Journal of Project Management 9 (2024) 493–512

Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

NSGA-II simheuristic to solve a multi-objective flexible flow shop problem under stochastic machine breakdowns

Daniel Felipe Rodríguez-Espinosa[a], Daniela Cruz-Vargas[a], Daniel Esteban Delgado-Merchán[a], David Hernando Gonzalez-Estupiñán[a] and Eliana María González-Neira[a*]

[a]*Departamento de Ingeniería Industrial, Pontificia Universidad Javeriana, Bogotá, Colombia*

C H R O N I C L E

Article history:
Received: February 2, 2024
Received in revised format: May 30, 2024
Accepted: June 21, 2024
Available online:
June 21, 2024

Keywords:
Stochastic flexible flow shop
Machine breakdowns
NSGA-II
Tardy jobs
Makespan

A B S T R A C T

This study proposes a simheuristic that hybridizes NSGA-II with Monte Carlo simulation to address a stochastic flexible flow shop problem featuring stochastic machine breakdowns. In real-world scenarios, machine breakdowns frequently occur, resulting in negative impacts such as time loss, late deliveries, decreased productivity, and order accumulation. Therefore, this study considers the times between failures and times to repair as stochastic parameters. Multiple objectives are concurrently addressed, including expected makespan, expected tardy jobs, and the standard deviation of tardy jobs. A mathematical model was formulated for the deterministic version of the problem and separately solved for the minimization of tardy jobs and the minimization of makespan in small instances. Subsequently, the proposed simheuristic was executed for both small and large instances. The results demonstrate that the NSGA-II simheuristic enhances outcomes across all objective functions compared to the simulation of optimal solutions provided by the mathematical models in small instances, yielding average GAPs of -16.64%, -21.87%, and -53.33% for expected tardy jobs, expected makespan, and standard deviation of tardy jobs, respectively. Furthermore, the simheuristic outperforms the simulation of solutions given by seven dispatching rules, showcasing average improvements of 48.01%, 48.18%, and 95.63% for the same objectives, respectively.

1. Introduction

A flexible flow shop scheduling problem (FFS) consists of a series of production stages, wherein at least one of them has two or more parallel machines, and all jobs must follow the same route. Jobs flow from one stage to another, being processed by only one machine at each stage (Pinedo, 2012). FFS environments have been extensively studied due to their adaptability in real-world problems (Rajendran & Chaudhuri, 1992). These environments are commonly found in industries such as chemical, electronics manufacturing, pharmaceuticals, automotive, glass container fabrication, and others (Azadeh et al., 2018). A suitable scheduling model should take into account all uncertainty conditions to address real-world problems (Ebrahimi et al., 2014). In recent years, most work involving FFS has been conducted under deterministic parameters, with few studies considering uncertainty (González-Neira et al., 2017). Given that the majority of works are deterministic, using known or pre-established data, it is important to consider stochastic data that allows anticipating unpredictable scenarios within FFS problems. Based on the literature reviewed for this project, which covers articles published since 2010, it is evident that the majority of research efforts on stochastic FFS (SFFS) primarily focus on uncertain processing times. In contrast, fewer studies have delved into the impact of machine breakdowns as an uncertain event, a trend confirmed by Mirabi et al. (2013). However, machine breakdowns stand out as one of the factors with the most significant impact on late jobs and production losses in real-world environments. The use of the exponential distribution to model failures is common due to its tractability and ease of understanding, along with demonstrated good approximations for modeling failures (Das, 2008). Given these considerations and the precedent set by works such as those by Zandieh and Gholami (2009), Zandieh

* Corresponding author.
E-mail address: eliana.gonzalez@javeriana.edu.co (E. M. González-Neira)

doi: 10.5267/j.jpm.2024.6.002

and Hashemi (2015), and Silva et al. (2017), which employed the exponential distribution to model both times between failures and times to repair, this project will also implement the exponential distribution.

Concerning the type of parallel machines in FFS under uncertainty, it was found that most of the analyzed articles focused on identical parallel machines. However, in real-world applications, it is common to encounter machines with different technologies in a stage (Chen & Chen, 2009). For that reason, the analysis of unrelated parallel machines is necessary. Some examples of real applications can be found in drilling operations in printed circuit board manufacturing (Hsieh et al., 2003), dicing in compound semiconductor fabrication (Kim et al., 2003), ceramic tile manufacturing systems (Ruiz & Maroto, 2006), among others. In the realm of objective functions, most of the research in FFS under uncertainty has predominantly focused on the expected makespan as a single objective. However, this project diverges by exploring multiple objectives simultaneously. Specifically, three objective functions will be considered. The first, makespan, is chosen for its efficacy in reducing job lateness, total work-in-process inventories, and shop flow congestion due to uncompleted jobs (Tavakkoli-Moghaddam et al., 2009). The second objective is the number of tardy jobs, directly correlated with the percentage of on-time deliveries—a pivotal metric for evaluating managers across various industries (Allahverdi et al., 2016). The third objective is the standard deviation of tardy jobs, chosen because, as stated by Liu et al. (2011), the variability of delays serves as a robust indicator.

To tackle multiple objectives, a posteriori scheme is chosen to obtain the set of non-dominated solutions, known as the Pareto frontier, illustrating the trade-off between the desired objectives. The Elitist Non-Dominated Sorting Genetic Algorithm II (NSGA-II), proposed by Deb et al. (2000), serves as a metaheuristic that provides the Pareto frontier. It has demonstrated a better spread of solutions and superior convergence near the true Pareto-optimal front in various applications such as in (Wang et al., 2017; Singh & Shukla, 2020; Yu et al., 2020).

To address uncertainties, one of the primary and successfully implemented methods is the simheuristic (Juan et al., 2015). This approach involves integrating simulation into a metaheuristic-driven framework, capitalizing on the benefits of fast executions of metaheuristics while handling uncertain conditions. Simheuristics prove particularly valuable in scheduling applications. Examples of simheuristic applications in scheduling can be found in works by Juan et al. (2014), Gonzalez-Neira et al. (2017), Caldeira and Gnanavelbabu (2021), and Abu-Marrul et al. (2023). Therefore, the proposed simheuristic will hybridize NSGA-II with Monte Carlo simulation, a proven effective approach in solving a Berth allocation problem (de León et al., 2021) and a stochastic flexible job shop problem (Rodríguez-Espinosa et al., 2023).

Considering the mentioned elements, this project aims to study a SFFS with stochastic machine breakdowns to minimize and obtain the Pareto front of expected makespan, expected tardy jobs, and standard deviation of tardy jobs.

The remainder of the paper is organized as follows. Section 2 contains the state of the art in stochastic FFS. Section 3 presents the mixed integer linear programming for the deterministic version of the problem. Section 4 explains the proposed multi-objective simheuristic approach. Section 5 details the computational experiments performed. Finally, section 6 provides conclusions and future research.

2. Literature review

As this project aims to address a multi-objective stochastic FFS (SFFS), this section presents a literature review of the SFFS focused on three main aspects that are shown in Table 1 for each article reviewed: (i) characteristics related to the shop environment, such as type of parallel machines, inclusion of setup times, limited buffers, among others; (ii) objective function(s); (iv) stochastic parameter(s); and (iv) solution method. From these papers, the following conclusions can be highlighted:

- Regarding the types of parallel machines, approximately 35% of the articles investigate unrelated parallel machines, considering various conditions like no-wait, sequence-dependent setup times, limited buffers, among others. The remaining 65% focus on identical parallel machines.
- Concerning objective functions, two-thirds of the literature addresses single-objective problems, with a predominant emphasis on makespan minimization. Only 32% of the papers are multi-objective, and late deliveries comprise 18% of the objective functions.
- Stochastic processing times are included in three-quarters of the articles, while stochastic machine breakdowns are analyzed in only 18% of them.
- Based on the implemented solution methods, 35% of the articles employ a genetic algorithm with different variations based on their interests. The remaining articles use other metaheuristics or hybridizations such as simulated annealing, particle swarm optimization, variable neighborhood search, among others.

The literature review reveals two works related to our proposal, with distinctions that highlight our contribution. The study by (Ebrahimi et al., 2014) shares similarities in using NSGA-II and optimizing makespan and total tardiness. However, differences arise in our case study, which considers unrelated parallel machines, introduces a robust approach involving the standard deviation of tardy jobs, and addresses uncertain machine breakdowns—more prevalent in real industries than the due dates considered by Ebrahimi et al. Additionally, the work by (Zandieh & Hashemi, 2015), while involving unrelated parallel machines and stochastic breakdowns, minimizes a single objective function (expected makespan), whereas our project incorporates multiple objective functions and a robust approximation.

Table 1
Literature review on stochastic flexible flow shop

Reference	Objective Function	Type of parallel machines	Parameters under uncertainty	Solution method
Wang & Choi (2010)	Makespan	Identical	Setup times Processing times	Decomposition based approach that integrates genetic algorithm and shortest process time rule
Al-Turki et al. (2012)	Average flow	Identical	Processing times	Simulation model using ARENA
Choi & Wang (2012)	Makespan	Identical	Processing times	Decomposition-based approach that hybridizes shortest processing time rule genetic algorithm
Kianfar et al. (2012)	Mean tardiness	Identical	Dynamic arrival	Hybridization of dispatching rule and hybrid genetic algorithm, along with a discrete event simulation model
Almeder & Hartl (2013)	Utilization of machine and buffers	Unrelated	Processing times	Discrete event simulation with variable neighborhood search
J. T. Lin et al. (2013)	Makespan	Unrelated	Processing times	Simulation optimization method, employing a combination of particle swarm optimization and optimal computing budget allocation
Mirabi et al. (2013)	Makespan	Identical	Breakdowns	Firstly, an optimal approach for job precedence with a single machine in both stages. Subsequently, a heuristic algorithm for scenarios involving M machines.
Rahmani et al. (2013)	Makespan	Identical	Breakdowns	Reactive method that uses stability and nervousness measures
Wang et al. (2013)	Makespan	Identical	Breakdowns Processing times	Cluster-based scheduling model that amalgamates shortest processing time rule with simulated annealing
Ebrahimi et al. (2014)	Makespan; Tardiness	Identical	Due dates	NSGA-II and Multi Objective Genetic Algorithm separately
Wang & Choi (2014)	Makespan	Identical	Processing times	Decomposition based holonic approach that involves k-means clustering, back propagation networks genetic algorithm and shortest processing time
Wang et al. (2014)	Makespan	Identical	Processing times	Two phase simulation-based estimation of distribution algorithm
Jiang et al. (2015)	Waiting time; Earliness/T ardiness	Identical	Processing times	The problem is decomposed into a Parallel Machine Scheduling Problem and HFS. Hybrid differential evolution with VNS addresses the parallel machine problem, while iterative backward list scheduling algorithm tackles HFS
Lin & Chen (2015)	Makespan; Mean flow time	Unrelated	Processing times	Simulation optimization approach, integrating model evaluation, genetic algorithm optimization, and optimal computing budget allocation
Tang et al. (2015)	Energy uptake; makespan	Unrelated	Breakdowns; dynamic arrival	A particle swarm optimization algorithm based on Hill function to provide Pareto frontier of makespan and energy consumption.
Wang et al. (2015)	Makespan; Makespan deviation	Identical	Processing times	Order-based estimation of distribution algorithm with computer budget allocation
Zandieh & Hashemi (2015)	Expected value of makespan	Unrelated	Breakdowns	Simulation with genetic algorithm
González-Neira et al. (2016)	Weighted tardiness costs and satisfaction of customers	Identical	Processing times	Integral analysis method that encompasses both quantitative and qualitative analyses. The quantitative analysis involves GRASP with Monte Carlo simulation, while the qualitative analysis employs stochastic multicriteria acceptability analysis.
Ji et al. (2016)	Makespan	Identical	Processing times Setup times	Hybridization of particle swarm optimization and simulated annealing
Qin et al. (2018)	Makespan	Unrelated	Processing times	Ant colony algorithm based rescheduling approach
Azadeh et al. (2018)	Tardiness	Identical	Processing times; Set up times	Hybridization of artificial neural network, genetic algorithm and computer simulation
Rooeinfar et al. (2019)	Makespan	Identical	Processing times	Computer simulation model with three widely used metaheuristic algorithms: genetic algorithm, simulated annealing, and particle swarm optimization
Fu et al. (2020)	Makespan; Tardiness	Identical	Processing times	Hybrid multi-objective optimization algorithm that manages two populations, conducting global search across the entire solution space and local search within promising regions
Lin & Huang (2020)	Makespan	Unrelated	Machines capacity	New algorithm to obtain an estimated interval for network reliability
Han et al. (2021)	Makespan; Tardiness	Identical	Processing times	Seven multi-objective evolutionary algorithms with heuristic decoding
Wang & Xie (2021)	Makespan	Unrelated	Processing time	Artificial bee colony algorithm
Liu et al. (2023)	Makespan	Unrelated	Processing times, due dates	Reinforcement learning-based simulation-optimization within a genetic algorithm
Huang et al. (2023)	Makespan and total cost	Unrelated	processing time, demand, due date, unit production cost, unit holding cost, unit external production cost, and unit delayed completion cost	Pointer-based discrete differential evolution and two-stage stochastic programming, in the first stage the makespan and in the second stage total cost

