RAIRO-Oper. Res. 55 (2021) S2125–S2159
https://doi.org/10.1051/ro/2020077

**RAIRO Operations Research**
www.rairo-ro.org

# SOFT-COMPUTING APPROACHES FOR RESCHEDULING PROBLEMS IN A MANUFACTURING INDUSTRY

Jaime Acevedo-Chedid[1], Jennifer Grice-Reyes[1], Holman Ospina-Mateus[1], Katherinne Salas-Navarro[2], Alcides Santander-Mercado[3] and Shib Sankar Sana[4,*]

**Abstract.** Flexible manufacturing systems as technological and automated structures have a high complexity for scheduling. The decision-making process is made difficult with interruptions that may occur in the system and these problems increase the complexity to define an optimal schedule. The research proposes a three-stage hybrid algorithm that allows the rescheduling of operations in an FMS. The novelty of the research is presented in two approaches: first is the integration of the techniques of Petri nets, discrete simulation, and memetic algorithms and second is the rescheduling environment with machine failures to optimize the makespan and Total Weighted Tardiness. The effectiveness of the proposed Soft computing approaches was validated with the bottleneck of heuristics and the dispatch rules. The results of the proposed algorithm show significant findings with the contrasting techniques. In the first stage (scheduling), improvements are obtained between 50 and 70% on performance indicators. In the second stage (failure), four scenarios are developed that improve the variability, flexibility, and robustness of the schedules. In the final stage (rescheduling), the results show that 78% of the instances have variations of less than 10% for the initial schedule. Furthermore, 88% of the instances support rescheduling with variations of less than 2% compared to the heuristics.

**Mathematics Subject Classification.** 90B35, 90B36, 68M20.

Received March 2, 2020. Accepted July 4, 2020.

## 1. Introduction

Competition forces organizations to act quickly to preserve their position in the market Flexible manufacturing systems (FMS) allow the production of a variety of products to meet demand. FMS are the result of technological innovations generated through the years, where these problems are classified as highly complex (NP-Hard) [13]. The shop floor is a dynamic environment affected by the arrival of new activities, customer, and materials. The dynamic factors to consider are machine breakdowns, absenteeism from work, the arrival of new orders, and a change in the priority of work. Immediately consideration of dynamic factors in production is

*Keywords.* Flexible manufacturing system, scheduling, reactive scheduling, Petri net, memetics algorithm.

[1] Department of Industrial Engineering, Universidad Tecnológica de Bolívar, Cartagena, Colombia.
[2] Department of Productivity and Innovation, Universidad de la Costa, Barranquilla, Colombia.
[3] Department of Industrial Engineering, Universidad del Norte, Barranquilla, Colombia.
[4] Kishore Bharati Bhagini Nivedita College, Behala, Kolkata 700060, India.
*Corresponding author: shibsankarsana@gmail.com, shib_sankar@yahoo.com

known as reactive or real-time schedules [34]. Seeking the optimization of manufacturing processes, inventories and manufacturing costs has a significant impact on the organizational management of a company, and these have impacts on the decision-making process [10,15,16,23,32,48]. The plant floor will always require optimized tasks, to better meet all customers' requirements [40,41,47]. The processes are aimed to be sustainable, efficient, and profitable [36,50,56]. Customers will recognize the effective management of an organization by factors such as quality, time, opportunity, and economy in goods and services [24,51,59].

The research proposes a three-stage hybrid algorithm that allows the rescheduling of operations in an FMS. The novelty of the research is presented in two approaches: One is integration of the techniques of Petri nets, discrete simulation, and memetic algorithms and second is rescheduling environment with machine failures to optimize the makespan and Total Weighted Tardiness. The hybrid algorithm was called "PetNMA". The PetNMA algorithm is composed of three (3) sub algorithms. The first performs the initial scheduling of the jobs. The second simulates the initial scheduling until the machine fails, so a rescheduling is required. The third generates a reactive scheduling through the application of a memetic algorithm. The proposed algorithm is validated with problems of the FMS library. The results of the initial scheduling are compared with the bottleneck and dispatch rules, and the reactive scheduling was compared with the dispatch rules.

The model has a novel proposal that integrates the Petri net technique as an effective method for production scheduling. The innovative component of the proposed model has two key elements. First integrates the Petri nets with a genetic algorithm to develop active initial scheduling and second integrates the use of memetic algorithms with Petri nets for a production rescheduling environment. The integration of Petri net starts from the generation of active schedules and their configuration when the machine failure occurs. This research is considered a pioneer in a memetic algorithm decoded by Petri nets in an environment of rescheduling when there are failures or breakdowns of machines.

The proposed model establishes a modification of the libraries with different ranges of scenarios. The failure of the machine is considered in four instances with the fault simulations in different periods. This analysis identifies the flexibility and robustness of production schedules. This paper is organized as follows: the theoretical framework and literature review are in Section 2. In Section 3, the applied methods are described, such as Petri nets, genetic algorithms, memetic algorithms, simulated annealing, and local search. In Section 4, the algorithm "PetNMA" is presented and explained. In Section 5, the results obtained with the "PetNMA" algorithm. The findings, conclusions, and discussion are in Section 6.

## 2. Background and literature review

Production scheduling is the assignment and sequence of jobs to resources for the manufacture of goods or services [45]. Production schedule are aimed at optimizing deadlines, delivery times, setups, inventories, and the use of machines [45]. There are many methods and strategies used to develop a production schedule [5]. Among the basic and efficient strategies are the well-known dispatch rules. These consider the allocation and sequencing based on delivery dates, processing time, machine loading, among other strategies, as follows: Shortest Processing Time (SPT), Largest Processing Time (LPT), Most Remaining Work Time (MRWT), Earliest Due Date (EDD), Critical Ratio (CR), Apparent Tardiness Cost (ATC), and Minimum Slack (MSLACK). The objectives are used to evaluate the most important production schedule are: Makespan, Maximum Tardiness, Total Tardiness, Total Weighted Tardiness, Total Flow, and Total Weighted Flow [45].

### 2.1. Flexible manufacturing system (FMS)

An FMS is a system with a technological and automated component that can produce a variety of operations and tasks for production. The route of a job within the system is flexible and is taken in the scheduling process. An FMS be a flexible job shop with an additional number of restrictions [44]. The flexibility within a production system allows us to diversify products, anticipate demand, forecast inventory, increase the efficiency and quality of processes [28,49,53].

Low and Wu [31] symbolically represented the FMS as a set of machines $M^T = \{m_1, m_2, \ldots, m_M\}$ and a set of jobs $J = \{J_1, J_2, \ldots, J_N\}$, where each job $J_j, \forall_j \in \{1, 2, \ldots, N\}$ has a delivery date $d_j$ promised and it consists of a sequence of $n_j$ operations. Each operation $O_{i,j}, \forall i \in \{1, 2, \ldots, N\}$ and $\forall j \in \{1, 2, \ldots, n_i\}$, it can be processed continuously on any machine $m_k$ of set of machines $M^S, (M^S \subseteq M^T)$ in a processing time $p_{i,j,k}$, it is including setup time on the machine. Each machine $m_k$, $\forall k \in \{1, 2, \ldots, M\}$ can process only one operation. It is assumed that all $p_{i,j,k}$, $d_j$ and $n_j$ are deterministic as well as the jobs available $(r_j = 0, \forall j)$. From an Operations Research perspective, scheduling in FMS is a more complex version of the classic flexible scheduling problem, which is known to be NP-hard [13]. This complexity is caused by the versatility, flexibility, and alternate routes in the operations. An FMS contains many variables and restrictions that change over time, and these characteristics justify the use of dynamic scheduling [28].

## 2.2. Reactive scheduling

Random events involve the rescheduling of production, which affects initial scheduling and the level of customer service. This action is known as reactive scheduling [11]. The dynamic production environment is exposed to constant adjustments. Among them: machine failures, the arrival of new jobs, change of priorities, job cancellation, supplier failures, low-quality materials, absenteeism, and variability in jobs (processing times or delivery dates), changes in due dates. Most of the approaches to reactive scheduling or rescheduling, had been based on the generation of a basic predictive schedule. The three most common methods for rescheduling are: regeneration, partial rescheduling, and right-shift scheduling [60]. Regeneration includes all operations. Partial rescheduling only considers affected operations. The right shift method postpones the remaining operations; This method leads to more stable schedules.

