Capstone Final Project Application

[203020] A robust flexible flow shop problem under processing and release times uncertainty

Mario Botero Amaya[a,c] , Theo Nicolás Gelvez Gelvez[a,c] ,
Diego Andrés Rodas Herrera[a,c]

Mohamed Rabie NaitAbdallah[b,c], Eliana María González Neira[b,c],
Gabriel Mauricio Zambrano Rey[b,c]

[a]*Industrial Engineering Student*
[b]*Professor, Degree Project Director, Industrial Engineering Department*
[c]*Pontificia Universidad Javeriana, Bogotá, Colombia*

Abstract
The aim of this paper is to present a simheuristic approach that obtains robust solutions for a multi-objective hybrid flow shop problem under uncertain processing and release times. This approach minimizes the expected tardiness and standard deviation of tardiness, as a robustness measure for the stated problem. The simheuristic algorithm hybridizes the NSGA-II with a Monte Carlo Simulation process. Initially, the deterministic scenario was tested on 32 different created small size instances and 32 medium and large benchmarked instances. As a result, the proposed algorithm improved quality of solutions by 1.21% against the MILP model and it also performed better than ERD, NEHedd, and ENS2, while consuming a reasonable computational time. Afterwards, one experimental design was carried out using 10 random instances from the same benchmark as a blocking factor, where four factors of interest were considered. The factors and their respective values are number of generations (50, 100), crossover probability (0.8, 0.9), mutation probability (0.1, 0.2), and population size (60, 100). Results show that the factors instance, mutation probability and number of generations, as well as other interactions between them, have a significant effect in the total tardiness for the deterministic scenario, proving the importance of an appropriate selection of parameters when using genetic algorithms to obtain quality solutions. Then, the performance of the proposed NSGA-II was compared against ERD, NEHedd, and ENS2 methods. Results show that our algorithm improves the quality of the solutions for both objective functions, proving the robustness of our solutions for the HFS problem. Finally, two additional generalized experiments were carried out to analyze the effect of number of jobs (10, 20), number of stages (2, 3), shop condition (0.2, 0.6), probability distribution (uniform, lognormal), and CV (0.05, 0.25, 0.4) on both objective functions. The shop condition, probability distribution and CV were proven to be highly influential on the variability of the results, with the only exception being the coefficient of variation having no statistically significant effect on the total tardiness.

Keywords: NSGA-II, HFS, uncertain parameters, processing times, release times, simheuristic approach.

1. Introduction

Scheduling is the process in which a given number of resources are allocated to different tasks over time. Its role has been extensively studied due to its importance in manufacturing and service industries. The goal of this process is to optimize an established number of performance criteria, by sequencing a collection of jobs that require processing in a certain machine environment and are subject to given constraints or special characteristics (Pinedo, 2012).

This research addresses a Flexible Flow Shop (FFS), which is a general manufacturing system that consists of different stages, where each one of them has one or more machines in parallel and multiple jobs to be processed at each stage. At least one stage has two or more parallel machines, and all jobs follow the same route throughout the stages. Each one of the parallel machines at each stage could be identical, uniform, or unrelated (Zandieh, Dorri & Khamseh, 2009). Recently, many researchers have made considerable efforts to solve FFS problems using different approaches (Lee, Yoon & Park, 2019), particularly production efficiency objectives (Öztop, Tasgetiren, Kandiller, Eliiyi & Gao, 2020). This type of machine environment is relevant due to its multiple applications to real world industries such as manufacturing (Tseng, Liao & Liao, 2008), medical (Erdem, Qu & Shi, 2012), chemical (Azizoğlu, Çakmak & Kondakci, 2001), petrochemical (Rahmani & Ramezanian, 2016) and electronics industries (Yin, Stecke, Swink & Kaku, 2017).

In the related literature, many problems around the FFS configurations are treated as deterministic models, which means that all the necessary information for sequencing the jobs is known before providing and initial schedule. Deterministic modeling has created a gap between literature and practice, since real world environments have an uncertain and dynamic nature. This means that unexpected events may happen, such as new job arrivals, variation in processing times, machine breakdowns, modification in order details, among others (Goren & Sabuncuoglu, 2010).

The approach to mitigate the effects of disruptions on the system performance plays a decisive role in scheduling research. Herroelen & Leus (2005), reviews five approaches used to deal with uncertainty: reactive scheduling, proactive (robust) scheduling, stochastic scheduling, scheduling under fuzziness, and sensitivity analysis.

Reactive scheduling examines and aims improving the schedule when unexpected events occur since it does not consider uncertainty when formulating the initial schedule. Distinctively, the proactive scheduling approach seeks to obtain an initial schedule that absorbs the effects of future disruptions using performance measures such as robustness and stability. A hybrid between both approaches, usually named reactive-proactive approach, has also been studied. For examples on the reactive-proactive approach on FFS problems please refer to (F. Liu, Wang, Hong & Yue, 2017; Rahmani, 2017; Rahmani & Heydari, 2014).

According to Goren & Sabuncuoglu (2010), robustness and stability are the two topics studied to determine the negative impacts that uncertainty has on scheduling solutions. The first topic deals with the deterioration of the schedules when facing disruptions, therefore, a schedule performance that is not sensitive to change is known as robust. The second topic studies variability, where a stable schedule does not deviate from the initial schedule when it contemplates uncertainty.

The stochastic scheduling approach considers uncertain parameters as random variables and solves the decision problem as a stochastic program (Liao, Sarin & Sherali, 2012). If a fuzzy scheduling approach is applied, instead of using random variables, uncertainty is modeled with fuzzy numbers and the constraints are defined using fuzzy sets and membership functions. The advocates of the fuzzy activity duration approach claim that probability distributions for the activity durations are usually unknown due to the lack of accurate historical data (Hazır & Ulusoy, 2020). Lastly, sensitivity analysis is used to gain essential insights on quantitative models' behavior, structure, and response to changes in their inputs. Due to the number of problems that have been addressed through scientific modeling over the years, it is necessary to consistently develop sharper sensitivity analysis methods and applications (Borgonovo & Plischke, 2016).

1.1. Problem statement

This research project studies the robust FFS problem under the effect of two job-related disruptions: uncertain processing times and unexpected release times. Its aim is to minimize the total tardiness of the system by implementing a metaheuristic. This type of scheduling problem is proven to be NP-hard (Gupta, 1988).

To optimize the established criteria, a Genetic Algorithm (GA) was designed, as it is proven to perform exceptionally well when providing good and quality schedules to problems of this nature (Bozorgirad & Logendran, 2016; Ruiz & Vázquez-Rodríguez, 2010).

1.2. Justification

Since uncertainty is introduced to this research, the models that were developed during this project provided meaningful insights to real shop configurations that have a resemblance to the one studied in this paper. Uncertainty was addressed from a proactive approach because it is expected that the schedule becomes relatively insensitive to a changing environment and disruptions (Sevaux & Sörensen, 2004). Additionally, by aiming to optimize an objective function that takes due dates into consideration, this project addresses the improvement of indicators such as on-time delivery and can be applied to manufacturing environments where a good level of service is intended to be maintained.

The selected disruptions contribute vastly to the study of uncertainty in FFS problems. Despite the fact that processing times are the most common source of uncertainty studied in the literature (González Neira, Montoya-Torres & Barrera, 2017), its consideration continues to be relevant because of the occurrence of disruptive events such as machine breakdowns, processing limits, equipment conditions, operator skills and shortage of raw materials. Furthermore, release times are influenced by the upstream processing procedure, delivery time, and other uncertain factors of the supplier's logistic chain. As a result, exact release times usually cannot be known in advance, and only estimated release times intervals are available based on historical data and real-time prediction (M. Liu & Liu, 2019; Yue, Song, Zhand, Gupta & Chiong 2018).

The metaheuristic that was implemented is a GA, since it has been proven that it outperforms other heuristics and metaheuristics in FFS environments. For example, Ruiz & Vázquez-Rodríguez (2010) compared different case studies such as a check-processing company and a ceramic tile production shop. They concluded that the GA is superior to Ant Colony Optimization (ACO) heuristics, Tabu Search (TS), Simulated Annealing (SA), and some other procedures. The extensive research dedicated to this population-based algorithm provides a framework from which some adaptations can be considered to improve the performance of the metaheuristic (Luo & El Baz, 2018; Werner, 2011).

2. Literature Review

Over the years, scheduling problems have been broadly studied in the literature due to the extensive number of methodologies continuously emerging. Researchers develop models around different machine configurations, such as single machine, parallel machine, flow shop, job shop, among others. After a solution is obtained, its performance is usually compared to other algorithms in the literature. This review focuses on flow shop problems under some source of uncertainty.

Chaari, Chaabane, Loukil & Trentesaux (2011) studied a FFS problem under stochastic processing times aiming to minimize makespan. They used a GA to solve the problem, which allowed them to achieve good solutions and find that the GA helps quantify the impact of uncertainty, managing risk. Feng, Zheng & Xu (2016) studied the makespan minimization scheduling problem in a two-stage hybrid flow shop. The authors assumed job processing times to be uncertain as well and proposed both exact and heuristic algorithms to solve this problem. Other researchers such as Goren & Sabuncuoglu (2010), considered uncertain processing times and machine failures as stochastic parameters. A beam search heuristic (BS) was proposed to solve the problem addressing five robustness and stability measures.

Other authors have studied the FFS scheduling problem only under uncertain machine breakdowns. For example, Fazayeli, Aleagha, Bashirzadeh & Shafaei (2016) used genetic and simulated annealing algorithms, outperforming other solution methods in terms of β-robustness of makespan. Sahar, Hany, Hamed & Rasoul (2019) proposed an imperialist competitive algorithm (ICA) and GA to address the robust scheduling problem of a two-stage assembly FS. The authors used an artificial neural network to predict the value of parameters under uncertain conditions. Cui, Lu, Li & Han (2018) used a proactive approach along with a two-loop algorithm to deal with failure uncertainty, which as an important remark, proved that the impact of idle times is larger than the impact of job's sequence. A surrogate measure was also implemented, aiming to reduce the computation time of the algorithm.