3. Mixed integer linear programming model (MILP) for the deterministic FFS

In this section, the mathematical model of the deterministic version of the FFS, which minimizes total tardy jobs and makespan, is presented. Small instances will be solved using this model, addressing each objective function separately as two single objective independent models. The characteristics of these small instances are detailed in section 5.1, and the experiments conducted with these instances are discussed in subsections 5.3 and 5.4.

Sets:
$J: Jobs\ \{1, .., |J|\}$
$S: Stages\ \{1, .., |S|\}$
$I_s: Machines\ \{1, .., |I_s|\}, s \in S$

Parameters:
$p_{j,s,m}$: $Processing\ time\ for\ the\ job\ j \in J\ on\ machine\ m \in I_s\ of\ stage\ s \in S$
d_j: $Due\ date\ of\ the\ job\ j \in J$
M: $A\ very\ large\ number$

Decision variables:
$$X_{j,s,m}: \begin{cases} 1\ if\ the\ job\ j \in J\ is\ processed\ in\ the\ machine\ m \in I_s\ in\ the\ stage\ s \in S \\ 0\ otherwise \end{cases}$$
$ST_{j,s,m}$: $Starting\ time\ of\ the\ job\ j \in J\ in\ the\ machine\ m \in I_s\ in\ the\ stage\ s \in S$
$CT_{j,s,m}$: $Completion\ time\ of\ the\ job\ j \in J\ in\ the\ machine\ m \in I_s\ in\ the\ stage\ s \in S$
$Cmax$: $Makespan$
$$U_j: \begin{cases} 1\ if\ the\ job\ j \in J\ is\ delivered\ after\ the\ due\ date \\ 0\ otherwise \end{cases}$$
$$Y_{j,k,s}: \begin{cases} 1\ if\ the\ job\ j \in J\ is\ processed\ in\ the\ k-th\ position\ in\ the\ stage\ s \in S \\ 0\ otherwise \end{cases}$$

Objective functions:

$$min\ Z_1 = \sum_{j \in J} U_j \tag{1}$$
$$min\ Z_2 = Cmax \tag{2}$$

subject to:

$$\sum_{m \in I_s} X_{j,s,m} = 1 \quad \forall\, j \in J, \forall\, s \in S \tag{3}$$

$$\sum_{k \in J} Y_{j,k,s} = 1 \quad \forall\, j \in J, \forall\, s \in S \tag{4}$$

$$\sum_{j \in J} Y_{j,k,s} = 1 \quad \forall\, k \in J, \forall\, s \in S \tag{5}$$

$$CT_{j,s,m} = ST_{j,s,m} + (p_{j,s,m} \cdot X_{j,s,m}) \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{6}$$

$$ST_{j,s,m} \geq \sum_{n \in I_{s-1}} CT_{j,s-1,n} - M \cdot \left(1 - X_{j,s,m}\right) \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s, s > 1 \tag{7}$$

$$ST_{j,s,m} \geq CT_{i,s,m} - M \cdot \left(4 - X_{j,s,m} - X_{i,s,m} - Y_{j,k,s} - \sum_{n \in J, n < k} Y_{i,n,s}\right) \quad \forall j \in J, \forall k \in J, \forall i \in J, \forall s \in S, \forall m \in I_s \tag{8}$$

$$\sum_{m \in I_s} ST_{j,s,m} \geq \sum_{m \in I_s} ST_{i,s,m} - M \cdot (2 - Y_{j,k,s} - \sum_{n \in J, n < k} Y_{i,n,s}) \quad \forall\, j \in J, \forall\, k \in J, \forall\, i \in J, \forall\, s \in S \tag{9}$$

$$ST_{j,s,m} \leq X_{j,s,m} \cdot M \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{10}$$

$$CT_{j,s,m} \leq X_{j,s,m} \cdot M \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{11}$$

$$\sum_{m \in I_{|S|}} CT_{j,|S|,m} \leq d_j + G \cdot U_j \quad \forall\, j \in J \tag{12}$$

$$CT_{j,|S|,m} \leq Cmax \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{13}$$

$$ST_{j,s,m} \geq 0 \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{14}$$

$$CT_{j,s,m} \geq 0 \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{15}$$

$$X_{j,s,m} \in \{0,1\} \quad \forall\, j \in J, \forall\, s \in S, \forall\, m \in I_s \tag{16}$$

$$Y_{j,k,s} \in \{0,1\} \quad \forall\, j \in J, \forall\, k \in J, \forall\, s \in S \tag{17}$$

$$U_j \in \{0,1\} \quad \forall\, j \in J \tag{18}$$

Eq. (1) represents the objective function that minimizes the number of tardy jobs, while Eq. (2) is the objective function that optimizes makespan. Constraint set (3) ensures that a job is processed only once at each stage. Constraint sets (4) and (5) guarantee that there is only one position for each job and one job in each position, respectively. Eq. (6) calculates the completion time for a job based on the sum of its starting time and its processing time, as long as this job is processed on this machine. Constraint set (7) ensures that the starting time of a job in a stage must be greater than or equal to the completion time of the same job in the previous stage. Eq. (8) and Eq. (9) calculate the starting time of a job, making it greater or equal than the starting time of job i that is in a lower position in the sequence than job j and greater or equal than the completion time of the same job j in the previous stage. Constraint sets (10) and (11) ensure that the starting and completion times of a job in a specific machine of a stage only take values different from zero when the corresponding binary variables of assignment take the value of one. Constraint set (12) evaluates if a job is delivered after its due date, defining it as a tardy job. Constraint set (13) evaluates the makespan. Finally, constraint sets (14), (15), (16), (17), and (18) refer to the domain of decision variables.

4. Proposed NSGA-II simheuristic

According to Minella et al. (2011), a metaheuristic providing the Pareto frontier is the non-dominated elitist classification genetic algorithm II (NSGA-II), allowing a balance between multiple objectives. It achieves better dispersion and convergence. Implementing NSGA-II requires defining chromosome structure, generating the initial population, and processes of parent selection, crossover, and mutation (Fig. 1). The simheuristic has a 90-minute runtime limit for each instance. Due to stochastic breakdowns, a Monte Carlo simulation will calculate objective functions across generations, explained in subsection 4.6.

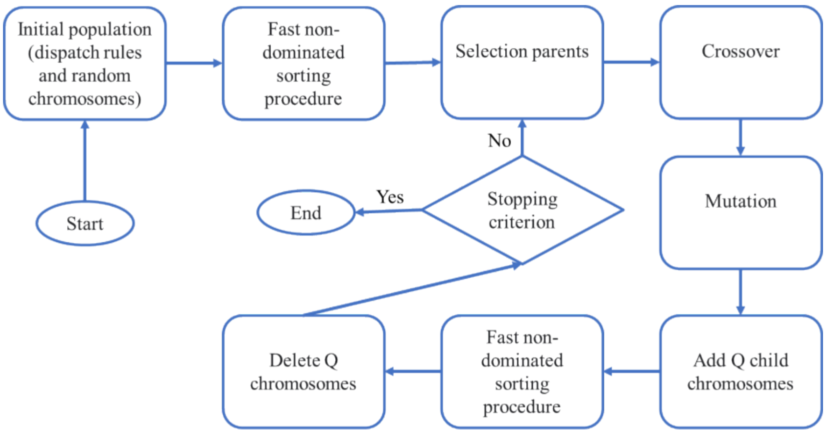


Fig. 1. Proposed simheuristic flowchart

4.1. Chromosome

For an FFS, establishing a chromosome requires collecting information for the processing sequence and machine assignment of jobs at each stage. Following the proposal of Schulz (2019), a matrix chromosome of size $|C| \cdot |J|$ is implemented for this project. Each matrix element is a positive rational number, where the integer component indicates machine $m \in I_s$ for processing job $j \in J$ at stage $s \in S$. The decimal component represents the job allocation sequence on machine m in stage c. Smaller decimal values prioritize job processing. If two jobs share the same decimal component, assignment is based on job number. Each column represents a job.

In Fig. 2, with 4 jobs in 3 stages, each having 2 parallel machines, the first stage processes job 1 and job 4 on machine 1, and jobs 2 and 3 on machine 2. The sequence for machine 1 in stage 1 is 4-1 due to the smaller decimal component of job 4. Similarly, for machine 2, the sequence in stage 1 is 2-3 as the decimal component of job 2 is smaller than that of job 3.