## 2.3. Machine failure or breakdowns

One of the most frequent random events is machine failure or breakdown. This event may require two types of rescheduling. Total repair or reprocessing of the schedule. Repair refers to the local configuration of the current schedule, while the total rescheduling of a new schedule is generated from the base. To address rescheduling, machine failures must determine which jobs have been completed and which are affected. A job is called "complete" if it is completed before the failure. A task is considered "affected" if it needs to be relocated due to the interruption. Reactive scheduling is based on the affected jobs [19].

In a manufacturing system, the unavailability of a machine can be known in advance and better managed, compared to when an unexpected failure occurs. A robust schedule is one that can contain interrupts, such as events that were considered in their original planning [2]. The probability of failure of the machines is quantified by the time the machine has been occupied divided by the total time. A machine with higher occupancy is more likely to fail. If a manufacturing system has historical data to provide an approximate distribution of machine failures, it can be used to predict the time of machine failure and repair time to obtain a robust and stable predictive schedule [33].

## 2.4. Literature review

The literary review synthesizes the previous contributions of authors in the field of flexible manufacturing systems and their scope in reactive scheduling or rescheduling. In summary, Table 1 outlines all these findings.

Hatono *et al.* [20] used a Stochastic Petri Nets to describe the uncertain events of stochastic behavior in the FMS, such as machine failures, repair time, and processing time. Cho [8] developed a Petri net model for message manipulation and event monitoring in an FMS cell. ElMaraghy and Elmekkawy [12] proposed scheduling algorithm that used Petri nets to deal with machine faults in real-time. Chen and Chen [6] simulated a Petri net in its method for non-hierarchical control for the performance of an FMS. Chen and Chen [7] developed an algorithm for the rescheduling of random events such as machine failures in an FMS. Acevedo and Mejía [1] carried out a Genetic Algorithm for reactive scheduling in FMS with arrivals of new jobs and priority changes.

TABLE 1. Contributions of previous authors.

| Author(s) | Production system | Scheduling strategy | Rescheduling strategy | Event | Objective(s) | Petri nets | Heuristic(s) |
|---|---|---|---|---|---|---|---|
| Hatono *et al.* [20] | FMS | Petri nets | Hierarchical structures | Machine failures, repair and, processing time | Total time | Stochastics | FMS simulation |
| Cho [8] | Shop Floor Control System (SFCS) | Petri nets | × | × | Equipment, workstation, and shop | Interpreted petri nets | Hierarchically decomposed |
| ElMaraghy and ElMekkawy [12] | FMS | Petri nets | Deadlock-free | Machine breakdowns | Total time | Timed and minimal siphons | × |
| Chen and Chen [6] | FMS | Petri nets | Client-server paradigm | Object-oriented approach | Total time | Colored Petri net simulation | × |
| Chen and Chen [7] | FMS | Adaptive scheduling | Rolling horizon | Machine failures, repair time | Completion time | × | Markov process |
| Kumar *et al.* [27] | FMS | Fuzzy-based | Operation machine allocation vector | Machine loading problem | System imbalance, throughput | Extended neuro-fuzzy Petri net | × |
| Acevedo and Mejía [1] | FMS | Petri nets | Reactive schedule | Machine breakdowns, new jobs | Total time | Timed Petri nets | Memetic algorithm |
| Mejia and Acevedo [34] | FMS | Reactive scheduling | Simulation | Machine breakdowns, new jobs | Total time | Timed Petri nets | Genetic algorithm |
| Tanimizu *et al.* [54] | FMS | Reactive scheduling | Based reactive | Delay and new jobs | Total time | × | Genetic algorithms |
| Kim *et al.* [25] | FMS | Reactive scheduling | RTA* and rule-based supervisor | Due date | Total time, tardiness | Timed Petri net | A reactive fast graph |
| Tashnizi *et al.* [55] | FMS | × | × | Sharing resources and processing times | Total time | Petri net | × |
| Tüysüz and Kahraman [58] | FMS | Petri nets fuzzy sets | × | Uncertainty in system | Time-critical | Stochastic | Fuzzy mathematics |
| Zhao *et al.* [61] | FMS | Petri nets | × | Discrete events | Total time | Timed Petri net | Genetic algorithm |
| Patel and Joshi [43] | FMS | Petri nets, simulation deadlock | × | × | Throughput, completion time | Stochastic | × |
| Han *et al.* [17] | FMS | Petri nets, deadlock | × | × | Makespan | Timed petri net | Genetic algorithm |
| Baruwa *et al.* [4] | FMS | Petri nets, deadlock-free | × | × | Makespan | Timed colored Petri net | Anytime heuristic search algorithm |
| Başak and Albayrak [3] | FMS | Petri nets, real-time scheduling | × | × | Timed marked graph | Object-oriented Petri nets | Artifex PN |
| Han *et al.* [18] | FMS | Petri nets, deadlock | × | Lot sizes, resource capacities, and routing flexibility | Makespan | Timed Petri net | Simulated annealing, swarm optimization |
| Li *et al.* [30] | FMS | Petri nets | × | Available time of shared resources | Makespan | Transition-timed Petri nets | Heuristic search function |
| Lei *et al.* [29] | FMS | Petri nets, deadlock | × | Deadlock-free | Makespan | Timed Petri nets | Heuristic search strategies |
| Mejía and Niño [35] | FMS | Petri nets | × | × | Makespan | Timed place petri net | Beam search strategy |
| Huang *et al.* [22] | FMS | Petri nets | × | × | Makespan | Timed place petri net | Ordered binary decision diagrams |
| This paper | FMS | Petri nets Active schedule | Reactive, Robust, Simulation | Machine breakdowns | Makespan, tardiness | Timed Petri nets | Genetic algorithm, simulated annealing, local search |

Mejia and Acevedo [34] development an integrated system of Petri nets to model an FMS. They presented a prototype that simulated the production plan and implemented dispatch rules to resolve possible conflicts.

Tanimizu *et al.* [54] developed a genetic algorithms of continuous rescheduling for the arrival of new jobs. Kim *et al.* [25] presented a new method for an FMS based on Petri nets and a reactive graphical search algorithm, looking for the minimization of the makespan and the total tardiness. Tashnizi *et al.* [55] develop a Petri nets with non-linear scheduling. Tüysüz and Kahraman [58] modeled a flexible manufacturing cell using stochastic Petri nets with fuzzy parameters. Patel and Joshi [43] developed a model for an FMS with deadlock and analyzed it to generate the reachability tree using Petri net system. Han *et al.* [17] proposed a scheduling method that provides a new approach to evaluate the performance of different deadlock controllers with Petri nets and genetic algorithms. Baruwa *et al.* [4] investigated a deadlock-free scheduling method for an FMS based on timed colored Petri nets with a heuristic.

Başak and Albayrak [3] presented a Petri net-based decision system modeling in real-time scheduling and control of an FMS. Han *et al.* [18] proposed an effective hybrid particle swarm optimization algorithm with a timed Petri net model to solve the deadlock-free scheduling problem of an FMS. Li *et al.* [30] modeled scheduling problems with a transition timed Petri net and an improved heuristic function that also considers the available time of shared resources within an FMS. Lei *et al.* [29] solved a deadlock-free scheduling of FMS with the controlled backtracking strategy based on the execution of the Petri nets. Mejía and Niño [35] developed a fast and efficient Beam Search strategy based on Petri Nets for FMS scheduling. Huang *et al.* [22] developed an FMS scheduling based on binary decision diagram and Petri net.

Based on the state-of-the-art review, several studies have addressed the problem of scheduling and rescheduling due to machine failure in an FMS. Research has focused on developing efficient algorithms, improving performance metrics, time and computational cost. In recent years, the use of genetic algorithms combined with local search techniques has been emphasized to obtain better results. However, the results found by its authors are still susceptible to improvement, through the development of new hybridization structures. In this subject, new applications are always required to improve the computational process in its structure, compilation, and integration. These innovations always allow us to find better solutions.