As stated earlier, a proactive-reactive approach to deal with disruptions in flow shop problems has gathered the interest of some researchers. F. Liu et al. (2017) used the total flow time as the schedule performance measure; the authors propose a hybridization strategy that successfully enhances the classic Non-dominated Sorting GA as the solution method. Rahmani & Heydari (2014) used a two-way procedure which addressed robustness and stability on an environment facing unexpected arrivals of new jobs and uncertain processing times. A similar method was used in Rahmani (2017), where a multi-criteria measure was defined not only by robustness and stability, but also by solution effectiveness and reduction of system nervousness.

Among other ways to deal with uncertainty, Zuo, Mo & Wu (2009) proposed a robust scheduling method by modeling an uncertain scheduling problem with a set of workflow models in which a multi-objective variable neighborhood immune algorithm (VNIA) was used to find a robust scheduling scheme. Long et al. (2020) focused on stochastic release times while trying to minimize makespan; this problem was solved using a robust dynamic scheduling approach based on release time series forecasting with two stages. These researchers presented a novelty consisting of the forecasting accuracy of the model, addressing uncertainty in release times to generate a robust solution. Rahmani, Heydari, Makui & Zandieh (2013) and Rahmani & Ramezanian (2016) considered a stochastic environment with new job arrivals. Both applied reactive scheduling methods to solve the problem. A variable neighborhood search (VNS) algorithm was used in the latter one and results showed that the proposed approach outperformed previous rescheduling processes.

The inclusion of robustness measures into the optimization criteria ensures that robust solutions will be obtained and the effect of disruptions on the realized schedules will be effectively anticipated and properly addressed. A review conducted by Sabuncuoglu & Goren (2009) will be referenced to contemplate different measurements used in proactive scheduling. A differentiation is made between measures based on the actual performance of realized schedules and measures associated with regret, which refers to the difference between realized and optimal performances. In this paper, only the former type of measurements will be contemplated, addressing the minimization of the following functions: expected realized performance, worst-case performance, worst-case scenario's performance, most probable scenario's performance, variance of realized performance, and expected deviation of the realized schedule's performance from the initial deterministic performance. A combination of two different measurements can be implemented by assigning different weights to each measure considered.

According to Hazır & Ulusoy (2020), stochastic scheduling has been studied for decades, while reactive, robust and fuzzy scheduling have all increased in popularity among researchers for the past 20 years. This review also points out the importance of considering more than just one source of uncertainty. González Neira et al. (2017) provided a review on FFS problems under uncertainty, identifying that dealing with more than one parameter is an opportunity for further research since it only was considered in the 21% of the reviewed papers. It is also important to note that release times were present as uncertain parameters in only 3% of the papers.

Taking the previous statement under consideration, this research contributes significantly to the study of FFS problems under uncertainty. It aims to study the influence of processing times and release times as sources of uncertainty, the latter one being one of the least studied in the literature. It also takes a proactive approach to deal with disruptions and effectively generate robust schedules, while minimizing the total tardiness of the system. This objective function is not commonly implemented in these procedures since most studies consider makespan as optimization criteria. This type of FFS problem, according to the conducted review, is yet to be studied. Similarities with problems addressed in the past will be considered while designing the methodology and development of the model, including robustness measurements and improved algorithms known for its performance.

3. Objectives

Design an optimization procedure for robust total tardiness minimization in a Flexible Flow Shop scheduling problem under uncertain processing and release times.

- Define the methodology for robustness measuring and uncertain parameters modelling.

- Implement a genetic algorithm to obtain robust solutions for a FFS under uncertainty with total tardiness minimization.
- Evaluate the solution method performance regarding the established robustness criteria.

4. Methodology

This section presents the methodology that was followed to achieve the results presented in section 5. First, the mixed integer lineal programming (MILP) model is presented to understand the constraints of our problem. Then, robustness measures used to obtain robust solutions are presented. Finally, NSGA-II representation and operators are explained, as well as the implemented framework for both deterministic and stochastic scenarios.

4.1. MILP model

In this section, a MILP model is proposed for the solution of the deterministic part of the problem, aiming to minimize tardiness in a HFS. The model is based on the one proposed by Naderi, Gohari, & Yazdani, (2014), and adapted to the characteristics of our problem. The MILP model for the stated problem is presented in this section. For better understanding of the model, the notation used is presented:

Sets

I_s = *machines of stage s {1,…,* m_s*}*

J = jobs {1,…, n}

S = stages {1, …, s}

Parameters

p_{js} = processing time of job j at stage s, $\forall\, j \in J, \forall\, s \in S$

r_j = release time of job j , $\forall\, j \in J$

d_j = due date of job j, $\forall\, j \in J$

M = large positive number

Variables

X_{jks} = 1, if job j is processed after job k at stage s, $\forall\, k \in J, \forall\, s \in S$; 0, otherwise

Y_{ijs} = 1, if job j is processed at stage s on machine i, $\forall\, j \in J, \forall\, s \in S, \forall\, i \in I_s$; 0, otherwise

T_j = tardiness of job j, $\forall\, j \in J$

C_{js} = completion time of job j at stage $s, \forall\, j \in J, \forall\, s \in S$

Equations

$\sum_{i \in I} Y_{ijs} = 1, \forall j \in J, s \in S$ (1)

$C_{js} \geq p_{js} + r_j, \forall j \in J, s \in S \mid s = 1$ (2)

$C_{js} \geq C_{js-1} + p_{js}, \forall j \in J, s \in S \mid s > 1$ (3)

$C_{js} \geq C_{ks} + p_{js} - \mathrm{M} * (3 - X_{jks} - Y_{ijs} - Y_{iks}), \forall\, s \in S, i \in I_s, (j \neq k) \in J$ (4)

$C_{ks} \geq C_{js} + p_{ks} - \mathrm{M} * X_{jks} - \mathrm{M} * (2 - Y_{ijs} - Y_{iks}), \forall\, s \in S, i \in I_s, (j \neq k) \in J$ (5)

$T_j \geq C_{js} - d_j, \forall j \in J$ (6)

$T_j \geq 0, \forall j \in J$ (7)

$C_{js} \geq 0, \forall j \in J, s \in S$ (8)

$X_{jks} \in \{0, 1\}, \forall j \in J, k \in J, s \in S$ (9)

$Y_{ijs} \in \{0, 1\}, \forall i \in I, j \in J, s \in S$ (10)

Objective function

$$Min \sum_{j \in J} T_j \quad (11)$$

Constraint set (1) specifies the job assignment to one machine among the available machines at each stage. Constraint sets (2) and (3) determine the completion time for each job at each stage. Constraint set (4) and (5) determine the completion time for each job at each stage regarding the completion time of jobs that have been previously processed at the same machine. Constraint set (6) determines the tardiness of each job at each stage. Constraint set (7), (8), (9) and (10) define the decision variables. The objective function (11) is to minimize the total tardiness.

4.2. Robustness measuring

The standard deviation of the total tardiness will be implemented in the optimization criteria, ensuring that solution robustness will be considered when providing a solution in an uncertain environment. Given our interest in obtaining both high quality and robust solutions, a multi-objective function composed by the expected value of total tardiness and its standard deviation across a finite number of disrupted simulations is introduced to the work.

4.3. Framework of deterministic NSGA-II

Since we are considering an NP-hard optimization problem with two objectives, in which Pareto-optimal solutions are introduced, a metaheuristic becomes necessary to obtain good solutions in an adequate computational time. Therefore, we chose the NSGA-II multi-objective evolutionary algorithm (MOEA) proposed by Deb, Pratap, Agarwal, & Meyarivan (2002), which is an improved version of the NSGA proposed by Srinivas & Deb (1994). This better version overcomes the main critiques made to the initial version: high computational complexity of nondominated sorting, lack of elitism and need for specifying the sharing parameter σ_{share}. This metaheuristic is proven to outperform other MOEAs such as Pareto-achieved evolution strategy (PAES) and strength Pareto evolutionary algorithm (SPEA) when it comes to diverse and converging solutions near the true Pareto-optimal set.

Before considering any introduction of stochastic components to our problem, this algorithm must be designed in a deterministic environment. Then, its performance will be validated by comparing its results in contrast to those of other solution methods.

4.3.1. Chromosome representation

Each chromosome contains a series of numbers, indicating the sequence in which the jobs must be processed at the first stage. Taking Figure 1 as an example of a permutation to be followed, the third job is the first one to be processed at the first available machine at the first stage, followed by job 5 and so on. In a hypothetical scenario where job 5 is completed before job 3 at the first stage, job 5 would be the first job to start processing at the next stage, according to the FCFS rule.

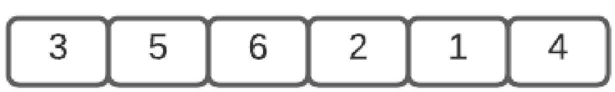

Figure 1. Representation of a chromosome.

It is important to highlight that an individual is a direct representation of a schedule that will be processed in the first stage (Rajkumar & Shahabudeen, 2009). This individual represents a permutation of the number of jobs *n* that must be processed. We implemented the First Come First Served (FCFS) dispatching rule to sequence the jobs through the rest of the HFS. We also used the First Available Machine (FAM) assignment rule since we have identical machines in each stage, so the total processing time in either machine of one job is the same.

4.3.2. Initial population

An initial population of size P_{size} is generated and represented as $pop = \{\pi_1, \pi_2, \dots, \pi_{Psize}\}$. The first individual π_1 is generated as the result of Earliest Release Date (ERD) dispatching rule, π_2 as the result of NEH_{EDD} heuristic of Nawaz, Enscore, & Ham (1983), and all other individuals randomly.