	Job 1	Job 2	Job 3	Job 4
Stage 1	1,76	2,34	2,68	1,27
Stage 2	1,13	2,79	1,25	2,21
Stage 3	2,30	2,73	2,09	1,47

Fig. 2. Chromosome representation

4.2. Initialization

To establish the initial population, chromosomes will be randomly generated under specified parameters. For improved solutions, seven chromosomes in the initial population are generated using seven dispatching rules, focusing on minimizing late deliveries and operating times in a deterministic case (without breakdowns). The implemented dispatching rules include Critical Ratio (CR), Earliest Due Date, Average Processing Time per Operation, Shortest Processing Time, NEH, NEH with Due Date, Apparent Tardiness Cost. The detailed implementations of these dispatching rules in the HFS are explained as follows:

- Critical Ratio (CR): Jobs are scheduled at each stage in ascending order of $\frac{d_j-\tau}{\sum_{h=s}^{|S|} p_{j,h,*}}$. This ratio indicates the proportion between remaining time before expiration and total time of remaining stages. * is associated with the machine for processing.
- Earliest Due Date (EDD): Jobs are scheduled in ascending order of due dates. Machine allocation is determined by the algorithm.
- Average Processing Time per Operation (AVPRO): Jobs are scheduled at each stage in ascending order of $\frac{\sum_{t=h}^{|S|} p_{j,h,*}}{|S|-s}$. This ratio corresponds to the average remaining time. * is associated with the machine for processing.
- Shortest Processing Time (SPT): Jobs are scheduled at each stage in ascending order of processing time $p_{j,s,m}$. Each job is assigned to the machine in that stage with the least processing time.
- NEH: Initial sequence of jobs is assigned in ascending order of processing time in missing stages, that is in ascending order of $\sum_{t=c}^{|C|} p_{j,t,*}$. * is associated with the machine for processing.
  To build the final sequence, the best partial sequence for the first two jobs is defined according to the best makespan. The third job is inserted in all possible positions of the previous partial sequence to define the partial sequence with the best makespan. This process continues for the remaining jobs to obtain the final sequence. The allocation of machines is determined by the algorithm.

- NEH with Due Date (NEHedd): Initial sequence of jobs is assigned in ascending order of due dates. The final sequence is built based on the best tardiness. The allocation of machines is determined by the algorithm.
- Apparent Tardiness Cost (ATC): Jobs are scheduled at each stage in ascending order of
  $\min Z_j = \frac{1}{\sum_{t=c}^{|C|} p_{j,t,*}} * e^{-\frac{T_j}{k \cdot \bar{p}}}$. * is associated with the machine for processing.

4.3. Order and parents' selection

After obtaining the initial population of N chromosomes, they are ordered according to the Fast Non-Dominated Sorting (FNS) procedure. FNS classifies chromosomes into different Pareto frontiers based on non-dominance conditions. Initially, chromosomes not dominated by any others form Pareto Frontier 1 (F_1). Then, non-dominated solutions from the subset $N \backslash F_1$ construct Pareto Frontier 2 (F_2), and so on. This process continues until all chromosomes are classified into a Pareto frontier. With the Pareto frontiers defined, the order of chromosomes within each frontier is established using crowding distance as a sorting criterion (refer to Pseudocode 1). Chromosomes at the extremes of the Pareto frontier have a pre-established crowding distance value close to infinity, reflecting their high diversification capacity for future generations. Other chromosomes are sorted in descending order of crowding distances, placing solutions with greater potential for optimal positioning in the later positions.

Once the chromosomes are sorted, parents for each generation are determined by selecting pairs with the help of random numbers. This process defines the pairs of chromosomes (parents) to be crossed by the entire population, generating Q children. It is important to note that the number of pairs of chromosomes is defined as the $Number\_of\_couples = Q/2$, as each couple generates 2 children (as shown in subsection 4.4).

Now, a population of size $2N$ is formed by parents P and children Q, where $|P| = N$ and $|Q| = N$. This combined population $P + Q$ is sorted using the FNS, and the first N ordered chromosomes constitute the population for the next generation.

Pseudocode 1. Order in each Pareto frontier

Begin /*Crowding distance*/
Initialize the parameters: Pareto_frontier_size(k), Crowding_dist(i), Order_Pareto_fon-tier($Pareto\_frontier\_size(k)$) For each Pareto frontier
Crowding_dist(Order_Pareto_fontier(1))=Infinite
Crowding_dist(Order_Pareto_fontier($Pareto\_frontier\_size(k)$))=Infinite
For each Objective function f
For each chromosome
If choromosome i is part of the pareto frontier k
For w=2 to $Pareto\_frontier\_size(k) - 1$
Crowding_dist(Order_Pareto_fontier(w))=Crowding_dist(Order_Pareto_fontier(w))+FO($i-1$, f)-FO($i+1$, f)
Next w End if
Next chromosome Next
f
Next Pareto frontier Return
Crowding_dist(i)
End

4.4. Crossover

According to the established procedure in Schulz (2019), crossover involves randomly selecting two parents. Each parent produces two offspring through recombination, wherein: i) a random number β_s is generated for each stage ($\beta_s \in \{0, \dots, |J|\}$) to determine the amount of information transferred from parent 1 to child 1 for the first β_s jobs at stage s. The missing information in child 1 is then filled with data from parent 2 in the same order. Similarly, child 2 is formed, with the first β_s jobs at stage s originating from parent 2, and the remaining information from parent 1. An illustrative example of the crossover is depicted in Figure 3, where $\beta_s = \{3, 1, 2\}$. This indicates that, in the first stage, information from the first 3 jobs of parent 1 will be assigned to child 1, and the information from job 4 will be assigned to child 2. Conversely, the first 3 jobs from parent 2 will be used for child 2, while the information from job 4 will be assigned to child 1. The same principle applies to stages 2 and 3 based on their respective values of β_s.

s	βs
1	3
2	1
3	2

Parent 1	Job 1	Job 2	Job 3	Job 4
Stage 1	1,76	2,34	2,68	1,27
Stage 2	1,13	2,79	1,25	2,21
Stage 3	2,30	2,73	2,09	1,47

Parent 2	Job 1	Job 2	Job 3	Job 4
Stage 1	1,44	1,55	2,27	2,82
Stage 2	1,80	2,13	2,55	1,27
Stage 3	2,65	2,14	1,58	1,93

Child 1	Job 1	Job 2	Job 3	Job 4
Stage 1	1,76	2,34	2,68	2,82
Stage 2	1,13	2,13	2,55	1,27
Stage 3	2,30	2,73	1,58	1,93

Child 2	Job 1	Job 2	Job 3	Job 4
Stage 1	1,44	1,55	2,27	1,27
Stage 2	1,80	2,79	1,25	2,21
Stage 3	2,65	2,14	2,09	1,47

Fig. 3. Example of crossover

4.5. Mutation

Once the chromosomes of the two children are obtained, a mutation will be performed to exchange the values between two jobs. The mutation is initiated by a random number, which is then compared with a mutation probability PM. If the random number is greater than or equal to PM, the chromosome will be modified; otherwise, it will remain unchanged after the crossover. The mutation involves using another random number to select two jobs, and then these jobs will exchange machines and assignments across each of their stages, as illustrated in Fig. 4.

Child 2	Job 1	Job 2	Job 3	Job 4
Stage 1	1,44	1,55	2,27	1,27
Stage 2	1,80	2,79	1,25	2,21
Stage 3	2,65	2,14	2,09	1,47

Child 2	Job 1	Job 2	Job 3	Job 4
Stage 1	2,27	1,55	1,44	2,82
Stage 2	1,25	2,13	1,80	1,27
Stage 3	2,09	2,14	2,65	1,93

Fig. 4. Example of mutation

4.6. *Monte Carlo simulation*

From subsection 4.1 to 4.5, all the elements of the proposed solution approach involve the NSGA-II metaheuristic, which allows solving the deterministic version of the multi-objective FFS. Now, the hybridization of Monte Carlo simulation with the NSGA-II metaheuristic is explained to transform it into the proposed NSGA-II simheuristic for handling stochastic multi-objective FFS.

For the design of the simheuristic that deals with stochastic machine breakdowns, it is necessary to establish two variables: time between failures (TBF) and time to repair (TTR). For this project, an exponential distribution was selected to model both TBFs and TTRs. The expected values will be named mean time between failures (MTBF) and mean time to repair (MTTR), respectively.

Based on what was proposed by Holthaus (1999), three different values are established for the MTTR, corresponding to $0.1\bar{p}$, $\bar{p}$, and $5\bar{p}$. Here, $\bar{p}$ is defined as the average processing time for each job at each machine. Throughout the study, these values that multiply the average processing times to provide the MTTR, i.e., {0.1, 1, and 5}, will be referred to as the Coefficient of Processing Times (CPT). On the other hand, to define the MTBF value, Eq. (19) proposed by Holthaus (1999) is implemented. Different Ag values will be established to represent the percentage of time that the machine is broken. Thus, from Equation (19), it is possible to solve for MTBF in terms of Ag, as shown in Eq. (20). Three equidistant values are established for Ag, corresponding to 0.03, 0.09, and 0.15.

For each solution found in the NSGA-II for the simheuristic, 100 replicates will be executed to evaluate the stochastic objective functions. Once the stop criterion is met, these objective functions will be recalculated based on a large Monte Carlo simulation comprising 1000 replicates.

$$Ag = \frac{MTTR}{MTBF + MTTR} \tag{19}$$

$$MTBF = MTTR \cdot \left(\frac{1}{Ag} - 1\right) \tag{20}$$

5. Computational experiments

This section presents all computational evaluations conducted to test the proposed approach and is divided into seven parts. Subsection 5.1 presents the small and large instances used to evaluate the simheuristic. The parametrization of the simheuristic is developed in subsection 5.2. Subsection 5.3 evaluates the performance of NSGA-II for the deterministic version of the problem in comparison with mathematical model results for each objective function, using small instances. The performance of the simheuristic, in comparison with the simulation of the solution obtained with the mathematical model for small instances, is presented in subsection 5.4. Subsection 5.5 evaluates the simheuristic results for each objective function. Subsection 5.6 evaluates the simheuristic in comparison with the simulation of the solution given by different dynamic dispatching rules. Finally, subsection 5.7 evaluates the simheuristic results for each performance measure.

Al experiments of this section were run in an Intel processor Core i5-6200U CPU 2.30 GHz 6th Gen, with a RAM of 8 Gb. The MILP models were implemented in GLPK and the NSGA-II was programmed in Java compiled by NetBeans.

5.1. *Instances*

For the evaluation of the simheuristic in comparison to the MILP model (subsections 5.3 and 5.4), a set of 35 small instances was generated following the same characteristics mentioned in Urlings et al. (2010). These small instances comprise the following combinations of jobs, stages, and machines per stage: {(3, 2, 2), (3, 3, 2), (4, 2, 2), (4, 2, 3), (4, 3, 2), (5, 2, 2), (5, 2, 3)}.

For the evaluation of the simheuristic in comparison to simulated solutions of dispatching rules (subsection 5.6) and in terms of the quality of the Pareto frontier (subsection 5.7), a total of 250 instances were used. Within these, 35 are the same small instances mentioned earlier, and the other 215 instances, of medium and large sizes, were randomly selected from the benchmark of Urlings et al. (2010). These 215 benchmark instances comprise the following combinations of jobs, stages, and the number of machines per stage: {(5, 3, 3), (7, 2, 3), (7, 3, 3), (9, 2, 3), (9, 3, 3), (11, 2, 3), (11, 3, 3), (13, 2, 3), (13, 3, 3), (15, 2, 3), (15, 3, 3), (50, 4, 2), (50, 4, 4), (50, 8, 2), (50, 8, 4), (100, 4, 2), (100, 4, 4), (100, 8, 2), (100, 8, 4)}. Table 2 shows the quantity of instances analyzed for each size.