This study presents a new hybrid algorithm construction that combines the Petri Nets and Genetic Algorithm for the initial schedule. This combination allows the achievement of very good solutions within active schedules in environments of an FMS. In reactive rescheduling due to machine breakdowns, the Petri Nets and Memetic Algorithms are integrated with local search techniques and Simulated Annealing. The integration allows robust schedules after the occurrence of machine failure and unavailability. The proposed model allows us to avoid conflicts, blocks, and implement real-time controls. The proposed model is justified by its robustness and stability, the ability to manage uncertainty due to machine failure, its resilience and flexibility to adapt easily to different circumstances. Finally, the novelty of the research is to consider the integration of efficient scheduling methods. Additionally, it is proposed to optimize two important metrics (makespan and tardiness) and simulate failures by scenarios to analyze in context of the complexity of an FMS.

## 3. Method and material

In this section, we describe Petri Nets, Genetic Algorithms, Memetic Algorithms, Simulated Annealing, and Local Search, as backgrounds.

### 3.1. Petri Net

A Petri Net (PetN) is a directed graph, heavy and bipartite, consisting of places, transitions, and arcs, where the arcs go from a place to a transition or a transition to a place. In the graphical representation places are represented by circles, transitions for bars, and bows by arrows. A marking on the PetN indicates the distribution of an integer (positive or zero) token for each position [8]. Murata [38] formally defined Petri Nets as six-folder presentation PetN $= (P, T, O, I, M, W)$, where:

- $n$, number of places in the PetN.

- $m$, number of transactions in the PetN.
- $P = \{p_1, p_2, \ldots, p_n\}$ is set of $n$ places are drawn as circles, for $n > 0$.
- $T = \{t_1, t_2, \ldots, t_m\}$ is set of $m$ transitions are drawn as bars or boxes, with $P \cup T \neq 0$ and $P \cap T = 0$ for $m > 0$.
- $O : T \times P \to N$ is the set of input arcs directed from $T$ to $P$, where $N = \{0, 1, 2, \ldots\}$.
- $I : P \times T \to N$ is the set of output arcs directed from $P$ to $T$, where $N = \{0, 1, 2, \ldots\}$.
- $M(p) : P \to N$ is the marking in the $p$th place or the token number in the $p$th place at some point. An initial labeling is denoted by $M(0)$. Tokens (black dots) reside in places and represent the truth of the condition of action associated with the corresponding place. Tokens move throughout the net by effect of transition firings.
- $W$ is the set of weights associated with the arcs.

For purposes of simplifying the other definitions in the Petri Nets, Murata [38] use the following symbolic representations, which have been adopted for purposes of this research:

- $K(p)$, the maximum number of tokens that can hold the place $p$.
- $w(p, t)$, weight of the arc that communicates the place $p$ with the transition $t$.
- $w(t, p)$, weight of the arc that communicates the transition $t$ with the place $p$.
- ${}^\bullet t$, set of places of input to the transition $t$.
- $t^\bullet$, set of output locations transition $t$.
- ${}^\bullet p$, set of input transitions to place $p$.
- $p^\bullet$, set of output transitions to place $p$.
- $R(M(0))$, set of possible markings made from $M(0)$.
- $L(M(0))$, the set of sequences to fire $M(0)$.
- $C = \{C_{i,j}\}$, incidence matrix of $n \times m$, such that if all weights are assumed as one, so: $C_{i,j} = 1$ if the place $i$ is an output place for the transition $j$, $C_{i,j} = -1$ if the place $i$ is an input place for the transition $j$, and $C_{i,j} = 0$ otherwise.

In Petri nets, a transition $t$ is said to be enabled if all its input places are marked at least $w(p, t)$ tokens. Also, a transition can be fired if enabled. When a transition is fired, $w(p, t)$ tokens are removed from their input places and they put $w(t, p)$ tokens in their output places, considering the constraints of the system. A transition without input place is called an enabled transition, while a transition without output place is a final transition. A transition $t$ and a place $p$ form a loop if $p$ is, at the same time, input, and output place of $t$.

One of the great advantages of the application of Petri nets to real systems is the ability to monitor their evolution. The evolution of the net allows us to identify the movements of the tokens between the places, which in turn define the vector of marking and its changes in time. If in a finite capacity PetN transition $t$ fires, it is satisfied that the number of tokens in each output place $p$ of that transition, does not exceed its maximum capacity $K(p)$, which is represented by: $M'(p) + w(t, p) \leq K(p)$. The equations of state make use of $C = \{C_{i,j}\}$, where $C_{i,j} = C^+_{i,j} - C^-_{i,j}$, with $C^+_{i,j} = w(i, j)$ and $C^-_{i,j} = w(j, i)$. The evolution of the system from one step to another, is represented by the equation of state $M(k + l) = M(k) + C \cdot u(k)$, where $M(k)$ represents vector marking, after of $k$ fired transitions, $l$ the net change in the tokens in place $i$ when transition $j$ is fired, and $u(k)$ the vector transition firing, which contains zeros except $j$th position containing a 1, indicating that $j$th transition is fired after $k$ events. If a marking $M(d)$ destination is reachable from $M(0)$ through a firing sequence $\{u_1, u_2, \ldots, u_d\}$, the state equation is written as: $M(d) = M(0) + C \cdot \sum_{k=1}^{d} u(k)$, or as $\Delta M = C \cdot X$, if makes $M(d) - M(0) = \Delta M$ and $\sum_{k=1}^{d} u(k) = X$, with $X$ column vector whose input the $j$th position is the number of times the transition has been fired $j$. Figure 1 illustrates a Petri Nets model of an FMS with two jobs.

The benefit of using the Petri Nets as a tool for modeling the systems is the knowledge about the state at each moment of its evolution using the vectors of markings. It is known as the evolution of the network every step it takes over time, that is, the movement of the tokens between the places after the fire of the transitions. Two types of properties of the Petri Nets have been identified, those that depend on the initial state or marking

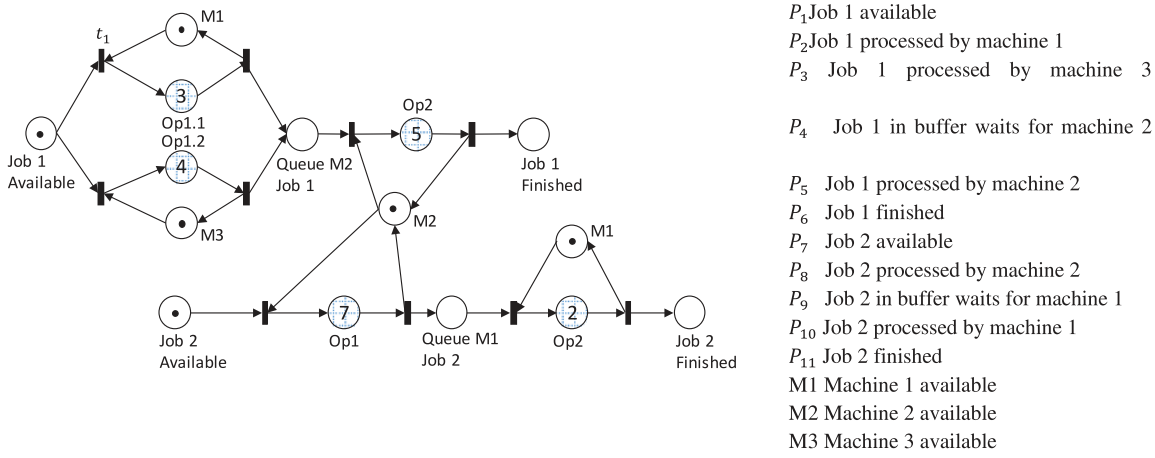



FIGURE 1. Petri Nets model for FMS with two jobs and three machines [1].

of the Petri Nets, called behavioral, and the properties that depend on the structure of the network, called structural. The marking of a Petri Net changes after each fire a transition.