4.3.3. Fitness evaluation

In our case, the fitness evaluation is made based on the total tardiness, which is the sum of the tardiness of jobs 1 through *n*. Let us be reminded that tardiness occurs when the completion time of any job *j* is greater than its due date, therefore the tardiness equation can be defined as $T_j = max\,(c_j - d_j, 0)$.

4.3.4. Selection of individuals

In the NSGA-II, the parent solutions are selected according to non-domination criteria, in which the initial solutions are divided into ranks $(F_1, F_2, \dots, F_n)$ so that solutions in rank F_1 are those chromosomes that have the best positions in the Pareto-optimal frontiers and must be emphasized when choosing the best individuals. Solutions belonging to rank F_2 are those only dominated by solutions in F_1, and so on. The selection of individuals continues until a new population with the same size as the initial population is obtained (Deb et al., 2002).

4.3.5. Crossover

Two-Point (TP) crossover is considered in this work. This operator randomly selects two points for dividing one parent. The jobs outside the selected two points are inherited from one parent to the child, and the remaining jobs are placed in the order in which they appear in the other parent, as shown in Figure 2. This procedure is executed in both possible ways, obtaining two children.

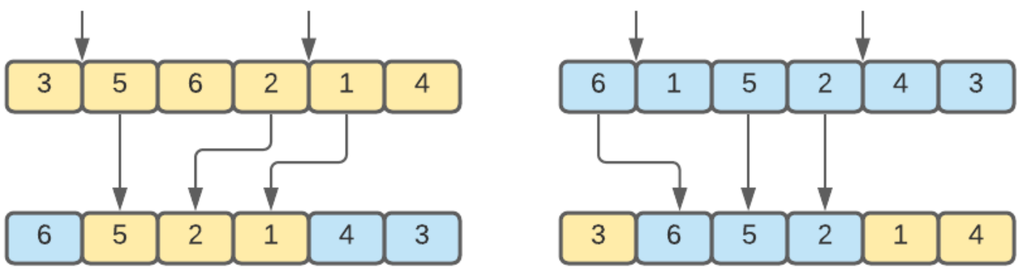

Figure 2. Two-Point Crossover Representation.

4.3.6. Mutation

A shift mutation is introduced. This mutation consists of selecting randomly two different positions and removing the job at one of such positions and inserting it at the other selected position. An example of this mutation operator is shown in Figure 3. Both operators were selected since they have been proven to perform better when using genetic algorithms (Murata, Ishibuchi, & Tanaka, 1996).

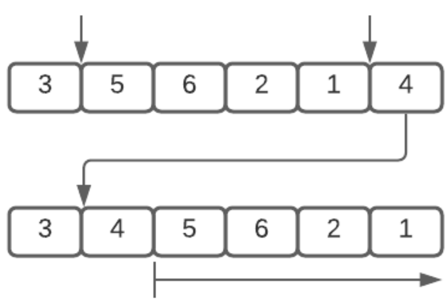

Figure 3. Shift Mutation Representation.

4.3.7. Stopping criteria

The algorithm stops when a number of generations (NGEN) is reached. In subsection 5.1 the parameter selection of the proposed NSGA-II is presented.

4.4. Framework of stochastic NSGA-II

4.4.1. Uncertain parameters modelling

One way to model uncertainty is through probability distributions. Given that this paper considers stochastic processing and release times, we chose uniform and lognormal distributions to model them. Uniform distributions are associated with the fact that there exists a time horizon with known upper and lower bounds (which in our case are always positive), and therefore a random variable fitted to this distribution could take any value in that range with the same probability. In the same way, lognormal distribution generates values given a mean and a standard deviation, but since it is skewed to the right, it allows data mostly on the positive side ("Appendix A: Practical Processing Time Distributions," 2018). Based on the explanation given above, these distributions fit well into our problem because of their resemblance to the stochastic behavior of these parameters in real world scenarios, the elimination of the possibility of generating negative values, and their capacity to accommodate to different values of coefficients of variation (CV).

4.4.2. Simheuristic approach

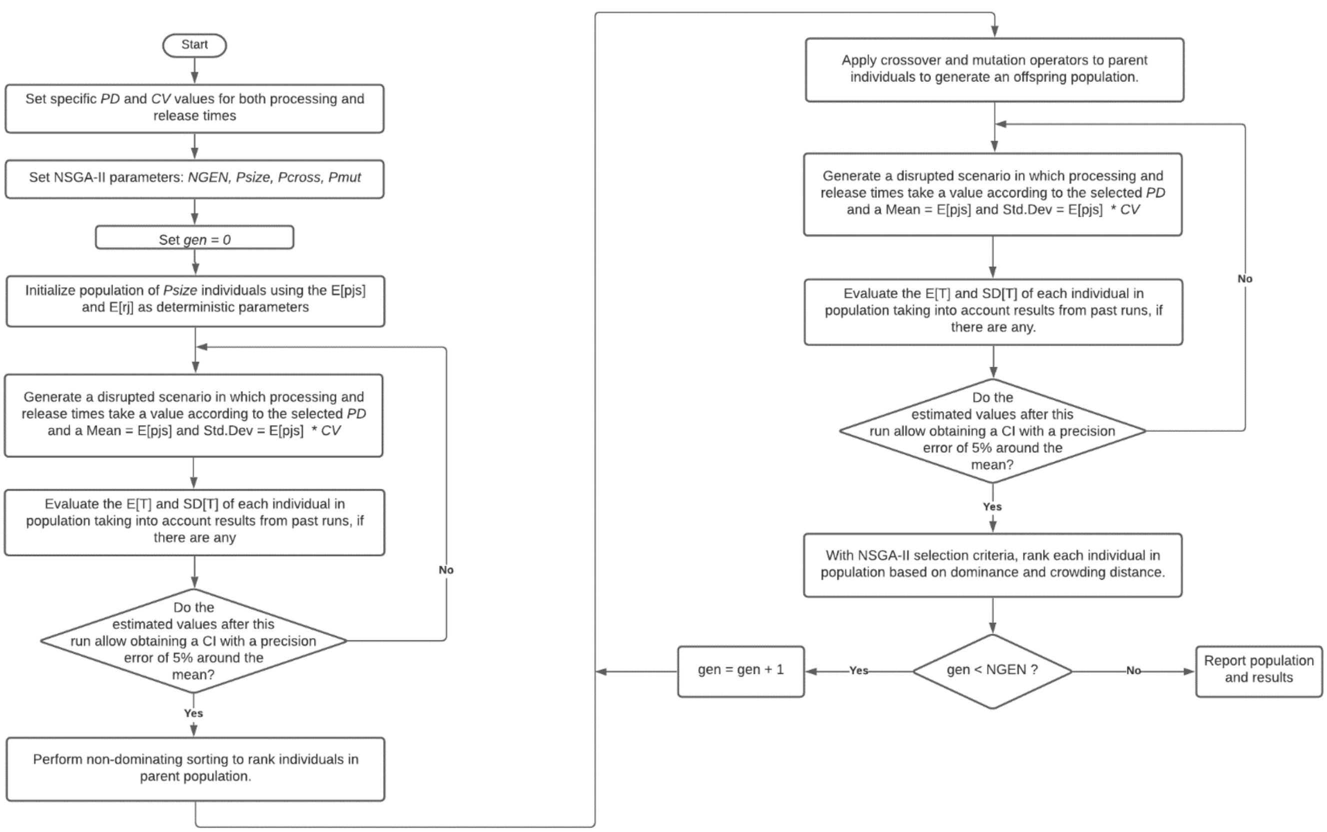

Figure 4. Multi-Objective Simheuristic NSGA-II Proposed Procedure.

The procedure starts with the selection of an instance, where the values of processing and release times correspond to their expected values. Next, a probability distribution and a CV are selected. The Monte Carlo simulation is used when it is required to evaluate individuals, generating at each run a realization for the stochastic processing and release times based on their expected values, the probability distributions and CV. To determine if the obtained results allow expected values for the total tardiness within a confidence interval with a 95% confidence and a precision error of 5% around the mean, the procedure proposed by Framinan & Perez-Gonzalez (2015) is implemented. In case the confidence interval of a run does not satisfy the desired error, another run must be executed with new processing and release times that are generated in the same way as mentioned above. Otherwise, the results are reported, and the NSGA-II continues its ranking and selection process. The process ends when a maximum of generations is reached and the final population is reported. This procedure is represented in Figure 4.

5. Computational results

This section presents all computational experiments for parameter selection and evaluation of the performance of the proposed NSGA-II. All of the runs were made in an Intel ® Core i5 1.60 GHz computer with 12GB RAM memory. The mathematical model and the NSGA-II were programmed in Spyder integrated development environment (IDE) from Python.

5.1. NSGA-II parameter selection

A generalized factorial design in RStudio software was carried out to determine the best levels of the considered factors. These factors are presented next with their respective levels: population size (60, 100), crossover probability (0.8, 0.9), mutation probability (0.1, 0.2), and number of generations (50, 100). Also, 10 random instances out of 3060 benchmarked instances from Pan, Ruiz, & Alfaro-Fernández (2017) were selected and used as a blocking factor, and the response variable is the total tardiness. There are 16 treatments in total and two observations for each of them, having a total of 320 observations for the proposed algorithm in Spyder IDE with Python for further statistical analysis.

First, the verification of assumptions (normality, homoscedasticity and independence of residuals) was made to determine if we could analyze the results of the ANOVA. Shapiro-Wilk and Anderson-Darling tests were performed to verify if the residuals adjust to a normal distribution. As a result of these tests, we obtained p-values of 6 x 10^{-15} and 2 x 10^{-16}, respectively. Therefore, since both p-values are lower than the significance of 0.05, this assumption is not satisfied with any of the tests.