Table 2
Quantity of instances analyzed for each size.

Instance size	Quantity of generated small instances	Quantity of medium and large size instances taken from benchmark of Urlings et al. (2010)
3_2_2	5	
3_3_2	5	
4_2_2	5	
4_2_3	5	
4_3_2	5	
5_2_2	5	
5_2_3	5	
5_3_3		5
7_2_3		5
7_3_3		5
9_2_3		5
9_3_3		5
11_2_3		5
11_3_3		5
13_2_3		5
13_3_3		5
15_2_3		5
15_3_3		5
50_4_2		20
50_4_4		20
50_8_2		20
50_8_4		20
100_4_2		20
100_4_4		20
100_8_2		20
100_8_4		20

5.2. Parametrization of NSGA-II simheuristic

To define the parameters of the simheuristic, a design of experiments was implemented through a non-parametric ANOVA. The response variable used was the mean modified ideal distance (MMID) as shown in Eq. (21), (22), (23), and (24), proposed by Ahmadi et al. (2016). The MMID measure represents the distance of the solutions in the Pareto frontier with respect to an ideal point. Note that the index i represents a solution of the Pareto frontier, TJ_i corresponds to the total tardy jobs of solution i, and $sdTJ_i$ is the standard deviation of tardy jobs of solution i.

$$MMID = \frac{\sum_{i=1}^{n} \sqrt{X_i^2 + Y_i^2 + W_i^2}}{n} \tag{21}$$

$$X_i = \frac{TJ_i - \min TJ}{\max TJ - \min TJ} \tag{22}$$

$$Y_i = \frac{Cmax_i - \min_i Cmax_i}{\max_i Cmax_i - \min_i Cmax_i} \tag{23}$$

$$W_i = \frac{sdTJ_i - \min_{\mathrm{i}} sdTJ_i}{\max_{\mathrm{i}} sdTJ_i - \min_{\mathrm{i}} sdTJ_i} \tag{24}$$

A design of experiments with six factors was carried out to parameterize the metaheuristic. Four factors corresponded to the parameters of the metaheuristic: the number of generations, the number of chromosomes, the probability of mutation, and the probability of crossover. The fifth factor was variability, which refers to the combination of parameters of the exponential distributions for the MTTR and MTBF. The sixth factor consisted of instances with ten levels, corresponding to ten instances selected at random from the set mentioned in subsection 5.1. Table 3 presents the tested levels for all factors, excluding the instance factor, along with their corresponding analyzed levels.

Table 3
Factor and levels for parametrization of simheuristic

Factor	Levels
Variability (combination of Ag and MTTR)	{Ag =0.03 with MTTR=0.1p, Ag =0.15 with MTTR=5p}
Number of generations	{400, 600}
Number of chromosomes	{800, 900}
Probability of mutation (PM)	{0.1, 0.15}
Probability of crossover (PC)	{0.72, 0.8}

Results of the non-parametric ANOVA are presented in Table 4, indicating, under a significance level of 10%, the factors or interactions that have a significant effect on the MMID. After analyzing the interval rankings provided by the non-parametric ANOVA, the combination of metaheuristic parameters that yielded the best statistical results corresponded to 800 chromosomes, 400 generations, a mutation probability of 0.1, and a crossover probability of 0.8.

Table 4
Significant factors and interactions under 10% of confidence

Factors	P-value	Factors	P-value
Generations	0.073	Instance:Chromosomes:PC	0.008
PM:PC	0.071	Chromosomes:PM:PC	0.088
Instance:Variability:PC	0.038	Generations:Chromosomes:PM:PC	0.068

5.3. Performance of NSGA-II metaheuristic vs MILP model (deterministic case)

To compare the performance of the NSGA-II metaheuristic, it is contrasted with the results produced by the MILP model after 90 minutes of execution. The chosen performance measure for this comparison is the GAP, calculated independently for each objective function according to Eq. (33). The value of each objective function for the NSGA-II was taken from the best extreme solution for each objective. Additionally, the MILP model was executed separately for each objective function, enabling the independent comparison of each objective function. A negative GAP indicates that the metaheuristic achieved better results than the MILP model within the 90-minute running time.

$$GAP = \frac{ObjectiveFunctionNSGAII - ObjectiveFunctionMILP}{ObjectiveFunctionMILP} \cdot 100\% \tag{25}$$

Table 5
Tardy jobs obtained NSGA-II metaheuristic vs MILP model (Tardy Jobs)

Instance	Results for the best values of tardy jobs and makespan among all the solutions in Pareto frontier obtained by NSGA-II metaheuristic		MILP Tardy Jobs	GAP Tardy Jobs	MILP Makespan	GAP Makespan
	Tardy Jobs	Makespan				
3_2_2_1	3	149	3	0.00%	149	0.00%
3_2_2_2	2	125	2	0.00%	125	0.00%
3_2_2_3	0	103	0	0.00%	103	0.00%
3_2_2_4	2	103	2	0.00%	103	0.00%
3_2_2_5	2	63	2	0.00%	63	0.00%
3_3_2_1	2	195	2	0.00%	195	0.00%
3_3_2_2	3	151	3	0.00%	151	0.00%
3_3_2_3	2	169	2	0.00%	169	0.00%
3_3_2_4	2	162	2	0.00%	162	0.00%
3_3_2_5	2	162	2	0.00%	162	0.00%
4_2_2_1	3	181	3	0.00%	181	0.00%
4_2_2_2	3	103	3	0.00%	103	0.00%
4_2_2_3	1	137	1	0.00%	137	0.00%
4_2_2_4	2	137	2	0.00%	137	0.00%
4_2_2_5	4	82	4	0.00%	82	0.00%
4_2_3_1	2	81	2	0.00%	81	0.00%
4_2_3_2	1	81	1	0.00%	81	0.00%
4_2_3_3	1	68	1	0.00%	68	0.00%
4_2_3_4	0	67	0	0.00%	67	0.00%
4_2_3_5	1	69	1	0.00%	69	0.00%
4_3_2_1	2	107	2	0.00%	107	0.00%
4_3_2_2	3	144	3	0.00%	144	0.00%
4_3_2_3	2	132	2	0.00%	132	0.00%
4_3_2_4	2	161	2	0.00%	161	0.00%
4_3_2_5	3	107	3	0.00%	107	0.00%
5_2_2_1	1	123	1	0.00%	123	0.00%
5_2_2_2	3	150	3	0.00%	151	-0.66%
5_2_2_3	1	132	1	0.00%	132	0.00%
5_2_2_4	3	132	3	0.00%	132	0.00%
5_2_2_5	3	129	3	0.00%	129	0.00%
5_2_3_1	2	90	2	0.00%	90	0.00%
5_2_3_2	1	92	1	0.00%	92	0.00%
5_2_3_3	1	84	1	0.00%	84	0.00%
5_2_3_4	1	126	1	0.00%	149	-15.44%
5_2_3_5	3	126	3	0.00%	149	-15.44%

Table 5 presents the results of the metaheuristic's performance compared to the MILP model that minimizes tardy jobs and the one that minimizes makespan. It is important to note that when dealing with tardy jobs, the optimal solution may be zero, resulting in a division by zero in the GAP equation. To address this, we avoid the division by zero by recognizing that in instances where a zero best solution was identified, NSGA-II also attained this optimal result. Consequently, the GAP will be zero in these specific instances. In the case of the model that minimizes tardy jobs, the results reveal an average

GAP for tardy jobs of 0%. The percentage of instances that reached the optimum or improved upon the solution given by the MILP model is 100.00%.

The results of the metaheuristic's performance compared to the MILP model optimizing makespan show an average GAP of -0.90%. It can be noted that 100.00% of instances either reached the optimum or improved upon the solution given by the MILP model. This second situation happens because for three instances the MILP model did not obtain the optimal solution in 90 minutes but obtained a feasible one.

5.4. Performance of NSGA-II simheuristic vs simulation of solution provided by MILP model

The solution provided by the MILP model that minimizes tardy jobs and the solution provided by the MILP model when minimizing makespan, for each instance, are subjected to a Monte Carlo simulation of 1000 replicates to obtain their expected tardy jobs, expected makespan, and standard deviation of tardy jobs. These results are then compared with the results obtained with the same instance in the proposed simheuristic.

Table 6
Results of simheuristic vs. simulation of solutions of MILP model that minimizes tardy jobs and MILP model that minimizes makespan