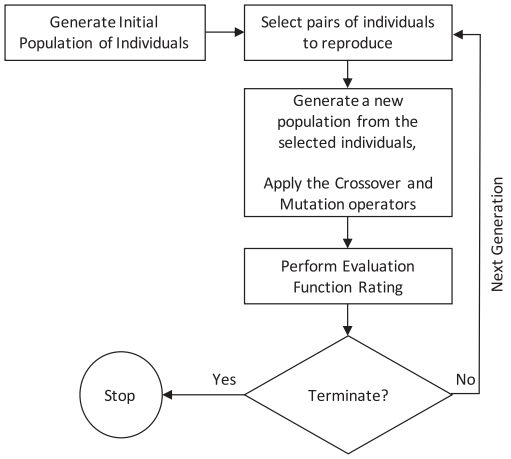
Petri Nets have been widely used for the scheduling and rescheduling of production in manufacturing systems. A Petri Nets is most often used as a tool to model and control the production process in an FMS [3, 18, 22, 29, 35, 39]. Due to the graphic nature, descriptive capacity, Petri Net as a powerful tool to play an important role in modeling and analysis [27, 28, 43].

### 3.2. Memetic algorithm

Memetic Algorithm (MA) is an optimization technique that combines other metaheuristics (population-based search and local improvement). The MA is based on individual improvements of the solutions in each of the agents along with cooperation processes and competitions. Moscato and Cotta [37] related the origins of MA in the late eighties when evolutionary strategies and genetic algorithms began to take advantage. The name Memetic comes from the term meme introduced by Richard Dawkings to represent a unit of cultural evolution. In the context of optimization, a meme represents a learning or development strategy. MA contributes to the solution of production scheduling in optimal or near-optimal solutions, avoiding local minima or premature convergence [9]. MA convergence is treated with local search techniques [14].

The MA starts from a genetic algorithm, it plays an important role in its structure. They constitute the global search tool providing the coverage of diversity while the local search provides the intensification. Genetic algorithms are part of the evolutionary algorithms that constitute a general technique for solving search and optimization problems [46]. The form in which evolutionary algorithms work is related to how species evolve in nature. Genetic algorithms use a set of possible solutions or individuals to calculate adaptation measures. Through an iterative process this population changes and each iteration are called generation. For an individual to survive and pass on to the next generation it must have a high level of adaptation and participate in genetic operations, with which new individuals are created who constitute the next generation. These algorithms allow us to address problems of great complexity of search and optimization. The most popular evolutionary algorithms are given its efficiency and ease of implementation. Solutions are represented as a bit string before being decoded. The behavior is defined as the following parameters: the size of the initial population, the number of generations, the percentages of mutation and crossover. A schematic representation of the procedure of the genetic algorithm is shown in Figure 2.

Next, the simulated annealing, and the local search are related, as a hybrid strategy within the memetic algorithms. Local search is a search process in the space of possible solutions. The search begins for a random




Pseudo-code

P1.Generate Initial population
P1.1. Define a genetic representation of the system (Chromosome Representation).
P1.2 Generate a random population of n chromosomes (Initial Population).
P2. Evaluate the level of adaptation of each chromosome in the population (Evaluate the fitness).
P3. Create a new population (Next Generation)
P3.1. Select two chromosomes parents of a population according to their level of adaptation (Selection mechanism).
P3.2. Chromosomes reproduce among themselves according to a predefined crossover probability, create a new offspring (the children) (Crossover)
P3.3. Genes of a resulting child individual are exchanged according to the mutation probability(Mutation).
P3.4 The best $\alpha\%$ individuals generate the $\beta\%$ of the children at the next generation(Elitism $(\alpha, \beta)$)
P3.5. Place the new offspring in a new population (New Generation).
P4. The algorithm terminates after a predefined number of generations or if after several generations, the algorithm has not found a better solution (Termination Criterion).

FIGURE 2. Flow diagram and pseudocode of a genetic algorithm.

solution and then focuses on neighboring solutions iteratively. Iterations are made by setting a parameter by the memory of the found solutions [21]. The local search has as components: a search space, a set of feasible solutions, a neighborhood relation, an initialization function, a step function, and a completion condition. Simulated annealing was proposed by Kirkpatrick *et al.* [26] motivated by the annealing of the solids, a process in which the solids are heated and subsequently cooled slowly to obtain perfect crystal structures. The main idea of SA is an observed analogy between a complex system optimization and a description of the physical behavior of a system. This method is applied in many domains: operational research, production scheduling, and many others. Simulated annealing can be considered as a process in which in a neighborhood you try to move from one current solution to another of your neighbors. SA generates a new solution $S'$ in the neighborhood of the current solution, then calculates the change $d = C(S') - C(s)$, where $C(s)$ the value of the objective function of the initial solution. The algorithm must be designed using certain methods to represent solutions, generate neighboring solutions, and reduce the temperature. The parameter $T$ decreases gradually by a cooling function until a stop condition is satisfied.

## 4. Problem definition, assumptions, and notation

This section contains problem definition, assumptions followed by notation are used such as it is much easier to understand the model. The hybrid PetNMA algorithm is composed of three (3) sub algorithms. The first performs the initial scheduling of the jobs. The second simulates the initial scheduling until the machine fails that results a rescheduling. The third generates a reactive scheduling through the application of a memetic algorithm.

### 4.1. Notation

The notation used to represent mathematically what happens with the set of operations that enter the reactive scheduling of the production is the following:

$n$ : number of places.

The places are operational $(O)$ or resources $(R)$: $P = \{p_1, p_2, \ldots, p_n\} = O \cup R$.

$m$ : number of transitions.
$ump$ : vector of enabled transitions.
$X_{i,j,k}$ : time when the $i$th operation of the job $j$ on the machine $k$ starts in the initial schedule.

$p_{i,j,k}$ : processing time of the $i$th operation of the $j$th job on the machine $k$.
$C_{i,j,k}$ : completion time of the $i$th operation of the $j$th job on the machine $k$.
$O_{i,j,s}$ : $i$th operation of $j$th job on a machine of workstation $s$.
$O^R_{i,j,k}$ : $i$th operation of $j$th job on $k$th machine.
$T^R_k$ : time availability of $k$th machine reactive scheduling.
$\alpha$ : percentage of the makespan in the weighted objective.
$\beta$ : percentage of total weighted tardiness in the weighted objective.
CONP : set of undeveloped operations.
PSize : represents the GA parameter, associated with the size of the population.
$nbd$ : represents the number of faults of simulated machines.
$r$ : $r$th reactive scheduling activity, due to failure number $r = (1, 2, \ldots, nbd)$.
$t_{bd}$ : instant when the failure occurs.
NTPI : number of transitions programmed to be fired in the initial schedule.
NES($t_{bd}$) : number of transitions fired during the simulation until the moment of failure.
$ng$ : number of genes that make up the chromosome.
$ng_r$ : number of genes that make up the chromosome for reactive scheduling, due to failure $r$.
$ng_{r-1}$ : number of genes that formed the chromosome in reactive scheduling, due to the failure $r-1$. When $r = 1, ng_0 = 2\sum_{j=1}^{N_0} n_j$.

The operational place represents the action "Operation $O_{i,j,k}$ in process" and the conditions "job available", "job completed" and "job Waiting at workstation $s$". The resource places represent the "availability of a machine", where the initial marking of the resource places is 1 for all resources. All resource places are untimed.

### 4.2. Assumptions

The problem of reactive scheduling due to machine failures in an FMS, is formally declared with the following assumptions:

(1) The FMS is composed of a set of machines $M^T = \{m_1, m_2, \ldots, m_k, \ldots, m_M\}$ and a set of work orders $J = \{J_1, J_2, \ldots, J_j \ldots, J_N\}$.
(2) Each work order $J_j$ has a delivery date $d_j$ committed to the customer, a priority defined by their level of importance $w_j$ and consists of a sequence of operations $O_j = \{O_{1,j}, O_{2,j}, \ldots, O_{i,j}, \ldots, O_{n_j,j}\} \forall j$.
(3) Each machine $m_{k \forall k}$ can process only one operation until failure $P(\text{Fault})_k$ occurs in the scheduling horizon.
(4) Assumed that all $p_{i,j,k}$, $d_j$ and $w_j$ are deterministic and processing time $p_{i,j,k}$ includes the time of preparation or setup on the machine. Where $k$ represents the index of machines ($k = 1, 2, \ldots, M$), $j$ is the index of work orders ($j = 1, 2, \ldots, N$), and $i$ is the index of operations ($i = 1, 2, \ldots, n_j$).
(5) Each workstation $s$ has a space at the beginning which is used for temporary storage of all jobs will be processed. where $s$ is the index of workstations ($s = 1, 2, \ldots, S$).
(6) When a job is completely processed in a workstation, it is moved to the next workstation or inventory of finished products.
(7) The interruption of the scheduling is generated due to machine failures.
(8) In the reactive scheduling, the operations are in process in a machine without failure, *i.e.*, are not interrupted. Any operation that is being processed on a machine that fails, will be break and will have to be reinitialized.
(9) All operations conform to reschedule operations that have not been done and the operations are affected by the failure of the machine.
(10) The objective functions considered here is to minimize Makespan ($C_{\max} = \max(C_j)$), Total Weighted Tardiness ($\sum_{j=1}^{N} w_j T_j$) and weighted objective ($\alpha C_{\max} + \beta \sum_{j=1}^{N} w_j T_j$) with $\alpha + \beta = 1$.