Next, homoscedasticity assumption was analyzed graphically because it was not possible to perform Levene nor Bartlett tests since two observations were registered per treatment and to the non-satisfaction of normality assumption, respectively. As it is presented in Figure 5, not all the columns present a similar dispersion of points, meaning that the satisfaction of this assumption is doubtful. Finally, residuals independence over time was using Durbin-Watson test. We obtained a p-value of 0.062, concluding with a significance of 0.05 that this assumption is satisfied.

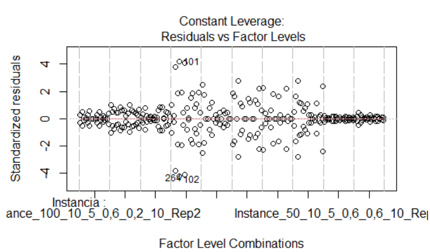

Figure 5. Residuals vs Adjusted Values.

Considering the results of the assumptions, ANOVA-Type Statistics (ATS) nonparametric test was performed due to the breach of normality and homoscedasticity assumptions. Figure 6 contains the results of ATS, concluding with a 0.05 significance that the factors instance, mutation probability, number of generations, and the interactions population size-crossover probability, instance-mutation probability, instance-number of generations, mutation probability-number of generations, instance-population size-crossover probability, and instance-population size-crossover probability-mutation probability have a significant effect on the total tardiness. Finally, Table 1 presents the best treatment(s) for each instance according to the descriptive analysis of the ATS test, using confidence intervals. Three of the treatments represent the best configuration for the selected instances, based on the number of times such configurations appear on Table 1. To break this tie, average computational time was analyzed as presented in Table

2, allowing to conclude that the best treatment among the selected instances is: $P_{size} = 60$, crossover probability = 0.8, mutation probability = 0.1, and number of generations = 100, since it obtained the lowest time.

```
ANOVA.Type.Statistic:
                                     Statistic df1  df2 p-Value
Instancia                              3.1e+03 3.6  26  0.0000
Psize                                  2.3e+00 1.0  26  0.1443
Instancia:Psize                        4.3e-01 3.6  26  0.7646
Pcruce                                 6.3e-01 1.0  26  0.4348
Instancia:Pcruce                       1.0e+00 3.6  26  0.4200
Psize:Pcruce                           7.9e+00 1.0  26  0.0093
Instancia:Psize:Pcruce                 2.9e+00 3.6  26  0.0448
Pmut                                   7.1e+02 1.0  26  0.0000
Instancia:Pmut                         2.3e+01 3.6  26  0.0000
Psize:Pmut                             1.8e+00 1.0  26  0.1871
Instancia:Psize:Pmut                   6.7e-01 3.6  26  0.6034
Pcruce:Pmut                            2.1e-01 1.0  26  0.6511
Instancia:Pcruce:Pmut                  5.4e-01 3.6  26  0.6914
Psize:Pcruce:Pmut                      3.3e+00 1.0  26  0.0797
Instancia:Psize:Pcruce:Pmut            5.2e+00 3.6  26  0.0044
Gen                                    6.2e+01 1.0  26  0.0000
Instancia:Gen                          3.0e+00 3.6  26  0.0402
Psize:Gen                              1.3e-01 1.0  26  0.7182
Instancia:Psize:Gen                    1.4e+00 3.6  26  0.2737
Pcruce:Gen                             5.4e-02 1.0  26  0.8186
Instancia:Pcruce:Gen                   1.0e+00 3.6  26  0.4052
Psize:Pcruce:Gen                       1.7e-01 1.0  26  0.6801
Instancia:Psize:Pcruce:Gen             7.5e-01 3.6  26  0.5539
Pmut:Gen                               6.6e+00 1.0  26  0.0165
Instancia:Pmut:Gen                     8.0e-01 3.6  26  0.5236
Psize:Pmut:Gen                         4.6e+00 1.0  26  0.0422
Instancia:Psize:Pmut:Gen               2.7e+00 3.6  26  0.0579
Pcruce:Pmut:Gen                        2.4e+00 1.0  26  0.1301
Instancia:Pcruce:Pmut:Gen              1.2e+00 3.6  26  0.3254
Psize:Pcruce:Pmut:Gen                  3.1e-01 1.0  26  0.5830
Instancia:Psize:Pcruce:Pmut:Gen        7.1e-01 3.6  26  0.5761
```

Figure 6. ATS Test Results.

Table 1. Best Treatment(s) per Instance.

Instance	Population size	Crossover probability	Mutation probability	Number of generations
1	60	0.9	0.1	100
2	100	0.8	0.1	100
	60	0.8	0.1	100
3	100	0.8	0.1	100
4	100	0.8	0.1	100
5	60	0.8	0.1	100
	100	0.9	0.1	100
6	60	0.8	0.1	100
	100	0.9	0.1	100
	60	0.9	0.1	100
	100	0.8	0.1	100
	100	0.8	0.1	50
7	100	0.9	0.1	100
	60	0.8	0.1	100
8	100	0.8	0.1	100
	100	0.9	0.1	100
	60	0.8	0.1	100
9	100	0.9	0.1	100
10	60	0.9	0.1	50

Table 2. Average Computational Time for Best Treatment(s).

Population size	Crossover probability	Mutation probability	Number of generations	Average computational time
100	0.8	0.1	100	847.7482877
60	0.8	0.1	100	775.380458
100	0.9	0.1	100	882.7456122

5.2. NSGA-II performance in a deterministic environment for small instances

To evaluate the performance of our NSGA-II in the deterministic scenario in small instances, 32 different instances were generated: 16 of 6 jobs and 16 of 8 jobs for each combination of tardiness factor (0.2, 0.6), due date range (0.2, 0.6), number of stages (2, 3) and number of machines per stage (2, 3). The processing and release times were generated following the procedure presented by Pan, Ruiz, Alfaro-Fernández (2017), where processing times are a random value generated from a uniform distribution between the interval [1, 99] and the release times follow the same procedures as the authors with and additional adaptation that considers the presence of due dates, as proposed by Haouari & Hidri (2008). Then, the 32 created small instances and 32 existing instances by Pan et al., (2017) for 10 and 15 jobs for each combination of the same factors were solved in Spyder IDE with Python with a maximum running time of 3600 seconds per instance. It is important to highlight that the comparison of the performance of our algorithm is made against the MILP model, ERD dispatching rule, NEHedd and Extensive Neighborhood Search (ENS2) heuristics, since a problem with the same characteristics of this research has not been found in the literature to our knowledge. This comparison considers only the solutions with best total tardiness of the Pareto frontiers, since only one of the objective functions is being compared for the deterministic scenario. This step is very important because it allows us to establish whether our algorithm provides good solutions or not.

Results for tardiness of NSGA-II, MILP, ERD, NEHedd and ENS2 are presented in Tables 3 through 6. The proposed metaheuristic obtained the same optimal solutions as the proposed MILP model in 85.71% of the tested runs. Also, when optimal solutions were not reached with the MILP model, our algorithm improved the solutions by 1.21% on average. Results also show that our algorithm provides better solutions than ERD, NEHedd, and ENS2, especially in larger instances, while consuming a reasonable computational time. These results prove the efficiency of our algorithm.

Table 3. Computational results in a deterministic environment of instances with 6 jobs.
* Is the optimal solution found before 3600 seconds of running.

Instance					Total tardiness				
Number of jobs	**Number of stages**	**Machines per stage**	**Tardiness factor**	**Due date range**	**NSGA-II**	**MILP**	**ERD**	**NEHedd**	**ENS2**
6	2	2	0.2	0.2	77	77*	105	129	77
				0.6	208	230	264	217	217
			0.6	0.2	562	562	562	562	562
				0.6	499	499	709	539	499
		3	0.2	0.2	252	264	348	264	252
				0.6	61	61*	95	125	61
			0.6	0.2	370	370*	401	370	370
				0.6	466	466*	466	485	466
	3	2	0.2	0.2	247	252	433	247	247
				0.6	265	262*	313	279	271
			0.6	0.2	703	736	839	703	703
				0.6	453	453	497	524	453
		3	0.2	0.2	96	96*	111	149	96
				0.6	141	141*	157	148	141
			0.6	0.2	363	363	372	395	363
				0.6	714	714	741	720	716
Average improvement of NSGA-II						**1.21%**	**16.36%**	**11.21%**	**0.42%**

Table 4. Computational results in a deterministic environment of instances with 8 jobs.

Instance					Total tardiness				
Number of jobs	**Number of stages**	**Machines per stage**	**Tardiness factor**	**Due date range**	**NSGA-II**	**MILP**	**ERD**	**NEHedd**	**ENS2**
8	2	2	0.2	0.2	554	554	711	586	576
				0.6	671	671	885	702	671
			0.6	0.2	804	804	978	845	804
				0.6	777	777	880	817	777
		3	0.2	0.2	801	801	948	802	801
				0.6	655	655	697	662	655
			0.6	0.2	801	830	842	823	820
				0.6	870	870	971	870	870
	3	2	0.2	0.2	949	947	1034	999	949
				0.6	856	849	1220	859	856
			0.6	0.2	1178	1178	1449	1227	1178
				0.6	904	908	1224	1033	904
		3	0.2	0.2	917	944	1007	924	924
				0.6	1255	1263	1364	1350	1255
			0.6	0.2	1001	1001	1117	1041	1001
				0.6	789	783	845	789	789
Average improvement of NSGA-II						**0.35%**	**14.34%**	**3.56%**	**0.43%**

Table 5. Computational results in a deterministic environment of instances with 10 jobs.

