scheduling in flexible job shop manufacturing systems," *Journal of Global Optimization*, 2021.
8. K. Sanogo, A. Mekhalef Benhafssa, M. Sahnoun, B. Bettayeb, M. Abderrahim, and A. Bekrar, "A multi-agent system simulation based approach for collision avoidance in integrated job-shop scheduling problem with transportation tasks," *Journal of Manufacturing Systems*, 2023.
9. W. Tan, X. Yuan, J. Wang, and X. Zhang, "A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced nsga-ii: An application from casting workshop," *Computers & Industrial Engineering*, 2021.
10. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, 2002.
11. S. Grabowska, S. Saniuk, and B. Gajdzik, "Industry 5.0: improving humanization and sustainability of industry 4.0," *Scientometrics*, 2022.
12. H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Computers & Operations Research*, 2022.
13. C. Destouet, H. Tlahig, B. Bettayeb, and B. Mazari, "Flexible job shop scheduling problem under industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement," *Journal of Manufacturing Systems*, 2023.
14. W. Tan, X. Yuan, J. Wang, and X. Zhang, "A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced nsga-ii: An application from casting workshop," *Computers & Industrial Engineering*, 2021.
15. S. Luo, L. Zhang, and Y. Fan, "Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization," *Journal of Cleaner Production*, 2019.
16. M. Sano, M. Nagao, and Y. Morinaga, "Balancing setup workers load of flexible job shop scheduling using hybrid genetic algorithm with tabu search strategy," *International Journal of Decision Support Systems*, 2016.
17. M. Jaber and W. Neumann, "Modelling worker fatigue and recovery in dual-resource constrained systems," *Computers & Industrial Engineering*, 2010.
18. X. Sun, S. Guo, J. Guo, and B. Du, "A hybrid multi-objective evolutionary algorithm with heuristic adjustment strategies and variable neighborhood search for flexible job-shop scheduling problem considering flexible rest time," *IEEE Access*, 2019.
19. S. Xu and N. Hall, "Fatigue, personnel scheduling and operations: Review and research opportunities," *European Journal of Operational Research*, 2021.
20. X. Gong, T. De Pessemer, L. Martens, and W. Joseph, "Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation," *Journal of Cleaner Production*, 2019.
21. G. Chiandussi, M. Codegone, S. Ferrero, and F. Varesio, "Comparison of multi-objective optimization methodologies for engineering applications," *Computers & Mathematics with Applications*, 2012.
22. G. Zhang, Y. Hu, J. Sun, and W. Zhang, "An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints," *Swarm and Evolutionary Computation*, 2020.

23. K. Deb and H. Jain, "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*, 2014.
24. X. Liang, J. Chen, X. Gu, and M. Huang, "Improved adaptive non-dominated sorting genetic algorithm with elite strategy for solving multi-objective flexible job-shop scheduling problem," *IEEE Access*, 2021.
25. Q. Luo, Q. Deng, G. Xie, and G. Gong, "A pareto-based two-stage evolutionary algorithm for flexible job shop scheduling problem with worker cooperation flexibility," *Robotics and Computer-Integrated Manufacturing*, 2023.
26. H. Wang, J. Cheng, C. Liu, Y. Zhang, S. Hu, and L. Chen, "Multi-objective reinforcement learning framework for dynamic flexible job shop scheduling problem with uncertain events," *Applied Soft Computing*, 2022.
27. N. E. 1005-1+A1, "Cen/tc 122," *Ergonomie*, 2008.
28. I. 11228-3:2006, "Iso/tc 159/sc 3," *Ergonomie*, 2007.
29. S. D. Akyol and A. Baykasoglu, "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*, 2019.
30. W. Song, C. Zhang, W. Lin, and X. Shao, "Flexible job-shop scheduling problem with maintenance activities considering energy consumption," *Applied Mechanics and Materials*, 2014.
31. X. Yang, Z. Zeng, R. Wang, and X. Sun, "Bi-objective flexible job-shop scheduling problem considering energy consumption under stochastic processing times," *PLOS ONE*, 2016.
32. M. Amjad, S. Butt, R. Kousar, R. Ahmad, M. Agha, Z. Faping, N. Anjum, and U. Asgher, "Recent research trends in genetic algorithm based flexible job shop scheduling problems," *Mathematical Problems in Engineering*, 2018.
33. I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic," *Intelligent Forecasting, Fault Diagnosis, Scheduling, and Control*, 2002.