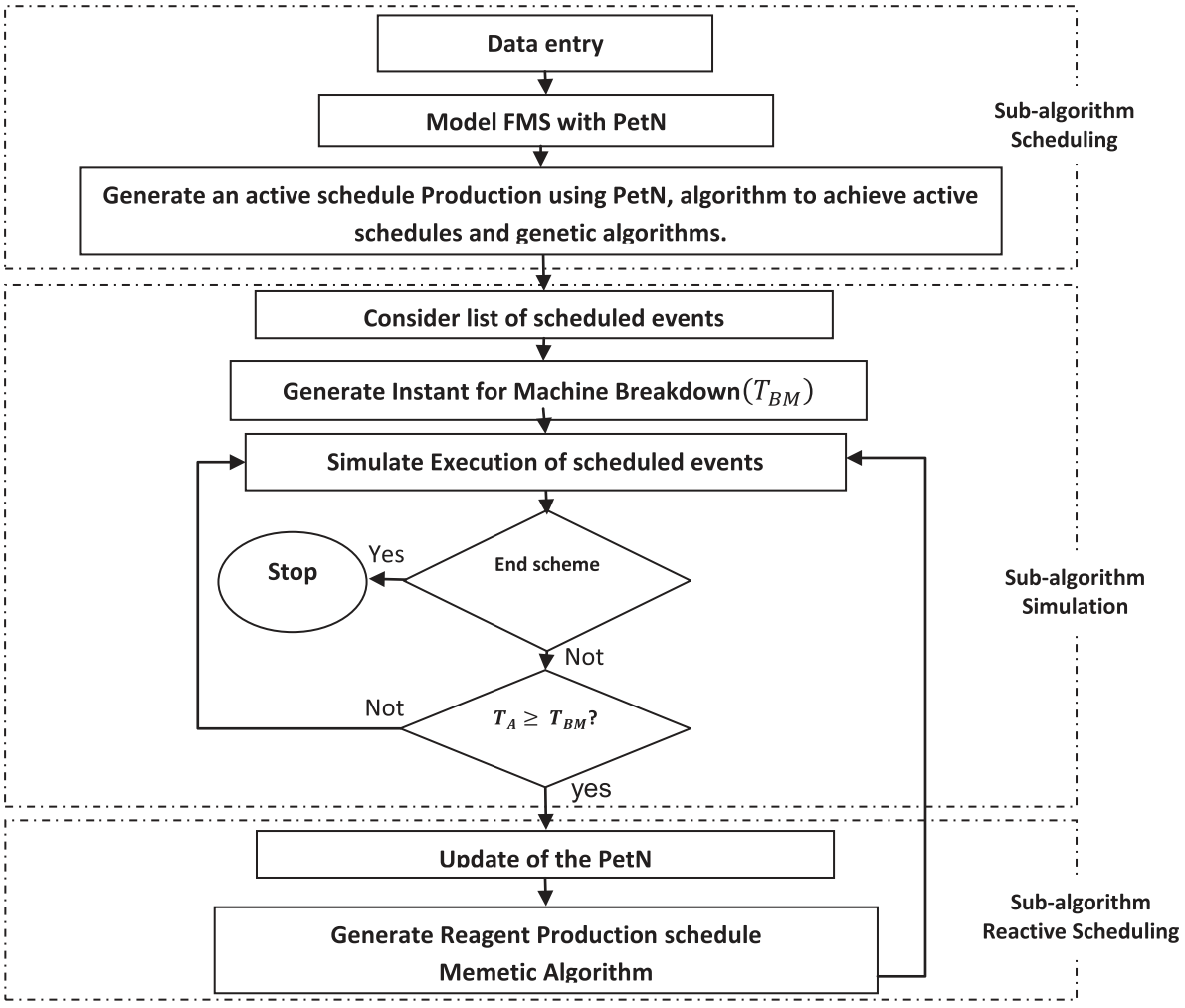



FIGURE 3. PetNMA: Reactive scheduling algorithm of an FMS with machine failure.

## 4.3. Scheduling and system simulation

The PetNMA algorithm is based on a series of sub-algorithms which develop in a sequence of three stages: Sub-algorithm of Scheduling, Sub-algorithm of Simulation and Sub-algorithm of Reactive Scheduling (see Fig. 3).

### *4.3.1. Sub-algorithm scheduling*

The sub-algorithm is to structure an initial production schedule in an FMS with all jobs $j$, using modeling Petri nets, the algorithm for obtaining active schedules, and genetic algorithms. The sub-algorithm is supported by the LEKIN software. The initial scheduling is constructed, by establishing the sequence of fired transitions supported by the dispatch rules (SPT, LPT, MRWT, EDD, CR, ATC, and MSNACK). The Sub-algorithm Scheduling has the objective of obtaining active schedules. Therefore, genetic algorithms are applied to obtain new schedules. In the implementation of the Sub-algorithm Scheduling in Microsoft Visual Studio 2010, the format of the Petri net class was taken from the Afs-Petrinet Algorithm [1] and adapted to represent machine failures. Below are the detailed steps of the Sub-algorithm Scheduling (see Algorithm 1).

**Algorithm 1.** Initial scheduling.

P.1 Create the FMS in the software "LEKIN".
P.2 Create PetN of the FMS.
- a. Read files generated by LEKIN (*.job and *.mch)
- b. Create the finite set of places (P), with $= N_0 + S + 2M + \sum_{j=1}^{N_0} \sum_{i=1}^{n_j} m_{i,j}$ .
- c. Create the finite set of transitions (T), with $m = 3 \sum_{j=1}^{N_0} \sum_{i=1}^{n_j} m_{i,j} + M$.
- d. Create the set of arcs directed from $t$ to $p(TxP)$ and set of arcs directed from $p$ to $(PxT)$.
- e. Create the vector of marking $M_p$ for the PetN, initializing it $(M_0)$.
- f. Create the time vector "$PT$" of the places of the PetN. The times associated with the machines, storage at the entrance of the stations and termination of the work order, are always zero (0).
- g. Create the remaining time vector "$Mr$" with value zero (0).
- h. Create the vector of time remaining work "$rw$".
- i. Establish the incidence matrix negative $(C^-)$ and positive $(C^+)$ of the PetN.

P.3 Generate the initial scheduling through the algorithm of active programs.
- a. Create the vector of enabled transitions $(ump)$.
- b. Select the transition to fire.
  - i. Calculate $C_{i,j,k}$ operations available to be programmed. Organize high to low and store in a matrix $C$ along with the transitions that would begin operations.
  - ii. Detect available machines $(m^*)$ in which the minimum time of termination $(f)$ occurs.
  - iii. Create the matrix $G$, how of the set operations that can be initiated on the machine $m^*$, and which initial time is lesser that $f$.
  - iv. Make assignments of available resources (machines) considering first operations would end later to have a gap in scheduling. Dispatching rules are used.
  - v. Identify the related transition with the selected operation on the machine $m^*$ and fire.
- c. Moving the transition fired of $ump$ to transition fired vector $(u)$ and update vectors "$Mp$", "$Mr$" and "$rw$".
- d. Update $ump$. If the $ump$ is empty, a workable program for the FMS has been obtained, otherwise go to step P3.b.
- e. Save the seven sequences obtained in the corresponding vectors and their objective values.