Instance					Total tardiness				
Number of jobs	**Number of stages**	**Machines per stage**	**Tardiness factor**	**Due date range**	**NSGA-II**	**MILP**	**ERD**	**NEHedd**	**ENS2**
10	2	2	0.2	0.2	140	140	298	177	140
				0.6	62	56	210	111	62
			0.6	0.2	1146	1172	1376	1147	1146
				0.6	588	592	706	610	610
		3	0.2	0.2	69	78	95	91	71
				0.6	243	244	297	276	242
			0.6	0.2	640	642	760	749	642
				0.6	771	771	872	785	771
	3	2	0.2	0.2	411	419	703	438	438
				0.6	31	38	126	144	104
			0.6	0.2	1003	987	1301	1044	1005
				0.6	1656	1667	1834	1724	1656
		3	0.2	0.2	163	159	223	202	163
				0.6	388	388	445	412	388
			0.6	0.2	1262	1266	1443	1382	1262
				0.6	989	996	1298	1051	1017
Average improvement of NSGA-II						**1.39%**	**28.47%**	**15.84%**	**5.35%**

Table 6. Computational results in a deterministic environment of instances with 15 jobs.

Instance					Total tardiness				
Number of jobs	**Number of stages**	**Machines per stage**	**Tardiness factor**	**Due date range**	**NSGA-II**	**MILP**	**ERD**	**NEHedd**	**ENS2**
15	2	2	0.2	0.2	242	297	423	329	253
				0.6	75	126	700	156	131
			0.6	0.2	1803	2111	2801	2115	1886
				0.6	1252	1292	1569	1256	1160
		3	0.2	0.2	257	392	334	398	269
				0.6	231	310	592	456	258
			0.6	0.2	1032	1154	1709	1194	1119
				0.6	1435	1649	1740	1501	1435
	3	2	0.2	0.2	370	571	727	508	384
				0.6	23	31	722	151	68
			0.6	0.2	2280	3147	3414	2396	2293
				0.6	1473	2116	2340	1610	1509
		3	0.2	0.2	517	707	767	743	567
				0.6	422	546	899	583	461
			0.6	0.2	1708	1952	2132	1925	1705
				0.6	2248	2650	3049	2454	2273
Average improvement of NSGA-II						**22.27%**	**42.32%**	**24.95%**	**9.83%**

5.2. NSGA-II performance in a stochastic environment

To prove the efficiency of our algorithm in the stochastic scenario, we compared our solutions with those obtained from two different heuristics and one dispatching rule, using a total of 4 random instances for 10 and 20 jobs by Pan et al., 2017). ENS2 and NEHedd heuristics were selected given their recognition as effective solution methods for the total tardiness problem, as proved in González-Neira, Montoya-Torres, & Caballero-Villalobos, (2019). Because of the presence of release dates in our problem, we considered convenient the inclusion of a solution method that takes them into account, considering the objective of this work. Hence, the inclusion ERD dispatching rule. ENS2 is a local search algorithm that starts from an initial sequence and aims to improve the objective function by interchanging a pair of jobs of the given sequence, interchanging the positions if there is an improvement in the objective function. This procedure ends when no improvement is achieved (Kim, Lim, & Park, 1996). NEHedd heuristic starts from the sequence obtained from the Earliest Due Date (EDD) dispatching rule, and then it evaluates the performance of each sequence when inserting a new job in every possible slot of the same sequence. For example, if sequence [1, 2, 3] is the one obtained from EDD, and we want to introduce job 4, the heuristic evaluates the sequences [4, 1, 2 ,3], [1, 4, 2, 3], [1, 2, 4, 3], and [1, 2, 3, 4]. The sequence with the best objective function remains (Fernandez-Viagas & Framinan, 2015).

Analyzed probability distributions are uniform and lognormal for both uncertain parameters, and CV values are 0.05, 0.25 and 0.4 for both probability distributions, accounting to a total of 24 observations to be compared. All instances were tested for two machines per stage scenario and for every observation the same distribution probabilities and CV values were assigned for both uncertain parameters. Also, shop condition definition was used according to Lodree, Jang, & Klein, (2004) to obtain a highly variated analysis of due date tightness and release dates variance for the performance of our algorithm. A high shop condition implies jobs arriving during a long interval of time after scheduling begins, and tight due dates, whereas low shop condition implies jobs arriving during small intervals of time short after the beginning of the scheduling, and loose due dates. For low shop and high shop conditions, we selected values of 0.2 and 0.6 for both tardiness factor and due date range, respectively.

Tables 7 and 9 present the performance of our NSGA-II and its comparison against all three proposed methods for total tardiness, while Tables 8 and 10 present the same results for standard deviation of tardiness. Table 11 presents the results of the performance of our algorithm in the deterministic scenario for the same instances that were tested for the stochastic scenario.

Table 7. NSGA-II, ERD, NEHedd and ENS2 total tardiness in a stochastic environment for 10 jobs.

Factor					Total tardiness			
Number of jobs	**Number of stages**	**Shop condition**	**PD**	**CV**	**NSGA-II**	**ERD**	**NEHedd**	**ENS2**
10	2	Low	Uniform	0.05	125.60	288.94	164.30	130.10
				0.25	156.71	321.77	211.22	180.69
				0.4	198.72	346.85	253.77	222.12
			Lognormal	0.05	100.10	254.20	136.12	109.24
				0.25	70.44	180.81	98.83	75.58
				0.4	75.78	168.52	107.37	80.92
		High	Uniform	0.05	741.18	888.44	816.20	956.75
				0.25	769.70	954.78	923.63	967.56
				0.4	807.78	996.81	951.53	1012.78
			Lognormal	0.05	702.50	909.06	844.16	966.71
				0.25	632.14	819.51	844.85	865.55
				0.4	598.79	587.70	485.32	487.87
Average improvement of NSGA-II						**35.49%**	**17.93%**	**11.96%**

Table 8. NSGA-II, ERD, NEHedd and ENS2 standard deviation of tardiness in a stochastic environment for 10 jobs.

Factor					Standard deviation of tardiness			
Number of jobs	**Number of stages**	**Shop condition**	**PD**	**CV**	**NSGA-II**	**ERD**	**NEHedd**	**ENS2**
10	3	Low	Uniform	0.05	8.46	21.00	23.29	15.51
				0.25	53.54	121.24	105.77	94.80
				0.4	104.63	186.91	154.35	139.96
			Lognormal	0.05	8.00	20.94	17.52	16.10
				0.25	37.71	87.82	62.87	55.05
				0.4	64.13	115.36	99.75	81.65
		High	Uniform	0.05	16.27	128.40	144.79	229.58
				0.25	75.41	195.73	249.56	263.80
				0.4	118.42	256.74	284.41	312.25
			Lognormal	0.05	13.38	128.12	143.96	194.18
				0.25	55.01	171.18	195.90	238.54
				0.4	84.92	152.32	141.01	138.04
Average improvement of NSGA-II						**60.60%**	**57.89%**	**54.37%**

Table 9. NSGA-II, ERD, NEHedd and ENS2 total tardiness in a stochastic environment for 20 jobs.

Factor					Total tardiness			
Number of jobs	**Number of stages**	**Shop condition**	**PD**	**CV**	**NSGA-II**	**ERD**	**NEHedd**	**ENS2**
20	2	Low	Uniform	0.05	311.36	955.54	336.74	357.80
				0.25	393.98	964.93	478.04	535.50
				0.4	487.65	1029.21	628.68	625.09
			Lognormal	0.05	255.96	857.84	279.80	283.86
				0.25	167.91	604.58	232.58	225.22
				0.4	147.12	582.45	249.59	233.65
		High	Uniform	0.05	1224.60	3194.88	1634.68	1512.72
				0.25	1278.76	3243.76	1814.65	1780.46
				0.4	1494.23	3441.31	2221.79	2187.79
			Lognormal	0.05	1119.44	3045.44	1510.80	1417.66
				0.25	924.76	2785.47	1354.10	1306.98
				0.4	970.33	2690.11	1441.62	1438.35
Average improvement of NSGA-II						**64.09%**	**25.22%**	**24.62%**

Table 10. NSGA-II, ERD, NEHedd and ENS2 standard deviation of tardiness in a stochastic environment for 20 jobs.

Factor					Standard deviation of tardiness			
Number of jobs	**Number of stages**	**Shop condition**	**PD**	**CV**	**NSGA-II**	**ERD**	**NEHedd**	**ENS2**
20	3	Low	Uniform	0.05	15.10	52.45	24.57	35.71
				0.25	77.07	220.69	169.78	196.65
				0.4	124.06	384.29	309.08	292.82
			Lognormal	0.05	10.55	47.10	23.61	32.53
				0.25	44.39	171.25	102.56	119.66
				0.4	56.38	274.46	175.72	178.61
		High	Uniform	0.05	29.45	88.05	69.68	43.60
				0.25	130.86	387.88	322.57	237.25
				0.4	260.90	1068.00	494.82	653.51
			Lognormal	0.05	23.16	76.87	72.79	40.73
				0.25	115.33	538.50	258.67	160.04
				0.4	191.38	653.92	334.72	414.23
Average improvement of NSGA-II						**71.89%**	**55.32%**	**53.11%**

Table 11. Deterministic NSGA-II results for the instances used for the stochastic environment.

Instance				NSGA-II	
Number of jobs	**Number of stages**	**Machines per stage**	**Shop condition**	**Tardiness**	**Time (s)**
10	2	2	Low	136	19.03125
10	3	2	Low	424	25.4375
10	2	2	High	588	14.265625
10	3	2	High	1656	25.421875
20	2	2	Low	241	34.6875
20	3	2	Low	475	50.28125
20	2	2	High	2299	34.765625
20	3	2	High	2711	51.59375

These results prove the efficiency of the proposed algorithm for both objective functions against all three compared methods. The standard deviation of tardiness is improved for all cases, showing the robustness of our solutions. Tables

7 and 9 evidence that ENS2 obtains the best results for total tardiness, and also that our NSGA-II improves such solutions in 11.96% and 24.62% on average for job size 10 and 20, respectively. Also, NEHedd heuristic presents the best results for standard deviation of total tardiness, and our algorithm improves these solutions 57.89% and 55.32% on average for both job sizes, respectively. Low shop conditions showed better results for both objectives compared to high shop conditions, which is understandable because of the reduced variability in the former. The same happens with the CV, where it is shown that when its value increases, the results tend to be higher because there is more variability of processing and release times. Finally, results show better performance when uncertainty is modelled through lognormal probability distribution.