Instances	Simheuristic results for the best value of each objective function among all solutions in Pareto frontier			Simulation results of MILP model that minimizes tardy jobs			GAP with respect to simulated solution of MILP model that minimizes tardy jobs			Simulation results of MILP model that minimizes makespan			GAP with respect to simulated solution of MILP model that minimizes makespan		
	Average of expected tardy jobs	Average of expected makespan	Standard deviation of tardy jobs	Average of expected tardy jobs	Average of expected makespan	Standard deviation of tardy jobs	Expected tardy jobs	Expected makespan	Standard deviation of tardy jobs	Average of expected tardy jobs	Average of expected makespan	Standard deviation of tardy jobs	Expected tardy jobs	Expected makespan	Standard deviation of tardy jobs
3_2_2_1	3.00000	160.81097	0.00000	3.00000	307.26101	0.00000	0.00%	-47.71%	0.00%	3.00000	164.21005	0.00000	0.00%	-1.89%	0.00%
3_2_2_2	2.05100	137.14553	0.00000	2.07289	208.92788	0.22103	-1.04%	-34.44%	-100.00%	3.00000	140.34906	0.00000	-31.63%	-2.13%	0.00%
3_2_2_3	0.08433	111.88054	0.10292	1.17911	158.65667	0.35064	-93.58%	-29.56%	-74.44%	3.00000	246.40997	0.00000	-97.19%	-54.65%	0.00%
3_2_2_4	2.03550	111.91415	0.00000	2.05767	172.26395	0.17940	-1.05%	-35.10%	-77.78%	3.00000	246.08305	0.00000	-32.15%	-54.58%	0.00%
3_2_2_5	2.01106	70.63878	0.08154	2.08678	87.52244	0.21209	-3.48%	-19.40%	-39.99%	2.04700	72.35208	0.16092	-1.71%	-2.21%	-30.68%
3_3_2_1	2.05594	211.17635	0.00000	2.07578	358.51194	0.21005	-0.92%	-41.16%	-100.00%	3.00000	217.39250	0.00000	-31.47%	-2.66%	0.00%
3_3_2_2	3.00000	165.36253	0.00000	3.00000	263.70386	0.00000	0.00%	-37.37%	0.00%	3.00000	168.10835	0.00000	0.00%	-1.52%	0.00%
3_3_2_3	2.08783	182.35982	0.00000	2.12033	347.35270	0.30068	-1.52%	-47.55%	-100.00%	3.00000	187.88878	0.00000	-30.41%	-2.75%	0.00%
3_3_2_4	2.01206	172.05215	0.00000	2.02078	227.39629	0.10773	-0.43%	-24.35%	-66.67%	3.00000	178.17063	0.00000	-32.93%	-3.24%	0.00%
3_3_2_5	2.01817	172.07849	0.00000	2.02800	259.66680	0.12630	-0.48%	-33.73%	-66.67%	3.00000	178.07882	0.00000	-32.73%	-3.17%	0.00%
4_2_2_1	3.02450	198.91479	0.05091	3.04322	361.00792	0.19007	-0.61%	-44.95%	-75.40%	3.04022	204.42917	0.18143	-0.52%	-2.55%	-71.26%
4_2_2_2	3.00722	110.57431	0.00676	3.03500	206.61109	0.14114	-0.90%	-46.48%	-63.89%	3.06956	114.07696	0.20569	-1.98%	-2.88%	-98.03%
4_2_2_3	1.13256	150.56586	0.07922	1.28944	213.89741	0.46812	-11.82%	-29.66%	-83.22%	3.01622	230.05604	0.09554	-62.47%	-34.60%	-3.70%
4_2_2_4	2.03233	150.33673	0.08455	2.05944	258.52942	0.18913	-1.28%	-41.90%	-55.77%	2.04800	154.07285	0.17322	-0.75%	-2.23%	-31.70%
4_2_2_5	4.00000	88.02969	0.00000	4.00000	227.76171	0.00000	0.00%	-61.38%	0.00%	4.00000	90.10430	0.00000	0.00%	-2.15%	0.00%
4_2_3_1	2.02828	89.64506	0.07473	2.03733	185.09693	0.17290	-0.44%	-51.65%	-46.24%	4.00000	205.01256	0.00000	-49.29%	-56.28%	0.00%
4_2_3_2	1.08883	89.63001	0.09166	1.57689	168.76869	0.55983	-27.97%	-46.95%	-84.40%	1.12378	91.96031	0.28660	-2.83%	-2.37%	-74.58%
4_2_3_3	1.03172	73.69060	0.06992	1.09333	89.43403	0.24911	-5.13%	-17.44%	-43.11%	2.10189	77.35397	0.23499	-50.85%	-4.26%	-52.93%
4_2_3_4	0.11033	74.14723	0.13048	0.16700	88.22752	0.38570	181.91%	-15.83%	-17.38%	0.19111	75.85814	0.40803	-47.73%	-2.14%	-57.44%
4_2_3_5	1.10756	75.30047	0.08329	1.13678	94.63031	0.34773	-2.46%	-20.46%	-69.25%	2.10344	78.08068	0.28128	-47.46%	-3.37%	-62.04%
4_3_2_1	2.11006	119.16284	0.00000	2.45867	168.85877	0.42654	-12.69%	-29.45%	-100.00%	4.00000	120.66685	0.00000	-47.25%	-1.19%	0.00%
4_3_2_2	3.03361	161.31049	0.02219	3.12144	259.84306	0.29421	-2.75%	-38.04%	-89.78%	3.05011	164.61756	0.16755	-0.53%	-1.87%	-63.20%
4_3_2_3	2.06956	144.88900	0.00000	2.13722	270.51410	0.34830	-3.00%	-46.51%	-100.00%	4.00000	149.83940	0.00000	-48.26%	-2.99%	0.00%
4_3_2_4	2.04317	174.39163	0.00000	2.11956	328.51039	0.28386	-3.43%	-46.99%	-88.89%	4.00000	178.93897	0.00000	-48.92%	-2.41%	0.00%
4_3_2_5	3.00672	118.24632	0.00000	3.01489	146.70295	0.09445	-0.27%	-19.39%	-66.67%	4.00000	121.39353	0.00000	-24.83%	-2.36%	0.00%
5_2_2_1	1.45167	135.66479	0.03836	5.00000	217.23678	0.00000	-70.97%	-37.66%	0.00%	3.27111	138.21741	0.49011	-56.08%	-1.73%	-89.60%
5_2_2_2	3.22111	163.97166	0.01648	3.33133	233.94633	0.46960	-3.22%	-29.89%	-93.08%	3.34256	168.01122	0.49217	-3.54%	-2.31%	-93.99%
5_2_2_3	1.26794	145.02958	0.14450	1.33011	163.91237	0.56274	-4.34%	-11.48%	-70.44%	2.27511	146.32610	0.42953	-44.81%	-0.86%	-56.16%
5_2_2_4	3.11178	144.68852	0.06999	3.14244	212.17634	0.33230	-0.95%	-31.84%	-69.97%	4.06911	146.58744	0.23048	-23.55%	-1.16%	-60.18%
5_2_2_5	3.14000	142.85161	0.02310	3.17733	190.76456	0.36813	-1.16%	-25.20%	-89.79%	4.14022	145.49255	0.32287	-24.18%	-1.67%	-87.93%
5_2_3_1	2.07722	101.86374	0.00424	2.32411	223.74899	0.53060	-10.17%	-54.61%	-99.06%	4.02522	105.23587	0.12256	-48.40%	-2.98%	-75.00%
5_2_3_2	1.06706	99.02297	0.07480	1.22911	136.95740	0.44181	-10.73%	-27.67%	-87.43%	2.16000	101.90528	0.33823	-50.51%	-2.69%	-79.40%
5_2_3_3	1.27200	93.17603	0.06016	1.56589	166.31495	0.75543	-17.66%	-44.06%	-92.11%	2.22678	97.50266	0.43629	-43.46%	-4.08%	-88.06%
5_2_3_4	1.36083	144.97102	0.05488	1.41789	241.32131	0.50076	-3.81%	-40.08%	-89.13%	1.41567	154.86092	0.52370	-3.77%	-6.44%	-89.98%
5_2_3_5	3.09889	144.19794	0.00000	3.22633	231.87067	0.42073	-3.87%	-37.97%	-100.00%	4.00689	154.63307	0.06176	-22.66%	-6.89%	-66.67%

Table 6 presents the results of the performance of the simheuristic compared to the simulation of the MILP model. In comparison to the simulation of solutions obtained by the MILP model that optimizes expected tardy jobs, the average GAPs for expected tardy jobs, expected makespan, and standard deviation of tardy jobs are -3.43%, -35.65%, and -68.59%, respectively. Concerning the simulations of solutions provided by the MILP model that minimizes expected makespan, the average GAPs of the simheuristic for expected tardy jobs, expected makespan, and standard deviation of tardy jobs are -29.85%, -8.09%, and -38.07%, respectively. These results demonstrate the importance of including stochasticity in the solution method to obtain solutions that better adapt to the uncertain environment.

5.5. *Simheuristic results for each objective function*

Three experimental designs, one for each objective function of the Pareto frontier, were conducted to analyze the influence of Ag and CPT on these objectives. Since the normality and homoscedasticity assumptions were not fulfilled, the non-parametric test called ANOVA-Type statistic (Brunner et al., 1997) was conducted for each of the three objective functions. Each one of the 250 instances was executed twice for this experiment. The factors and levels analyzed for each factor were: Ag {0.03, 0.09, 0.15}, CPT {10.1, 1, 0.5}, and instances with 250 levels. The results of the ANOVAs-Type statistic indicate that both Ag and CPT have significant effects on all three objective functions (see Table 7).

Table 7
P-values of ANOVA-Type statistic for each objective function

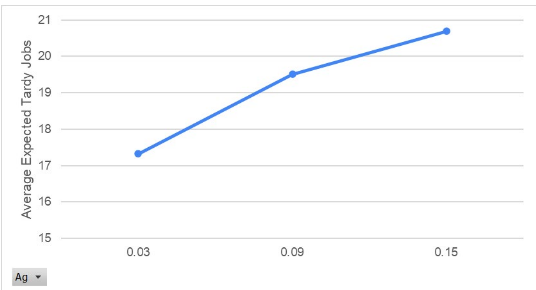
	p-values ANOVA-Type statistics		
Factor	Expected tardy jobs	Expected makespan	Standard deviation of tardy jobs
Ag	0.0000	0.0028	0.0000
CPT	0.0000	0.0000	0.0000
$Ag{:}CPT$	0.0020	0.0376	0.0000

The means plots provide more details about the results. On one hand, Fig. 5a, Fig. 5b, Fig. 5d, and Fig. 5e illustrate that the expected tardy jobs and expected makespan are directly proportional to the values of Ag, and CPT. This suggests that achieving lower values for tardy jobs and makespan is associated with effective management of machine breakdowns. On the other hand, concerning the standard deviation of tardy jobs, Fig. 5g depicts that the standard deviation of tardy jobs decreases as Ag increases, but when CPT increases, Fig. 5h displays that the standard deviation of tardy jobs also increases. Therefore, it is important to analyze the interaction between Ag and CPT. Fig. 5c and Fig. 5f show that when CPT values are low (i.e. 0.1 and 1), the expected tardy jobs and makespan remain almost the same regardless of Ag, whereas when CPT is high (i.e. 5), the expected number of tardy jobs increases as Ag increases. Instead, Fig. 5i shows that when CPT value is high, the standard deviation of tardy jobs reduces for $Ag = 0.15$, whereas for lower values of CPT, the behavior of the standard deviation of tardy jobs is practically the same for all values of Ag. This implies that maintaining lower repair times is preferable for obtaining more stable schedules.

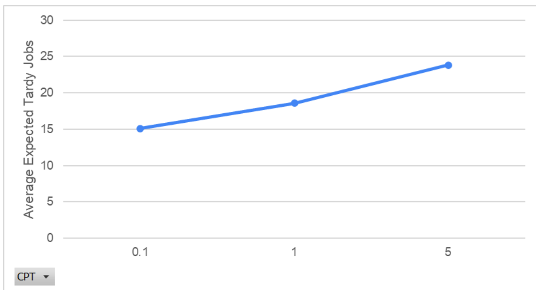
5.6. *Evaluation of simheuristic in comparison with the simulation of the solution given by different dispatching rules*

An experimental design, involving all benchmark instances mentioned in subsection 5.1, was conducted to determine whether there is an effect of Ag and CPT on the percentage of improvement in the three objective functions of the problem provided by the simheuristic, in comparison to the expected objectives obtained through the simulation of the solution given by dispatching rules mentioned in subsection 4.2. The percentage of improvement was calculated according to Equation (26). A positive result indicates that the simheuristic improves upon the dispatching rule. The results of non-parametric ANOVA confirm that Ag, CPT, and the interaction between Ag and CPT have a significant effect on the three percentages of improvement, with p-values < 0.01.