P.4 Generate Initial Schedule by GA.
- a. Chromosome Representation: Realize the genetic representation of the system with a sequence of $ng$ genes obtained randomly in the interval $[0, ng - 1]$, in an indirect representation chromosome, which is then decoded to represent transitions to be fired. The chromosome is a string of integer numbers ranging from 1 to a sufficiently large number, typically 10 to 100 times the total number of operations. The length of the chromosome corresponds to twice the total number of operations. This equals the total number of transitions to be fired.
- b. Initial population: All individuals were generated randomly except as even which were generated with the rules of dispatch (Active Programs), associated with $PSize$.
- c. Decoding: The module function (%) uses the vector enabled transitions $(ump)$ to obtain the vector of transitions firing $(u)$. The firing transition is selected by taking the module of the integer division between the gene and the number of enabled transitions. The selected transition fires, the marking of the net changes and a new set of transitions is enabled. The next gene in the chromosome determines the new transition to fire. This chromosome always guarantees a feasible sequence of transition firings.
- d. Adaptation level: The level of adaptation of the GA evaluates the chromosome, and thus the target values (makespan, total weighted tardiness or the weighted value).
- e. Generate a new population
  - i. Selection mechanism: The tournament strategy is used: A first pair individuals is chosen randomly among the current population and the fittest is selected to be the first parent. Another pair of individuals is chosen and a second parent is selected in the same manner. The two selected parents will reproduce according to the crossover probability. Choose the best individual of the population.
  - ii. Crossover: The classical one-point crossover strategy was followed: a random position (crossover point) in a chromosome is chosen.
  - iii. Mutation: Two genes of a resulting child individual are exchanged according to the mutation probability.
  - iv. Elitism$(\alpha, \beta)$: The best $\alpha$% individuals generate the $\beta$% of the children at the next generation.
  - v. Evaluate the fitness of new individuals.
  - vi. Next generation: The best individuals among parents and children produce the next generation.
- f. Termination criterion: The algorithm terminates after a predefined number of generations $(maxGenerations)$ or if the algorithm has not found a better solution.
- g. Consider the best solution to the current population, as the seventh sequence of transitions.

P.5 Select the best solution as the sequence of transitions to generate the schedule.
P.6 Change the sequence of fire transitions $(u)$ select to a sequence in Gant, according to the format of the *.seq file of LEKIN and follow the Sub-algorithm Simulation.

The genetic algorithms notice that the completion of an operation requires the firing of a pair of transitions: The first transition of the pair marks the start and the second represents the termination of any operation. The decoding is based on the work done by Mejia and Acevedo [34], *e.g.*, take Figure 4 and the following chromosome: The three transitions that are enabled at this state conform the array $ump(0) = [t_1, t_2, t_5]$. Transitions are put into the array in lexicographic order. The remainder of the integer division between the first gene (6) and the size (3) of the array $ump$ is 0. Thus transition $t_1$ in the 0th position of $ump$ is selected to fire. Transition $t_1$ is fire and changes the state of the net, generating a new vector of enabled transitions $ump(1) = [t_3, t_5]$ (see Fig. 4). The second gene (77) is taken. The remainder of the gene (77) and the size of the array (2) is 1 and the 1th position is selected. This corresponds to transition $t_5$. The process is repeated until reaching the final marking.

### *4.3.2. Sub-algorithm simulation*

To perform a simulation of the events we must consider the vector of fired transitions obtained in the Scheduling Sub algorithm. This vector relates the times in which each operation must be performed. This list of

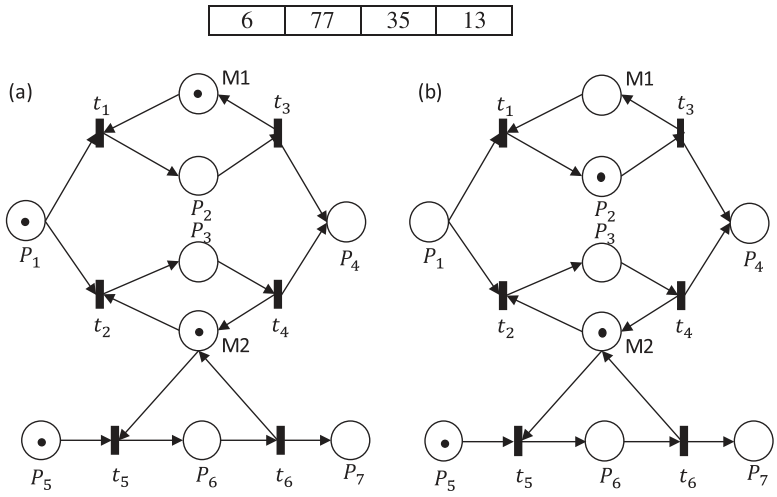



FIGURE 4. Example of chromosome decoding.

events to be performed varies with time as each event takes place. However, the operations are carried out in some machines can present failures. This is what should be done reactive scheduling or rescheduling of production. Previously, it is necessary to perform a simulation of the execution of operations that have been programmed and machine failures. It should be noted that it does not have historical data, which is why the failure to generate the methodology used by Mehta [33]. To determine the time when the failure occurs the normal probability distribution is used, to determine the machine fault, since all machines have the same probability of failure. However, in this study to determine the repair time of the machine we use the Log-Normal distribution to establish the failure time (Repair time). The time of each failure or repair time, is given by $\mathrm{LN}(\ln(\gamma_1 p), \gamma_2)$. Where, $\ln(\gamma_1 p)$ is the average of the function, $p$ is the average processing time of the system, $\gamma_1$ is the percentage of $p$ which is part of the repair timeand $\gamma_2$ is the variance. Below are the detailed steps of the Sub-algorithm Simulation of scheduled events and failures of machines (see Algorithm 2).

**Algorithm 2.** Sub-algorithm simulation – Part A.

P.1 Scheduled Events list
- a. Consider the u in the Sub-algorithm Scheduling, as a LIST OF PROGRAMMED EVENTS.
- b. Create a list of random events that represent the failures machines. In this list of events you must specify the $nbd$, $t_{bd}$ and the $T_k^R$ it does it.

P.2 Simulation of the execution of Scheduled Events
- a. Remove the first entry of the LIST OF SCHEDULED EVENTS to the list of events executed.
- b. Forward the clock time to the time associated with the removed event and update the state of the system to the occurrence of the event.
- c. Evaluate if the list of scheduled events is empty (if STOP, otherwise go to P2.d).
- d. Evaluate if the clock time is greater than or equal to the associated time of the machine failure (If, execute the Sub-algorithm Reactive Scheduling, otherwise go to step P2.a).

To carry out the Reactive scheduling in PetNMA, all the operations that at the time of the failure had not been executed must be considered, as well as the operation that was being processed in the machine at that moment, forming the set CONP. The operations that were being processed during the failure inoperable machines are not interrupted until they have been completed. The following is a series of steps to obtain the list of those operations (see Algorithm 3).

**Algorithm 3.** Sub-algorithm simulation – Part B.

P.1 Start with an empty set $CONP = \emptyset$.
P.2 Add to $CONP$ the operation that was being processed at the time of the machine failure $(CONP = \{O_{i,j,k}^R\})$.
P.3 Add to $CONP$all the operations that at the time of the failure had not been executed of the initial program $(CONP = CONP + \{O_{r,j,k}\}\ \forall r, j, k)$,where it is satisfied that$X_{r,i,k} \geq t_a$and$j \leq N$.

TABLE 2. Route of the operations of two jobs in a manufacturing system with four machines.

| Operation | Job 1 | Job 2 |
|---|---|---|
| (Operation 1) | M1(6) | M2 (7) |
| (Operation 2) | M3 (8) o M4 (8) | M4 (5) |

*4.3.3. Sub-algorithm of reactive scheduling*

When a machine failure occurs during the development of the Sub-algorithm, the rescheduling must be done through the Sub-algorithm Reactive Scheduling. For this scheduling, all operations in CONP must be considered. The MA used consists of an evolutionary algorithm and a local search technique. The GA was selected for the evolutionary part. Meanwhile, as a local search technique, the simulated annealing algorithm runs after each generation. The Sub-algorithm Reactive Scheduling is composed of the following steps (see Algorithm 4).