Regarding the results of Table 11, it is shown that our algorithm performs better in a stochastic scenario in comparison with the deterministic scenario when high shop condition is considered, while there does not exist a significant difference when using low shop condition between both scenarios. These results prove that our algorithm provides robust solutions, being relatively insensitive to disruptions.

5.3. Pareto fronts for bi-objective stochastic problem

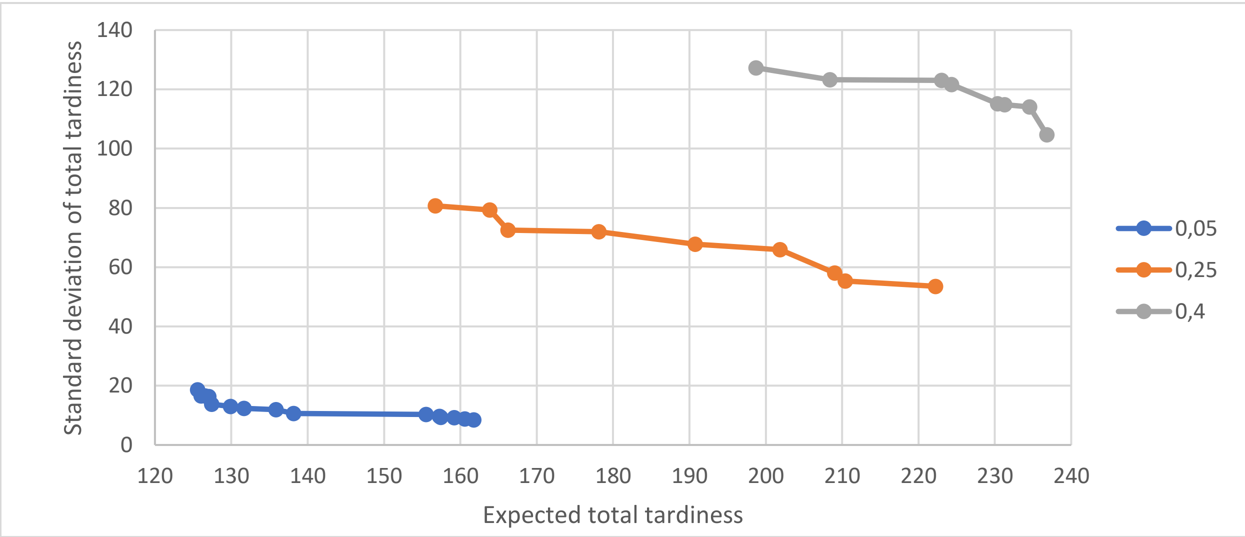

Figure 7. Pareto frontiers for instance 10_2_2_0.2_0.2_10_Rep0 for uniform distribution.

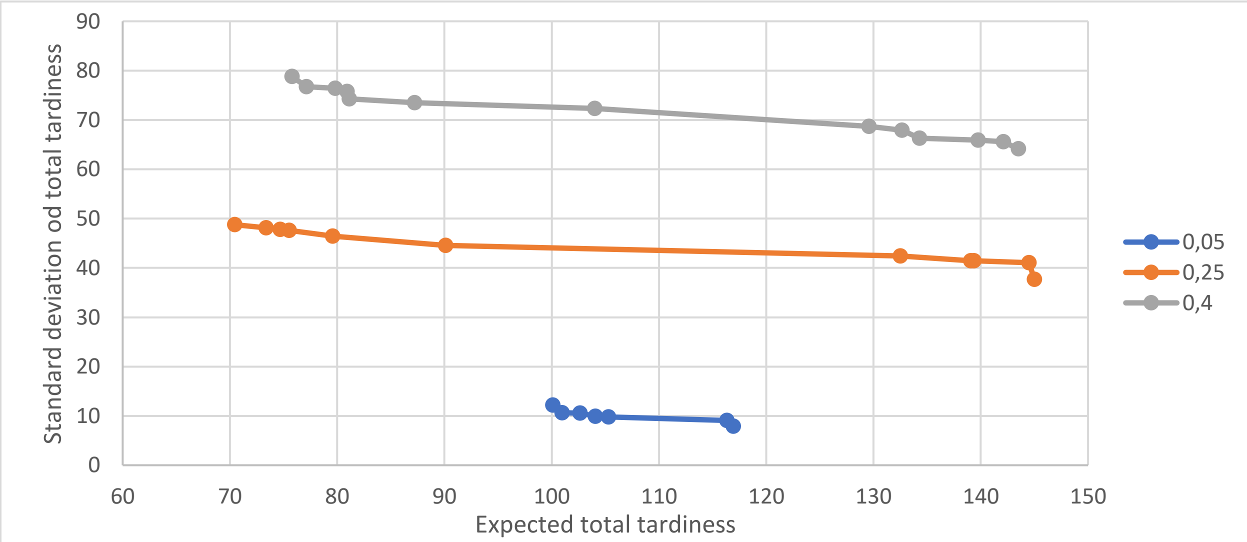

Figure 8. Pareto frontiers for instance 10_2_2_0.2_0.2_10_Rep0 for lognormal distribution.

48 Pareto frontiers were obtained. One frontier was created for each combination of an instance, a probability distribution (uniform or lognormal), and a CV (0.05, 0.25, 0.4) of processing and release times. Figures 7 and 8 present the six Pareto frontiers for the instance 10_2_2_0.2_0.2_10_Rep0 when both uncertain parameters are generated from uniform and lognormal distributions, respectively. In this example, an inverse relation can be observed between the CV and the number of Pareto frontiers for the uniform distribution, since less frontiers are generated while increasing the CV value. For the lognormal distribution, the relation is direct, obtaining more frontiers while increasing the CV value. Both graphics use the same scale to make a proper comparison between the results of both probability distributions. In this case, it can be seen than the results for lognormal distribution are lower than those of the uniform distribution. However, experimental designs are presented in the next subsection to conclude statistically about the behavior of these results.

5.4. Experimental design

Two additional generalized factorial designs in RStudio IDE were carried out to determine if there is an effect of the considered factors in the total tardiness and its standard deviation. These factors are presented next with their respective levels: number of jobs (10, 20), number of stages (2, 3), shop condition (Low, High), probability distribution (uniform, lognormal), and CV (0.05, 0.25, 0.4). Number of jobs and number of stages were included as blocking factors, because it is assumed both represent a source of variability but their effect and interaction with other factors are not of primary interest in this experiment. There are 48 treatments in total and one observation for each of them, having a total of 48 observations for each experiment. Each observation is taken from the extreme points of the Pareto frontier generated by executing our NSGA-II algorithm with specific levels of each factor as input parameters.

The resulting ANOVAs for both experiments are summarized in Table 11. Considering a significance of 5%, results showed that the number of jobs, stages, shop condition and probability distribution have a significant effect on the expected total tardiness of a solution. Additionally, the number of jobs, shop condition, probability distribution, coefficient of variation, as well as the effect between the interaction of shop condition with CV, and CV with probability distribution, were proven to be statistically significant on the standard deviation of a solution.

Before being able to make any conclusions on the effect of these factors in the obtainment of quality and robust solutions for the stated problem, the experiments must fulfill three assumptions: normality, homoscedasticity, and independence. If these assumptions were not accomplished, a non-parametric test had to be performed to validate the results of the ANOVA.

Table 11. ANOVA for the expected total tardiness and standard deviation of total tardiness.

Factor	Expected total tardiness	Standard deviation of total tardiness
Jobs	**1.48e-06**	**3.73e-07**
Stages	**4.64e-06**	0.0623
Shop Condition	**3.37e-11**	**2.15e-06**
PD	**0.0386**	**0.0007**
CV	0.9415	**2.98e-14**
Shop Condition:PD	0.5585	0.8356
Shop Condition:CV	0.9681	**0.0211**
PD:CV	0.4806	**0.0125**
Shop Condition:PD:CV	0.9204	0.9731
R^2	82.7%	89.4%
R^2_{adj}	76.1%	85.3%

Tables 12 and 13 show the results obtained for the assumptions evaluation. The p-value of the performed tests must be higher than 0.05 for the assumption to be satisfied. Taking the previous statement into account, it can be observed that both experiments fulfilled independence of residuals. However, normality was only satisfied by the standard deviation model and homoscedasticity was met only in the expected total tardiness model. Given the results of the assumption tests, two ATS tests were performed to study the effect of the factors of interest on the experiments.

Table 12. Assumptions evaluation for the expected total tardiness experiment.

Assumption	Test	p-value
Normality	Shapiro-Wilk	7.805e-05
	Anderson-Darling	1.703e-06
Homoscedasticity	Levene	**0.6083**
Independence	Durbin-Watson	**0.206**

Table 13. Assumptions evaluation for the standard deviation of total tardiness experiment.

Assumption	Test	p-value
Normality	Shapiro-Wilk	**0.5173**
	Anderson-Darling	**0.2835**
Homoscedasticity	Levene	2.2e-16
Independence	Durbin-Watson	**0.364**

As shown in Table 14, ATS results validated those obtained in the previous ANOVAs, since they showed that the number of jobs, shop condition and probability distribution have a statistically significant effect on both objective functions. Additionally, and as expected, the effect of the coefficient of variation is also significant on the standard deviation of total tardiness of a solution. The effects of the interaction between factors did not reach the desired level of significance in neither experiment.

Table 14. ANOVA-Type Statistics test for the expected total tardiness and standard deviation of total tardiness.