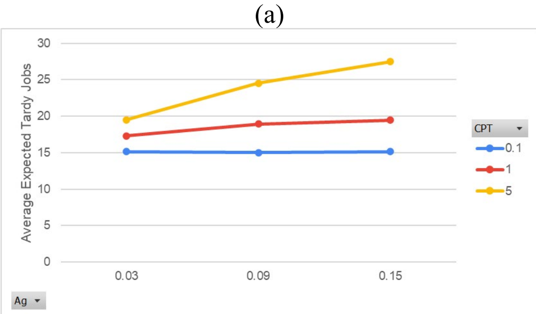
$$PercentajeOfImprovement = \frac{ObjectiveSimulatedDispatchingRule - ObjectiveSimheuristic}{ObjectiveSimulatedDispatchingRule} \quad (26)$$



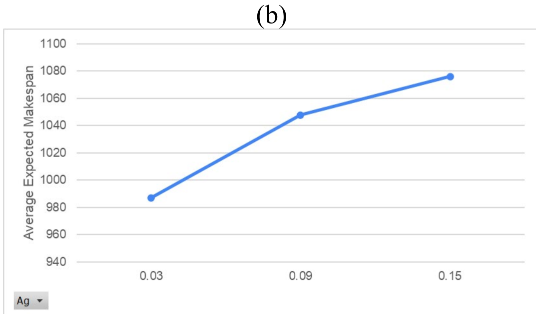
(a)



(b)



(c)



(d)

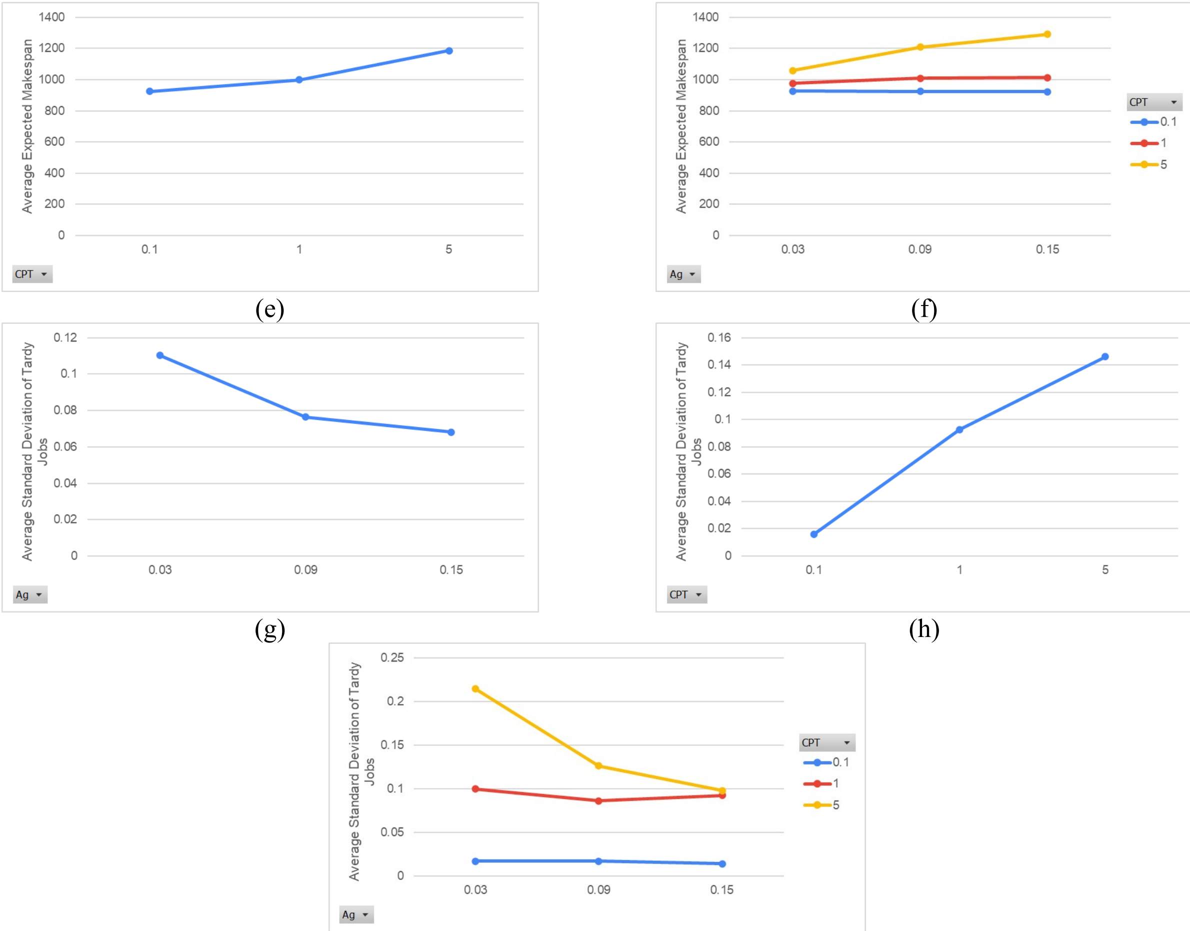


Fig. 5. Means plots of expected tardy jobs, expected makespan and standard deviation of tardy jobs for factor Ag, factor CPT and interaction $Ag - CPT$.

Fig. 6 presents the mean plot of the percentage of improvement achieved by the simheuristic in objective functions compared to dispatching rules. In the case of tardy jobs, this figure shows that the minimum improvement achieved by the simheuristic is in comparison to EDD, with a value of 21.26%. On the other hand, the maximum average improvement of the simheuristic is 56.37%, observed in comparison with the CR dispatching rule. Regarding the makespan, Fig. 6 reveals that the minimum improvement reached by the simheuristic is also in comparison to EDD, with an average of 7.53%, whereas the maximum improvement obtained was in comparison to the CR dispatching rule with an average of 62.56%. Lastly, with respect to the standard deviation of tardy jobs, Fig. 6 shows that the simheuristic gained the minimum improvement in comparison to EDD with a value of 75.81% and the maximum improvement in comparison to CR with a value of 99.53%. It is important to note that the simheuristic achieves the best improvements for the standard deviation of tardy jobs in comparison to all dispatching rules, demonstrating the importance of considering robustness measures to obtain more stable schedules.

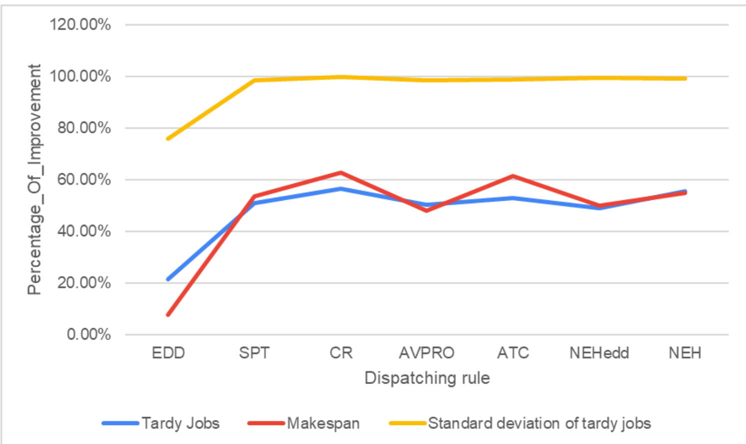


Fig. 6. Mean plots of percentage of improvement achieved by simheuristic in objective functions in comparison to dispatching rules.

5.7. Quality indicators of Pareto frontiers obtained by the proposed simheuristic

To the best of our knowledge, this is the only investigation that has explored a S FFS with machine breakdowns to derive the Pareto frontier of expected tardy jobs, expected makespan, and standard deviation of tardy jobs. We introduce four additional indicators, in addition to MMID (Eq. 21), tailored for the multi-objective problems:

- Diversity: As presented in Ahmadi et al. (2016), this criterion quantifies the Euclidean distance between the initial and final solutions within a Pareto frontier (Equation 27). Elevated diversity values indicate a higher quality of the Pareto frontier.

$$Diversity = \sqrt{\sum_{of=1}^{3} \left(\max Z_{of} - \min Z_{of}\right)^2} \tag{27}$$

- Spread: Another measure of diversity used by Behnamian et al. (2009), this indicator is calculated as presented in Eq. (28), Eq. (29), and EQ. (30).

$$Spread = \frac{\sqrt{\sum_{i=1}^{n}(MID - c_i)^2}}{n} \tag{28}$$

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{29}$$

$$c_i = \sqrt{{TJ_i}^2 + {Cmax_i}^2 + {sdTJ_i}^2} \tag{30}$$

- Number of solutions in the Pareto frontier: Also presented in Ahmadi et al. (2016). Increased values of this measure suggest a broader array of options for managers in decision-making scenarios, providing administrators with access to a greater number of alternative solutions.
- Execution time: It represents the time required to obtain the Pareto frontier with the proposed NSGA-II simheuristic.

Table 8 displays the averages of the five mentioned indicators for each instance size. It can be observed that diversity, spread, and the number of solutions are higher for larger instances. In contrast, MMID remains relatively consistent, independent of the instance size.

Additionally, an experimental design was conducted to evaluate the effects of Ag, CPT and their interaction in the five indicators. The factors and their levels are the same as those presented in subsection 5.5. Table 9 presents the significant results obtained through the implementation of the ANOVA-Type statistic, as the assumptions of normality and homoscedasticity of ANOVA were not fulfilled. According to the ANOVA-Type statistic tests, with a significance level of 5%, Ag, CPT, and the interaction between them have a significant effect on MMID, Diversity, Spread, and the number of solutions on the Pareto frontier. The p-values marked with an asterisk were the most significant, i.e., significant under the 0.001 significance level, which is the reason for presenting their mean plots in Figure 7.

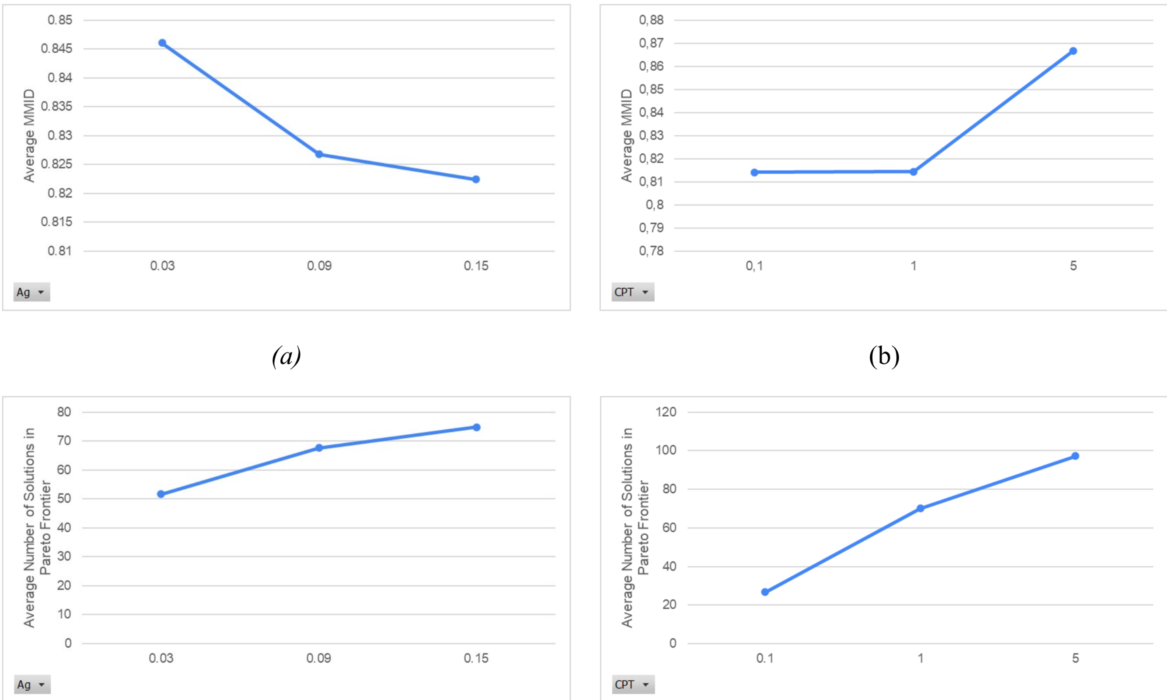
According to the mean plots in Figures 7a and 7b, it is evident that MMID decreases as the Ag values increase and exhibits higher values for $CPT = 5$. The number of solutions in the Pareto frontier, as shown in Figures 7c and 7d, is directly proportional to both Ag and CPT values. Additionally, Spread and Diversity exhibit a directly proportional behavior with respect to the CPT values, as shown in Figures 7e and 7f. Finally, concerning the interaction between Ag and CPT, Figure 7g demonstrates that the number of solutions on the Pareto Frontier remains almost the same for all values of Ag when CPT is 0.1 but increases as Ag increases when CPT values are 1 or 5.