**Algorithm 4.** Sub-algorithm of reactive scheduling.

P.1 Update of the PetN
- a. Set the clock time at the time the machine failure occurs.
- b. Update the vector of remnant process times according to the time of the clock in the simulation.

P.2 Generate Reactive Scheduling for MA
- a. Chromosome Representation: Realize the genetic representation of the system with a sequence of $ng_r$ genes obtained randomly in the interval $[0, ng_r - 1]$, in an indirect representation chromosome. $ng_r$ depends on the state of the PetN in the instant machine failure occurs and is given by $ng_r = NTPI - NES(t_{ar}) + 2$.
- b. Initial population: All individuals were generated randomly except a seven which were generated with the rules of dispatch (Active Programs).
- c. Decoding: The module function (%) uses the vector enabled transitions ($ump$) to obtain the vector of transitions firing($u$).
- d. Adaptation level: The level of adaptation of the GA evaluates the chromosome, and thus the target values (makespan, total weighted tardiness or the weighted value).
- e. Choose the best individual of the population.
- f. SA is applied with the technique of LS.
  - i. The best solution obtained from the latest generation (local optima) is chosen.
  - ii. The initial temperature and the final temperature are initialized.
  - iii. A neighbor is generated by exchanging two genes of the local optimum.
    - i. The fitness neighbor is evaluated and a delta equivalent to the neighboring fitness least local optimum fitness is generated.
    - ii. The termination criterion is evaluated, if the neighbor's fitness is less than optimal local new local optimum will be the neighbor, but probabilistic acceptance criteria is evaluated.
    - iii. The temperature is updated.
  - iv. If the temperature is higher than the final temperature, return to step 2.f.iii, otherwise continue with step P2.g.
- g. Generate a new population
  - i. Select chromosomes for reproduction (tournament).
  - ii. Apply crossover (one-point strategy)and mutation (two genes are exchanged according to the mutation probability), elitism operators($\alpha, \beta$).
  - iii. Evaluate the fitness of new individuals.
  - iv. Next generation: The best individuals among parents and children produce the next generation.
- h. Termination criterion: The algorithm terminates after a predefined number of generations ($maxGenerations$) or if the algorithm has not found a better solution.
- i. Consider the best solution to the current population, as the seventh sequence of transitions.

P.3 Select the best solution as the sequence of transitions to generate the reactive scheduling.

P.4 Change the sequence of fire transitions ($u$) select to a sequence in Gantin the Reactive Scheduling.

## 5. EXPERIMENTAL EVALUATION

The experimental tests were developed by running on a Dell Inspiron computer with an Intel Inside CORE I5 processor, 8 gigs of RAM, 1 TB of memory, and Windows 7. The algorithms were developed in Microsoft Visual Studio 2010 in C++ language.

### 5.1. Numerical illustration of the sub-algorithm scheduling

To better understand this stage of the algorithm, corresponding to phase 1 of the Sub-algorithm Scheduling, an example of its application is specified using a numerical example. A manufacturing system develops two (2) jobs with two (2) operations with an established sequence and four (4) machines in the system. The path of each job is shown in Table 2, where the process times of each operation are shown in parentheses.

The modeling system described above is as follows:

The input and output functions of Petri nets used to represent the manufacturing system in Figure 5, can be represented using the following incidence matrices (Tabs. 3–5):

In the present study, the Timed Petri Nets were used to model the FMS together with the random faults of the machines. In Figure 6, the system represented above can be evidenced with additional places and transitions

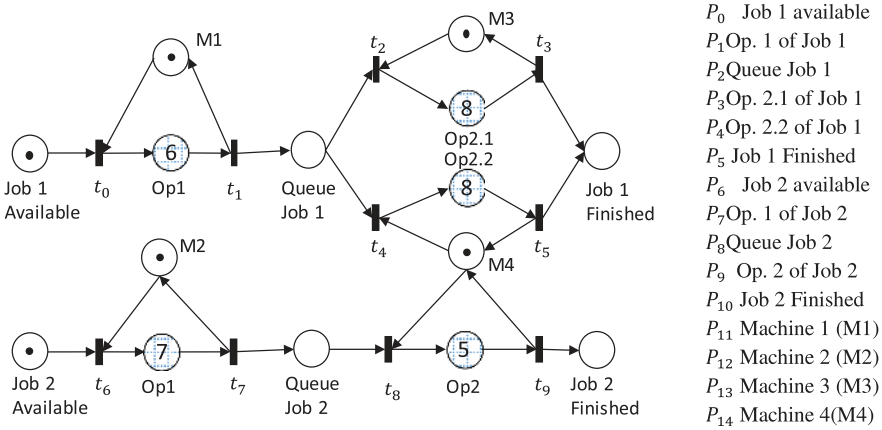



FIGURE 5. PetN model for a Job Shop with two jobs, four machines.

TABLE 3. Positive incidence matrix for the job shop represented in Figure 5.

| $C^+ =$ | $p_0$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $p_{11}$ | $p_{12}$ | $p_{13}$ | $p_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_0$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_1$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $t_2$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $t_4$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_5$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $t_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| $t_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $t_9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

TABLE 4. Negative incidence matrix for the job shop represented in Figure 5.

| $C^- =$ | $p_0$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $p_{11}$ | $p_{12}$ | $p_{13}$ | $p_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_0$ | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 |
| $t_1$ | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_2$ | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 |
| $t_3$ | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_4$ | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 |
| $t_5$ | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_6$ | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 |
| $t_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | −1 |
| $t_9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 |

TABLE 5. Incidence matrix for the job shop represented in Figure 5.

| | | $p_0$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $p_{11}$ | $p_{12}$ | $p_{13}$ | $p_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $t_0$ | −1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 |
| | $t_1$ | 0 | −1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | $t_2$ | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 |
| | $t_3$ | 0 | 0 | 0 | −1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | $t_4$ | 0 | 0 | −1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 |
| $C =$ | $t_5$ | 0 | 0 | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | $t_6$ | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 0 | −1 | 0 | 0 |
| | $t_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | $t_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 0 | −1 |
| | $t_9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 1 | 0 | 0 | 0 | 1 |

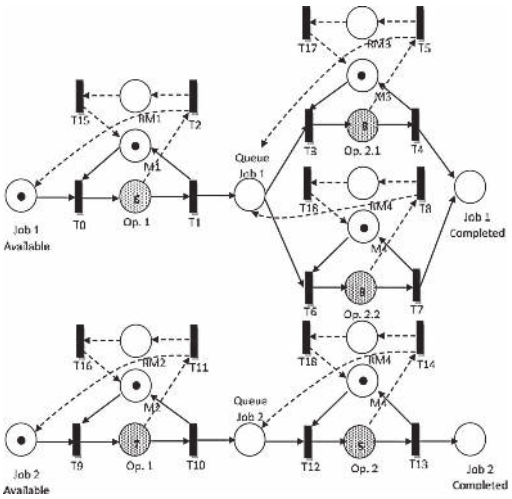



FIGURE 6. Petri Net model for a Job Shop with 2 jobs, 4 machines, considering machine failures.

that correspond to the machines under repair after a failure. Transitions machine failures are fired when the corresponding machine simulation has failed while the operation is processed. For this, one must consider the $M_k$ vector marking at the time the machine fails. When the corresponding transition is fired, the token that is in the operation place is removed and two tokens are added to the place corresponding to the repair of the machine and to the available job place (if the operation was the initial one) or to the queue from a job. At the end of the repair time of the machine, the corresponding transition is fired, and the token is returned to the machine so that it is available again.