Factor	Expected total tardiness	Standard deviation of total tardiness
Jobs	**0.0001**	**0.0000**
Shop Condition	**0.0000**	**0.0000**
PD	**0.0103**	**0.0006**
CV	0.9552	**0.0000**
Shop Condition:PD	0.6411	0.1642
Shop Condition:CV	0.9188	0.2797
PD:CV	0.4205	0.0983
Shop Condition:PD:CV	0.9687	0.5892

For a better understanding of the previous results, the means of the observations at each level of the factors for the expected value and standard deviation of total tardiness are shown in Figures 9 and 10, respectively. Apart from factor CV in Figure 9, all plotted lines show a pronounced slope that represent the variation of both objective functions in each factor. These same factors proved to have a statistically significant effect on the response variables according to the ATS test, allowing to deepen the analysis using visual representations of the effects.

As expected, a higher CV will lead to a higher standard deviation of the objective, and solutions obtained under high shop conditions will have a worse value of total tardiness than those under low shop conditions. Additionally, it was found that high shop conditions also increase the standard deviation of the solutions, meaning that the presence of tight due dates in scheduling environments also make more difficult the obtainment of robust solutions. The distribution probability selected to model the uncertainty will also show very different results, with lognormal distribution having lower values than uniform distribution for both objectives. This effect also reflects on the importance of having an adequate probability distribution fitting when trying to obtain precise results that lead to suitable decision making.

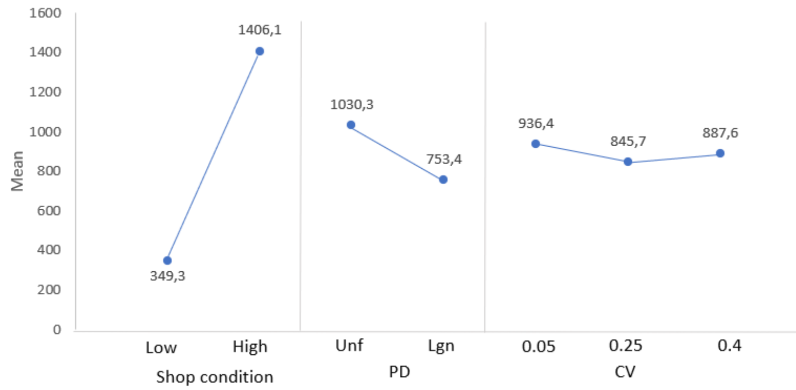

Figure 9. Main effects plot for Expected total tardiness.

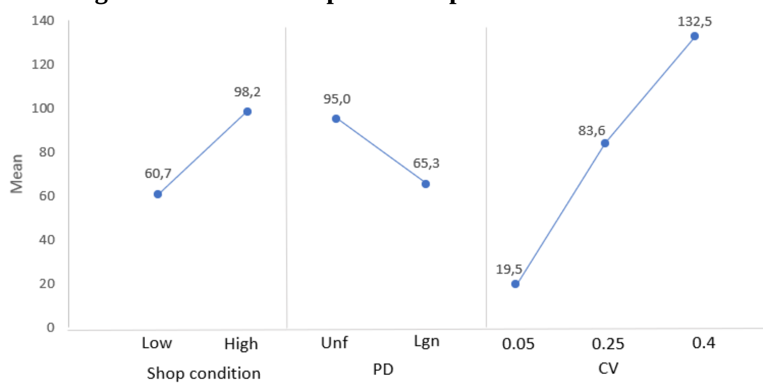

Figure 10. Main effects plot for Standard deviation of total tardiness.

For this work three indicators were presented to measure the quality of Pareto solutions. These indicators are the ones presented in Karimi, Zandieh, & Karamooz, (2010): Number of Pareto optimal solutions (NPS) , Mean ideal distance (MID) that presents the closeness between Pareto solutions and ideal point (0,0) and Spread of non-dominance solutions (SNS) that measures the diversity across the Pareto frontier. Tables 15 through 18 show the results for each indicator for each combination of jobs, stages, shop conditions, probability distribution and CV. As it was expected, the MID and SNS tend to increase as the number of jobs and number of stages also increase. Additionally, it can be observed that with the uniform probability distribution when the CV increases the MID also increases, but with the lognormal probability distribution this indicator tends to decrease.

Table 15. Indicators for 10 jobs and 2 stages.

Number of jobs	Stages	Shop condition	PD	CV	MID	NPS	SNS
10	2	Low	Uniform	0.05	141.94	15	14.25
				0.25	201.07	9	17.82
				0.4	252.93	8	8.37
			Lognormal	0.05	107.10	7	6.41
				0.25	115.79	11	27.88
				0.4	131.07	13	20.28
		High	Uniform	0.05	748.86	9	4.63
				0.25	807.06	14	21.23
				0.4	862.86	9	33.64
			Lognormal	0.05	713.03	17	6.42
				0.25	658.29	11	14.16
				0.4	637.75	7	22.31

Table 16. Indicators for 10 jobs and 3 stages.

Number of jobs	Stages	Shop condition	PD	CV	MID	NPS	SNS
10	3	Low	Uniform	0.05	285.95	14	21.34
				0.25	338.97	6	30.23
				0.4	405.89	6	18.12
			Lognormal	0.05	248.46	10	13.73
				0.25	228.03	9	20.89
				0.4	235.91	16	17.05
		High	Uniform	0.05	1190.14	4	23.64
				0.25	1243.10	5	66.57
				0.4	1268.53	5	31.97
			Lognormal	0.05	1141.47	7	7.07
				0.25	1033.56	5	12.77
				0.4	975.53	6	9.28

Table 17. Indicators for 20 jobs and 2 stages.

Number of jobs	Stages	Shop condition	PD	CV	MID	NPS	SNS
20	2	Low	Uniform	0.05	357.48	9	42.74
				0.25	461.93	8	38.53
				0.4	547.00	8	13.02
			Lognormal	0.05	293.38	11	46.13
				0.25	234.30	13	42.65
				0.4	219.15	14	32.39
		High	Uniform	0.05	1293.60	16	93.17
				0.25	1462.28	6	116.74
				0.4	1614.62	10	63.28
			Lognormal	0.05	1221.79	10	74.44
				0.25	1000.81	8	45.25
				0.4	1040.71	3	80.84

Table 18. Indicators for 20 jobs and 3 stages.

Number of jobs	Stages	Shop condition	PD	CV	MID	NPS	SNS
20	3	Low	Uniform	0.05	768.13	5	34.70
				0.25	862.66	6	24.97
				0.4	1035.72	5	29.21
			Lognormal	0.05	649.46	12	24.24
				0.25	520.90	4	40.96
				0.4	542.45	7	38.18
		High	Uniform	0.05	2793.34	7	32.97
				0.25	2914.13	4	23.30
				0.4	3213.02	4	23.46
			Lognormal	0.05	2710.39	7	55.18
				0.25	2310.55	7	45.55
				0.4	2196.50	7	42.98

6. Limitations, Conclusions and Recommendations

This paper aims to obtain robust solutions of a stochastic HFS scheduling problem by minimizing expected total tardiness and standard deviation of total tardiness. To solve the problem, a simheuristic approach that hybridizes a NSGA-II algorithm with a Monte Carlo Simulation process is proposed. The aim was to obtain Pareto frontiers for the expected total tardiness and its standard deviation as a robustness measure for the stated problem under uncertain processing and release times.

First, an experimental design was carried out to select the parameters that showed the best performance for the NSGA-II. Results show that in most instances, the best configuration for the tested parameters is $P_{size} = 60$, crossover probability = 0.8, mutation probability = 0.1, and number of generations = 100. Then, an evaluation of performance of the NSGA-II was made by testing 8 random deterministic benchmarked instances for jobs size 6, 8, 10 and 15 to compare its results in contrast to those of the MILP model, ERD, NEHedd, and ENS2 methods. These results proved that the proposed algorithm outperforms the MILP model solutions by 1.21% and it also improved the solutions of all three methods while using a reasonable computational time. Afterwards, the performance of the proposed NSGA-II was compared against ERD, NEHedd, and ENS2 methods in two instances of 10 jobs and two instances of 20 jobs. Results show that our algorithm improves the quality of the solutions for both objective functions in comparison with the three mentioned methods in a reasonable computational time. With these results we can conclude that our algorithm does provide robust quality solutions for the HFS problem.

Complementary, two experiments were carried out to study the influence of the shop condition, probability distribution and coefficient of variation on the objective functions. The three factors were proven to be highly influential on the variability of the results, with the only exception being the coefficient of variation having no statistically significant effect on the total tardiness.

With the obtained results with our algorithm, we can conclude that these solutions are very good for FFS environment. This shop environment can be replicated in different industries, especially in manufacturing shops. The main advantage that the implementation of our algorithm provides when using it is the small sensitivity when facing disruptions, which is a common situation in real life scenarios. For instance, if a machine is represented by an employee in real life, it is implicit that the processing time of each person will be different. Regarding release times, in real life enterprises are subject to the delivery times of different suppliers, traffic inside the city that may delay orders, order releases, and other factors that can not always be controlled. Hence the relevance of our algorithm in real life situations.

As future work, we recommend testing the effect of the previously mentioned factors and their interactions on larger instances and with different levels, since further experimentation can lead to a better understanding of the problem. We also recommend using different CV values for each uncertain parameter to evaluate the effect of these combinations on the schedule. Also, varying the processing times between the machines of each stage would be interesting since it would represent a closer approach to reality. Additionally, it would be interesting to include total earliness and standard deviation of total earliness as objectives, expanding the scope of this work to study robust Just in Time strategies.