Table 8
Quality indicator of Pareto frontier

Instance size	Average MMID	Average Diversity	Average Spread	Average number of solutions in Pareto frontier	Average execution time (s)
3_2_2	0.6905	7.5209	2.7995	5.2222	24.9263
3_3_2	0.8733	7.1524	2.5259	5.3333	29.9417
4_2_2	0.6961	6.2950	1.9614	7.6778	26.8031
4_2_3	0.9686	12.7989	3.7041	15.5111	29.1527
4_3_2	1.0344	18.1605	6.1721	8.9111	31.6320
5_2_2	1.0707	20.5656	6.9583	17.8000	27.0907
5_2_3	0.9953	16.7482	5.1807	35.9667	29.6200
5_3_3	0.9161	30.2668	7.4973	29.4778	39.5801
7_2_3	0.9223	25.3524	5.4872	47.9000	33.5877
7_3_3	0.9262	24.0636	6.8430	30.5556	43.8007
9_2_3	0.8689	35.7305	8.5203	42.9333	37.7579
9_3_3	0.8544	50.5517	11.4070	53.9444	50.5293
11_2_3	0.8551	52.6931	12.0970	53.6333	42.0295
11_3_3	0.8585	67.1794	13.8069	47.3556	56.9048
13_2_3	0.8361	49.9858	9.4986	59.2778	47.6317
13_3_3	0.8341	46.2460	9.3309	40.4444	63.8043
15_2_3	0.7917	57.7219	11.9997	56.0111	55.0823
15_3_3	0.8459	56.1219	10.7602	53.3111	71.7549
50_4_2	0.7773	507.0685	86.1907	77.9111	823.2424
50_4_4	0.7927	254.0033	41.3478	109.5028	498.0969
50_8_2	0.9324	1260.3963	275.6771	49.0139	1209.4426
50_8_4	0.8498	434.9944	69.5198	104.2194	1257.2189
100_4_2	0.7644	1126.6287	175.8791	83.1139	2201.6448
100_4_4	0.7272	451.0481	70.1320	109.1750	2239.5723
100_8_2	0.8578	2495.9539	589.2393	42.1111	4633.0520
100_8_4	0.7475	1509.1368	253.8968	80.5889	4591.7322
Total average	**0.8572**	**331.7071**	**65.3243**	**48.7270**	**699.8320**

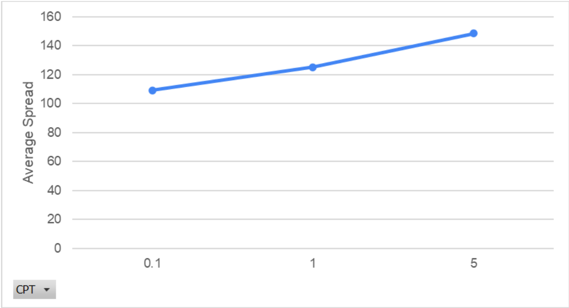
Table 9
P-values of ANOVA-Type statistic of factors Ag, CPT and the interaction for multi-objective performance measures

Factor	MMID	Diversity	Spread	Number of solutions	Running time
Ag	0.0007*	0.0015	0.0224	0.0000*	0.9072
CPT	0.0000*	0.0000*	0.0000*	0.0000*	0.5480
Ag:CPT	0.0433	0.0023	0.0118	0.0000*	0.3000

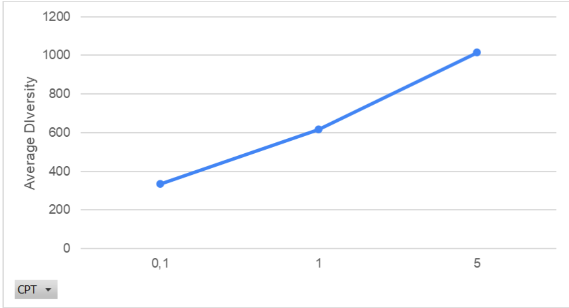


(a) (b)

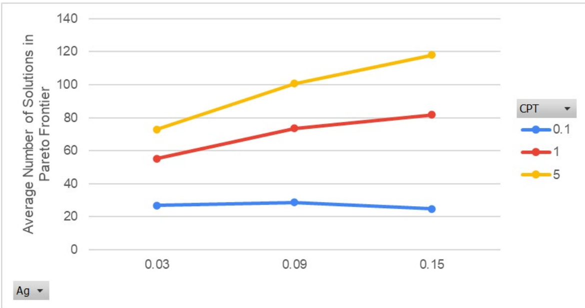
(c) (d)



(e)



(f)



(g)

Fig. 7. Means plots of Pareto frontier performance measures for factor Ag, factor CPT, and interaction $Ag - CPT$.

6. Conclusions and future work

The aim of this paper was to design a simheuristic that hybridizes an NSGA-II with Monte Carlo simulation to solve a multi-objective flexible flow shop problem subject to stochastic machine breakdowns. The objective functions analyzed were tardy jobs, makespan, and standard deviation of tardy jobs. The breakdowns were modeled with an exponential distribution for both times between failures and times to repair.

In the first stage, a MILP model was proposed to solve the deterministic version of the problem for tardy jobs and makespan separately. In the second phase, the proposed simheuristic was parameterized. In the third place, the NSGA-II metaheuristic (i.e., the proposed NSGA-II without the hybridization of Monte Carlo simulations) was evaluated for the deterministic version of the problem in comparison with the solutions obtained by the MILP model for each objective function independently, using small instances. The MILP model was executed with a time limit of 5400s. In the fourth place, to evaluate the quality of the Pareto frontiers given by the simheuristic, five different performance measures were selected: MMID, diversity, spread, the number of chromosomes in the last Pareto frontier, and execution time. Finally, the simheuristic was compared to the simulation of the solutions obtained with seven dispatching rules adapted to the problem.

Regarding the results of the metaheuristic for small instances in comparison to the MILP model, the metaheuristic always reaches the optimum when the model obtained the optimum solution. Additionally, the metaheuristic improves the objective function of the feasible solution obtained by the MILP model when, in 5400 seconds of execution, the model could not reach the optimum.

Once the solutions obtained in the MILP models were simulated, the NSGA-II simheuristic was compared to them, resulting in average GAPs of -16.64%, -21.87%, and -53.33% for expected tardy jobs, expected makespan, and standard deviation of tardy jobs, respectively. This implies that the simheuristic significantly improves upon the results of simulating optimal deterministic solutions. Moreover, the performance of the simheuristic was also evaluated against the results of simulating solutions provided by seven dispatching rules, showing improvements of 48.01%, 48.18%, and 95.63% for expected tardy jobs, expected makespan, and standard deviation of tardy jobs, respectively. These results suggest that designing a method involving stochasticity is better than implementing a deterministic method alone.

Additionally, the NSGA-II simheuristic was evaluated in terms of the quality of the Pareto frontier. For this evaluation, five multi-objective performance indexes were measured, confirming the quality of the proposed method.

For future studies, the implementation of new probability distributions for times between failures and times to repair is proposed. Likewise, it is important to suggest new values for the Ag and CPT parameters since, as observed in the non-parametric ANOVA, these are significant for most of the results obtained in the simheuristics. On the other hand, it is recommended for future studies to analyze other parameters under uncertainty, such as processing times, setup times, release times, due dates, among others.

References

Abu-Marrul, V., Martinelli, R., Hamacher, S., & Gribkovskaia, I. (2023). Simheuristic algorithm for a stochastic parallel machine scheduling problem with periodic re-planning assessment. *Annals of Operations Research*, *320*(2), 547–572. https://doi.org/10.1007/s10479-022-04534-5

Ahmadi, E., Zandieh, M., Farrokh, M., & Emami, S. M. (2016). A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Computers & Operations Research*, *73*, 56–66. https://doi.org/10.1016/j.cor.2016.03.009

Allahverdi, A., Aydilek, A., & Aydilek, H. (2016). Minimizing the number of tardy jobs on a two-stage assembly flowshop. *Journal of Industrial and Production Engineering*, *33*(6), 391–403. https://doi.org/10.1080/21681015.2016.1151466

Almeder, C., & Hartl, R. F. (2013). A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. *International Journal of Production Economics*, *145*(1), 88–95. https://doi.org/10.1016/j.ijpe.2012.09.014

Al-Turki, U. M., Saleh, H., Deyab, T., & Almoghathawi, Y. (2012). Resource Allocation and Job Dispatching for Unreliable Flexible Flow Shop Manufacturing System. *Advanced Materials Research*, *445*, 947–952. https://doi.org/10.4028/www.scientific.net/AMR.445.947

Azadeh, A., Goodarzi, A. H., Kolaee, M. H., & Jebreili, S. (2018). An efficient simulation–neural network–genetic algorithm for flexible flow shops with sequence-dependent setup times, job deterioration and learning effects. *Neural Computing and Applications*, *6*, 1–15. https://doi.org/10.1007/s00521-018-3368-6

Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2009). A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications*, *36*(8), 11057–11069. https://doi.org/10.1016/j.eswa.2009.02.080

Brunner, E., Dette, H., & Munk, A. (1997). Box-Type Approximations in Nonparametric Factorial Designs. *Journal of the American Statistical Association*, *92*(440), 1494–1502. https://doi.org/10.1080/01621459.1997.10473671

Caldeira, R. H., & Gnanavelbabu, A. (2021). A simheuristic approach for the flexible job shop scheduling problem with stochastic processing times. *SIMULATION*, *97*(3), 215–236. https://doi.org/10.1177/0037549720968891

Chen, C.-L., & Chen, C.-L. (2009). Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines. *Computers & Industrial Engineering*, *56*(4), 1393–1401. https://doi.org/10.1016/j.cie.2008.08.016

Choi, S. H., & Wang, K. (2012). Flexible flow shop scheduling with stochastic processing times: A decomposition-based approach. *Computers & Industrial Engineering*, *63*(2), 362–373. https://doi.org/10.1016/j.cie.2012.04.001

Das, K. (2008). A comparative study of exponential distribution vs Weibull distribution in machine reliability analysis in a CMS design. *Computers & Industrial Engineering*, *54*(1), 12–33. https://doi.org/10.1016/j.cie.2007.06.030

de León, A. D., Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2021). A simulation–optimization framework for enhancing robustness in bulk berth scheduling. *Engineering Applications of Artificial Intelligence*, *103*, 104276. https://doi.org/10.1016/j.engappai.2021.104276

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *CEUR Workshop Proceedings* (Vol. 1133, pp. 849–858). https://doi.org/10.1007/3-540-45356-3_83

Ebrahimi, M., Fatemi Ghomi, S. M. T., & Karimi, B. (2014). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling*, *38*(9–10), 2490–2504. https://doi.org/10.1016/j.apm.2013.10.061

Fu, Y., Zhou, M., Guo, X., & Qi, L. (2020). Scheduling Dual-Objective Stochastic Hybrid Flow Shop With Deteriorating Jobs via Bi-Population Evolutionary Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(12), 5037–5048. https://doi.org/10.1109/TSMC.2019.2907575

Gonzalez-Neira, E. M., Ferone, D., Hatami, S., & Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *79*, 23–36. https://doi.org/10.1016/j.simpat.2017.09.001

González-Neira, E. M., García-Cáceres, R. G., Caballero-Villalobos, J. P., Molina-Sánchez, L. P., & Montoya-Torres, J. R. (2016). Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria. *Computers & Industrial Engineering*, *101*, 128–144. https://doi.org/10.1016/j.cie.2016.08.026

González-Neira, E. M., Montoya-Torres, J. R., & Barrera, D. (2017). Flow-shop scheduling problem under uncertainties: Review and trends. *International Journal of Industrial Engineering Computations*, *8*(4), 399–426. https://doi.org/10.5267/j.ijiec.2017.2.001

Han, W., Deng, Q., Gong, G., Zhang, L., & Luo, Q. (2021). Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint. *Expert Systems with Applications*, *168*, 114282. https://doi.org/10.1016/j.eswa.2020.114282

Holthaus, O. (1999). Scheduling in job shops with machine breakdowns: An experimental study. *Computers and Industrial Engineering*, *36*(1), 137–162. https://doi.org/10.1016/S0360-8352(99)00006-6

Hsieh, J.-C., Chang, P.-C., & Hsu, L.-C. (2003). Scheduling of drilling operations in printed circuit board factory☆. *Computers & Industrial Engineering*, *44*(3), 461–473. https://doi.org/10.1016/S0360-8352(02)00231-0

Huang, Y., Deng, L., Wang, J., Qiu, W., & Liu, J. (2023). Modeling and solution for hybrid flow-shop scheduling problem by two-stage stochastic programming. *Expert Systems with Applications*, *233*. https://doi.org/10.1016/j.eswa.2023.120846

Ji, M., Yang, Y., Duan, W., Wang, S., & Liu, B. (2016). Scheduling of no-wait stochastic distributed assembly flowshop by hybrid PSO. *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2649–2654. https://doi.org/10.1109/CEC.2016.7744120

Jiang, S., Liu, M., Hao, J., & Qian, W. (2015). A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry. *Computers & Industrial Engineering*, *87*, 518–531. https://doi.org/10.1016/j.cie.2015.06.002

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *46*, 101–117. https://doi.org/10.1016/j.simpat.2014.02.005

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, *2*, 62–72. https://doi.org/10.1016/j.orp.2015.03.001

Kianfar, K., Fatemi Ghomi, S. M. T., & Oroojlooy Jadid, A. (2012). Study of stochastic sequence-dependent flexible flow shop via developing a dispatching rule and a hybrid GA. *Engineering Applications of Artificial Intelligence*, *25*(3), 494–506. https://doi.org/10.1016/j.engappai.2011.12.004

Kim, D.-W., Na, D.-G., & Frank Chen, F. (2003). Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer-Integrated Manufacturing*, *19*(1–2), 173–181. https://doi.org/10.1016/S0736-5845(02)00077-7

Lin, J. T., & Chen, C.-M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory*, *51*, 100–114. https://doi.org/10.1016/j.simpat.2014.10.008

Lin, J. T., Chen, C.-M., Chiu, C.-C., & Fang, H.-Y. (2013). Simulation optimization with PSO and OCBA for semiconductor back-end assembly. *Journal of Industrial and Production Engineering*, *30*(7), 452–460. https://doi.org/10.1080/21681015.2013.860926

Lin, Y.-K., & Huang, D.-H. (2020). Reliability analysis for a hybrid flow shop with due date consideration. *Reliability Engineering & System Safety*, *199*, 105905. https://doi.org/10.1016/j.ress.2017.07.008

Liu, Q., Ullah, S., & Zhang, C. (2011). An improved genetic algorithm for robust permutation flowshop scheduling. *The International Journal of Advanced Manufacturing Technology*, *56*(1–4), 345–354. https://doi.org/10.1007/s00170-010-3149-6

Liu, Y., Shen, W., Zhang, C., & Sun, X. (2023). Agent-based simulation and optimization of hybrid flow shop considering multi-skilled workers and fatigue factors. *Robotics and Computer-Integrated Manufacturing*, *80*. https://doi.org/10.1016/j.rcim.2022.102478

Minella, G., Ruiz, R., & Ciavotta, M. (2011). Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems. *Computers & Operations Research*, *38*(11), 1521–1533. https://doi.org/10.1016/j.cor.2011.01.010

Mirabi, M., Ghomi, S. M. T. F., & Jolai, F. (2013). A two-stage hybrid flowshop scheduling problem in machine breakdown condition. *Journal of Intelligent Manufacturing*, *24*(1), 193–199. https://doi.org/10.1007/s10845-011-0553-1

Pinedo, M. L. (2012). Scheduling: Theory, algorithms and systems. In *Springer* (4th ed., Vol. 4). Springer Science & Business Media. https://doi.org/10.1007/978-1-4614-2361-4

Qin, W., Zhang, J., & Song, D. (2018). An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. *Journal of Intelligent Manufacturing*, *29*(4), 891–904. https://doi.org/10.1007/s10845-015-1144-3

Rahmani, D., Heydari, M., Makui, A., & Zandieh, M. (2013). A new approach to reducing the effects of stochastic disruptions in flexible flow shop problems with stability and nervousness. *International Journal of Management Science and Engineering Management*, *8*(3), 173–178. https://doi.org/10.1080/17509653.2013.812332

Rajendran, C., & Chaudhuri, D. (1992). Scheduling in n-job, m-stage flowshop with parallel processors to minimize makespan. *International Journal of Production Economics*, *27*(2), 137–143. https://doi.org/10.1016/0925-5273(92)90006-S

Rodríguez-Espinosa, C. A., González-Neira, E. M., & Zambrano-Rey, G. M. (2023). A simheuristic approach using the NSGA-II to solve a bi-objective stochastic flexible job shop problem. *Journal of Simulation*, 1–25. https://doi.org/10.1080/17477778.2023.2231877

Rooeinfar, R., Raissi, S., & Ghezavati, V. (2019). Stochastic flexible flow shop scheduling problem with limited buffers and fixed interval preventive maintenance: a hybrid approach of simulation and metaheuristic algorithms. *SIMULATION*, *95*(6), 509–528. https://doi.org/10.1177/0037549718809542

Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, *169*(3), 781–800. https://doi.org/10.1016/j.ejor.2004.06.038

Schulz, S. (2019). *A Genetic Algorithm to Solve the Hybrid Flow Shop Scheduling Problem with Subcontracting Options and Energy Cost Consideration* (pp. 263–273). https://doi.org/10.1007/978-3-319-99993-7_23

Silva, C., Reis, V., Morais, A., Brilenkov, I., Vaza, J., Pinheiro, T., Neves, M., Henriques, M., Varela, M. L., Pereira, G., Dias, L., Fernandes, N. O., & Carmo-Silva, S. (2017). A comparison of production control systems in a flexible flow shop. *Procedia Manufacturing*, *13*, 1090–1095. https://doi.org/10.1016/j.promfg.2017.09.169

Singh, D., & Shukla, R. (2020). Multi-objective optimization of selected non-traditional machining processes using NSGA-II. *Decision Science Letters*, 421–438. https://doi.org/10.5267/j.dsl.2020.3.003

Tang, D., Dai, M., Salido, M. A., & Giret, A. (2015). Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, *81*, 82–95. https://doi.org/10.1016/j.compind.2015.10.001

Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, M., Izadi, M., & Sassani, F. (2009). Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. *Computers & Operations Research*, *36*(12), 3224–3230. https://doi.org/10.1016/j.cor.2009.02.012

Urlings, T., Ruiz, R., & Stützle, T. (2010). Shifting representation search for hybrid flexible flowline problems. *European Journal of Operational Research*, *207*(2), 1086–1095. https://doi.org/10.1016/j.ejor.2010.05.041

Wang, H., Fu, Y., Huang, M., Huang, G. Q., & Wang, J. (2017). A NSGA-II based memetic algorithm for multiobjective parallel flowshop scheduling problem. *Computers & Industrial Engineering*, *113*, 185–194. https://doi.org/10.1016/j.cie.2017.09.009

Wang, K., & Choi, S. H. (2010). A decomposition-based approach to flexible flow shop scheduling under stochastic setup times. *2010 5th IEEE International Conference Intelligent Systems*, 55–60. https://doi.org/10.1109/IS.2010.5548328

Wang, K., & Choi, S. H. (2014). A holonic approach to flexible flow shop scheduling under stochastic processing times. *Computers & Operations Research*, *43*(1), 157–168. https://doi.org/10.1016/j.cor.2013.09.013

Wang, K., Choi, S. H., & Qin, H. (2014). An estimation of distribution algorithm for hybrid flow shop scheduling under stochastic processing times. *International Journal of Production Research*, *52*(24), 7360–7376. https://doi.org/10.1080/00207543.2014.930535

Wang, K., Choi, S. H., Qin, H., & Huang, Y. (2013). A cluster-based scheduling model using SPT and SA for dynamic hybrid flow shop problems. *The International Journal of Advanced Manufacturing Technology*, *67*(9–12), 2243–2258. https://doi.org/10.1007/s00170-012-4645-7

Wang, S., Wang, L., Liu, M., & Xu, Y. (2015). An order-based estimation of distribution algorithm for stochastic hybrid flow-shop scheduling problem. *International Journal of Computer Integrated Manufacturing*, *28*(3), 307–320. https://doi.org/10.1080/0951192X.2014.880803

Wang, Y., & Xie, N. (2021). Flexible flow shop scheduling with interval grey processing time. *Grey Systems*, *11*(4), 779–795. https://doi.org/10.1108/GS-09-2020-0123

Yu, C., Andreotti, P., & Semeraro, Q. (2020). Multi-objective scheduling in hybrid flow shop: Evolutionary algorithms using multi-decoding framework. *Computers and Industrial Engineering*, *147*. https://doi.org/10.1016/j.cie.2020.106570

Zandieh, M., & Gholami, M. (2009). An immune algorithm for scheduling a hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *International Journal of Production Research*, *47*(24), 6999–7027. https://doi.org/10.1080/00207540802400636

Zandieh, M., & Hashemi, A. (2015). Group scheduling in hybrid flexible flowshop with sequence-dependent setup times and random breakdowns via integrating genetic algorithm and simulation. *International Journal of Industrial and Systems Engineering*, *21*(3), 377. https://doi.org/10.1504/IJISE.2015.072273