7. Attachements

8. References

Appendix A: Practical Processing Time Distributions. (2018, September 28). *Principles of Sequencing and Scheduling*, pp. 571–595. https://doi.org/https://doi.org/10.1002/9781119262602.app1

Azizoğlu, M., Çakmak, E., & Kondakci, S. (2001). A flexible flowshop problem with total flow time minimization. *European Journal of Operational Research*, *132*(3), 528–538. https://doi.org/https://doi.org/10.1016/S0377-2217(00)00142-9

Borgonovo, E., & Plischke, E. (2016). Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, *248*(3), 869–887. https://doi.org/https://doi.org/10.1016/j.ejor.2015.06.032

Bozorgirad, M., & Logendran, R. (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. *International Journal of Advanced Manufacturing Technology*, *83*(5–8), 1135–1151. https://doi.org/10.1007/s00170-015-7650-9

Chaari, T., Chaabane, S., Loukil, T., & Trentesaux, D. (2011). A genetic algorithm for robust hybrid flow shop scheduling. *International Journal of Computer Integrated Manufacturing*, *24*(9), 821–833. https://doi.org/10.1080/0951192X.2011.575181

Cui, W., Lu, Z., Li, C., & Han, X. (2018). A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops. *Computers and Industrial Engineering*, *115*(February 2017), 342–353. https://doi.org/10.1016/j.cie.2017.11.020

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Erdem, E., Qu, X., & Shi, J. (2012). Rescheduling of elective patients upon the arrival of emergency patients. *Decision Support Systems*, *54*(1), 551–563. https://doi.org/https://doi.org/10.1016/j.dss.2012.08.002

Fazayeli, M., Aleagha, M. R., Bashirzadeh, R., & Shafaei, R. (2016). A hybrid meta-heuristic algorithm for flowshop robust scheduling under machine breakdown uncertainty. *International Journal of Computer Integrated Manufacturing*, *29*(7), 709–719. https://doi.org/10.1080/0951192X.2015.1067907

Feng, X., Zheng, F., & Xu, Y. (2016). Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *International Journal of Production Research*, *54*(12), 3706–3717. https://doi.org/10.1080/00207543.2016.1162341

Fernandez-Viagas, V., & Framinan, J. M. (2015). NEH-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness. *Computers and Operations Research*, *60*(August), 27–36. https://doi.org/10.1016/j.cor.2015.02.002

Framinan, J. M., & Perez-Gonzalez, P. (2015). On heuristic solutions for the stochastic flowshop scheduling problem. *European Journal of Operational Research*, *246*(2), 413–420. https://doi.org/10.1016/j.ejor.2015.05.006

González-Neira, E. M., Montoya-Torres, J. R., & Caballero-Villalobos, J. P. (2019). A comparison of dispatching rules hybridised with Monte Carlo Simulation in stochastic permutation flow shop problem. *Journal of Simulation*, *13*(2), 128–137. https://doi.org/10.1080/17477778.2018.1473908

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

Goren, S., & Sabuncuoglu, I. (2010). Optimization of schedule robustness and stability under random machine breakdowns and processing time variability. *IIE Transactions (Institute of Industrial Engineers)*, *42*(3), 203–220. https://doi.org/10.1080/07408170903171035

Gupta, J. N. D. (1988). Two-stage, hybrid flowshop scheduling problem. *Journal of the Operational Research Society*, *39*(4), 359–3641. https://doi.org/10.1057/jors.1988.63

Haouari, M., & Hidri, L. (2008). On the hybrid flowshop scheduling problem. *International Journal of Production Economics*, *113*(1), 495–497. https://doi.org/10.1016/j.ijpe.2007.10.007

Hazır, Ö., & Ulusoy, G. (2020). A classification and review of approaches and methods for modeling uncertainty in projects. *International Journal of Production Economics*, *223*, 107522. https://doi.org/https://doi.org/10.1016/j.ijpe.2019.107522

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, *165*(2), 289–306. https://doi.org/https://doi.org/10.1016/j.ejor.2004.04.002

Karimi, N., Zandieh, M., & Karamooz, H. R. (2010). Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert Systems with Applications*, *37*(6), 4024–4032. https://doi.org/10.1016/j.eswa.2009.09.005

Kim, Y. D., Lim, H. G., & Park, M. W. (1996). Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. *European Journal of Operational Research*, *91*(1), 124–143. https://doi.org/10.1016/0377-2217(95)00119-0

Lee, J., Yoon, Y. S., & Park, K. (2019). A Scheduling Algorithm for On-Time Production in A Hybrid Flow Shop Manufacturing Process. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 1243–1247. https://doi.org/10.1109/ICTC46691.2019.8939899

Liao, L., Sarin, S. C., & Sherali, H. D. (2012). A scenario generation-based lower bounding approach for stochastic scheduling problems. *Journal of the Operational Research Society*, *63*(10), 1410–1420. https://doi.org/10.1057/jors.2011.150

Liu, F., Wang, S., Hong, Y., & Yue, X. (2017). On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions. *IEEE Transactions on Engineering Management, Engineering Management, IEEE Transactions on, IEEE Trans. Eng. Manage.*, *64*(4), 539–553. https://doi.org/10.1109/TEM.2017.2712611

Liu, M., & Liu, X. (2019). Machine scheduling with stochastic release and processing times. *IFAC-PapersOnLine*, *52*(13), 2116–2121. https://doi.org/10.1016/j.ifacol.2019.11.518

Lodree, E., Jang, W., & Klein, C. M. (2004). A new rule for minimizing the number of tardy jobs in dynamic flow shops. *European Journal of Operational Research*, *159*(1), 258–263. https://doi.org/10.1016/S0377-

2217(03)00404-1
Long, J., Sun, Z., Pardalos, P. M., Bai, Y., Zhang, S., & Li, C. (2020). A robust dynamic scheduling approach based on release time series forecasting for the steelmaking-continuous casting production. *Applied Soft Computing Journal*, *92*, 106271. https://doi.org/10.1016/j.asoc.2020.106271
Luo, J., & El Baz, D. (2018). A Survey on Parallel Genetic Algorithms for Shop Scheduling Problems. *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2018 IEEE International, IPDPSW*, pp. 629–636. https://doi.org/10.1109/IPDPSW.2018.00103
Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flowshop scheduling problems. *Computers and Industrial Engineering*, *30*(4), 1061–1071. https://doi.org/10.1016/0360-8352(96)00053-8
Naderi, B., Gohari, S., & Yazdani, M. (2014). Hybrid flexible flowshop problems: Models and solution methods. *Applied Mathematical Modelling*, *38*(24), 5767–5780. https://doi.org/10.1016/j.apm.2014.04.012
Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9
Öztop, H., Tasgetiren, M. F., Kandiller, L., Eliiyi, D. T., & Gao, L. (2020). Ensemble of metaheuristics for energy-efficient hybrid flowshops: Makespan versus total energy consumption. *Swarm and Evolutionary Computation*, *54*(January). https://doi.org/10.1016/j.swevo.2020.100660
Pan, Q. K., Ruiz, R., & Alfaro-Fernández, P. (2017). Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows. *Computers and Operations Research*, *80*, 50–60. https://doi.org/10.1016/j.cor.2016.11.022
Rahmani, D. (2017). A new proactive-reactive approach to hedge against uncertain processing times and unexpected machine failures in the two-machine flow shop scheduling problems. *Scientia Iranica*, *24*, 1571–1584. https://doi.org/10.24200/sci.2017.4136
Rahmani, D., & Heydari, M. (2014). Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times. *Journal of Manufacturing Systems*, *33*(1), 84–92. https://doi.org/10.1016/j.jmsy.2013.03.004
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
Rahmani, D., & Ramezanian, R. (2016). A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study. *Computers and Industrial Engineering*, *98*, 360–372. https://doi.org/10.1016/j.cie.2016.06.018
Rajkumar, R., & Shahabudeen, P. (2009). Bi-criteria improved genetic algorithm for scheduling in flowshops to minimise makespan and total flowtime of jobs. *International Journal of Computer Integrated Manufacturing*, *22*(10), 987–998. https://doi.org/10.1080/09511920902993353
Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, *205*(1), 1–18. https://doi.org/10.1016/j.ejor.2009.09.024
Sabuncuoglu, I., & Goren, S. (2009). Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing*, *22*(2), 138–157. https://doi.org/10.1080/09511920802209033
Sahar Tadayonirad, author, Hany Seidgar, author, Hamed Fazlollahtabar, author, & Rasoul Shafaei, author. (2019). Robust scheduling in two-stage assembly flow shop problem with random machine breakdowns: integrated meta-heuristic algorithms and simulation approach. *Assembly Automation*, *39*(5), 944–962. https://doi.org/10.1108/AA-10-2018-0165
Sevaux, M., & Sörensen, K. (2004). A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates. *4or*, *2*(2), 129–147. https://doi.org/10.1007/s10288-003-0028-0
Srinivas, N., & Deb, K. (1994). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, *2*(3), 221–248. https://doi.org/10.1162/evco.1994.2.3.221
Tseng, C.-T., Liao, C.-J., & Liao, T.-X. (2008). A note on two-stage hybrid flowshop scheduling with missing operations. *Computers & Industrial Engineering*, *54*(3), 695–704. https://doi.org/https://doi.org/10.1016/j.cie.2007.09.005
Werner, F. (2011). GENETIC ALGORITHMS FOR SHOP SCHEDULING PROBLEMS: A SURVEY. *Preprint Series*, *11*, 1–66.

Yin, Y., Stecke, K. E., Swink, M., & Kaku, I. (2017). Lessons from seru production on manufacturing competitively in a high cost environment. *Journal of Operations Management*, *49–51*, 67–76. https://doi.org/https://doi.org/10.1016/j.jom.2017.01.003

Yue, F., Song, S., Zhang, Y., Gupta, J. N. D., & Chiong, R. (2018). Robust single machine scheduling with uncertain release times for minimising the maximum waiting time. *International Journal of Production Research*, *56*(16), 5576–5592. https://doi.org/10.1080/00207543.2018.1463473

Zandieh, M., Dorri, B., & Khamseh, A. R. (2009). Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. *International Journal of Advanced Manufacturing Technology*, *43*(7–8), 767–778. https://doi.org/10.1007/s00170-008-1740-x

Zuo, X., Mo, H., & Wu, J. (2009). A robust scheduling method based on a multi-objective immune algorithm. *Information Sciences*, *179*(19), 3359–3369. https://doi.org/10.1016/j.ins.2009.06.003