Where, Op. 1 Operation 1; Op. 2 Operation 2; Op. 2.1 Operation 2 option 1; Op. 2.2 Operation 2 option 2; M1 Machine 1; M2 Machine 2; M3 Machine 3; M4 Machine 4; RM1 Repair of the machine 1; RM2 Repair of the machine 2; RM3 Repair of the machine 3; RM4 Repair of the machine 4; T0, T3, T6, T12 Transitions that start operations; T1, T4, T7, T10, T13 Transitions that end operations; T2 Failure M1 while working in Op. 1 of job 1; T5 M3 failure while working in Op. 2.1 of job 1; T8 M4 failure while working in Op. 2.2 of job 1; T11 M2 failure while working in Op. 1 of job 2; T14 M4 failure while working in Op. 2 of job 2; T15 Enables M1 after being repaired; T16 Enables M2 after being repaired; T17 Enables M3 after being repaired; T18 Enables M4 after being repaired. When the failure of a machine is generated, the rescheduling of the activities must be carried out with many strategies, through the sub-algorithm of reactive scheduling.

TABLE 6. Description of the 10 problems chosen to perform the experimental tests.

| No. of problem | No. of jobs | No. of workstations | No. of machines |
|---|---|---|---|
| 1 | 6 | 10 | 19 |
| 2 | 9 | 8 | 20 |
| 3 | 10 | 5 | 10 |
| 4 | 10 | 5 | 12 |
| 5 | 10 | 5 | 12 |
| 6 | 10 | 7 | 18 |
| 7 | 10 | 10 | 20 |
| 8 | 15 | 5 | 12 |
| 9 | 15 | 6 | 17 |
| 10 | 20 | 6 | 13 |

### 5.2. Experimental tests of the scheduling sub algorithm in PetNMA

To obtain the initial schedules with the Scheduling Sub algorithm in PetNMA, ten (10) problems that characterize an FMS were used (see Appendix A). Problems that were used for testing were adapted from the problems developed by Mejia and Acevedo [34] and the classic problems LA and ORB. Each problem has a few jobs to be scheduled in a few workstations and machines (see Tab. 6).

- The route of the work by stations in the FMS consists of times each of their operations. These times were set randomly within the range [36,37], to obtain variability problems.
- In the classical problems, the assignment of the machines to each of the workstations is performed randomly within the interval [10,13,34]. An assignment of machines stations than three (3) system produces downtime machines throughout the scheduling.
- The job weights were assigned randomly considering the uniform distribution within the interval [10,13,34].
- In the delivery times of each work, the total work rule (TWK) was used, where $d_j = kP_j$, where $P_j$ is the sum of the process times of all operations of work $j$ and $k$. However, a normal distribution with a range $[1.3, 1.5]$ was used to give more diversification to the delivery times.
- In designing experiments to validate the parameters, the problems according to the number of jobs in small, medium, and large are classified. Considering the classification problems from 1 to 3 correspond to small, from 4 to 7 to medium and from 8 to 10 to large.

To evaluate the performance of the Scheduling Sub algorithm in PetNMA, the results have been compared with the dispatch rules SPT, LPT, MRWT, CR, ATC, MSLACK, and with the results obtained with the bottleneck. The comparison is made between the objective functions evaluated, in this case the makespan, total weighted tardiness, and weighted objective. For each of the objectives it has conducted a factorial design experiment to choose the best parameters for the genetic algorithm. The parameters chosen for the bullfights were taken from previous studies identified in the literature review. Three types of problems (small problem (2), medium problem (7), and large problem (9)) are chosen to perform the parametrization runs in the experimental design (see Tab. 7). The results of the runs, the experiment design and its graphs are in Appendix B. In the case of the weighted objective, the tests were carried out with all the weighting options for alpha and beta with one decimal, to choose the combination that finds the best results, varying them simultaneously from 0 to 1. In accordance with the tests and the experimental design and Pareto graphs, the weighted objective combination was selected 50–50. Considering the results obtained, the following values were chosen for the parameters of the genetic algorithm of the Initial Scheduling Sub algorithm:

Next, the results of 10 runs of each of the 10 problems studied are shown with the parameters chosen for each problem (see Tabs. 8–10).

TABLE 7. Parameters selected for the genetic algorithm of initial Scheduling in PetNMA.

| | | | Makespan | | | TWT | | | Weighted objective | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | Nomenclature | Values | S | M | B | S | M | B | S | M | B |
| Number of generations | Maxgen | 100; 200 | 100 | 100 | 100 | 200 | 200 | 200 | 200 | 200 | 200 |
| Size of the initial population | $P$ | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Probability of crossing | Tc | 60%; 90% | 60% | 60% | 60% | 90% | 90% | 90% | 60% | 60% | 90% |
| Probability of mutation | Tm | 10%; 50% | 50% | 50% | 50% | 50% | 50% | 50% | 50% | 50% | 50% |
| Intensity of mutations | Im | 1; 2; 4 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| Elitism | Te1 | Off, On 25%; 50% | 50% | 50% | Off | 50% | 50% | 50% | 50% | 50% | 50% |

**Notes.** S: Small; M: Medium; B: Big.

TABLE 8. Results of the initial sub-algorithm scheduling in PetNMA for minimization Makespan.

| Problem | Bottleneck | Dispatch rules | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Min |
| FMS01 | 412 | 436 | 464 | 436 | 464 | 436 | 436 | 464 | 436 |
| FMS02 | 316 | 377 | 346 | 346 | 377 | 324 | 324 | 361 | 324 |
| FMS03 | 536 | 726 | 757 | 536 | 703 | 632 | 632 | 593 | 536 |
| FMS04 | 395 | 470 | 483 | 401 | 487 | 483 | 489 | 418 | 401 |
| FMS05 | 412 | 476 | 472 | 390 | 516 | 447 | 447 | 506 | 390 |
| FMS06 | 342 | 332 | 382 | 357 | 375 | 352 | 352 | 349 | 332 |
| FMS07 | 1143 | 1152 | 1146 | 1149 | 1110 | 1132 | 1155 | 1209 | 1110 |
| FMS08 | 470 | 561 | 559 | 459 | 574 | 569 | 552 | 555 | 459 |
| FMS09 | 269 | 288 | 288 | 263 | 298 | 261 | 275 | 299 | 261 |
| FMS10 | 753 | 836 | 935 | 759 | 814 | 912 | 873 | 779 | 759 |

| Problem | Active schedules with dispatch rules | | | | | | | Genetic algorithm | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Min | Min | Max | Average |
| FMS01 | 480 | 456 | 412 | 480 | 456 | 456 | 456 | 412 | 436 | 436 | 436 |
| FMS02 | 369 | 357 | 326 | 377 | 341 | 367 | 361 | 326 | 324 | 324 | 324 |
| FMS03 | 736 | 754 | 536 | 856 | 632 | 662 | 771 | 536 | 536 | 536 | 536 |
| FMS04 | 483 | 478 | 395 | 524 | 465 | 569 | 478 | 395 | 394 | 394 | 394 |
| FMS05 | 541 | 475 | 379 | 555 | 515 | 469 | 512 | 379 | 390 | 390 | 390 |
| FMS06 | 408 | 468 | 368 | 394 | 380 | 398 | 379 | 368 | 332 | 332 | 332 |
| FMS07 | 1291 | 1198 | 1164 | 1149 | 1158 | 1160 | 1323 | 1149 | 994 | 1036 | 1019.3 |
| FMS08 | 682 | 581 | 487 | 671 | 582 | 633 | 693 | 487 | 442 | 459 | 453.8 |
| FMS09 | 394 | 344 | 287 | 396 | 318 | 328 | 291 | 287 | 249 | 256 | 252.7 |
| FMS10 | 924 | 977 | 800 | 995 | 845 | 999 | 809 | 800 | 753 | 753 | 753 |

TABLE 9. Results of the initial sub-algorithm scheduling in PetNMA for minimization TWT.

| Problem | Bottleneck | Dispatch rules | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Min |
| FMS01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FMS02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FMS03 | 1716 | 1919 | 2525 | 2387 | 2332 | 2177 | 2130 | 2183 | 1919 |
| FMS04 | 50 | 208 | 390 | 205 | 326 | 152 | 404 | 183 | 152 |
| FMS05 | 156 | 226 | 669 | 472 | 347 | 410 | 519 | 240 | 226 |
| FMS06 | 0 | 0 | 130 | 0 | 0 | 0 | 0 | 0 | 0 |
| FMS07 | 359 | 1852 | 1946 | 0 2672 | 1325 | 1917 | 2415 | 2369 | 1325 |
| FMS08 | 304 | 548 | 1882 | 1043 | 648 | 477 | 995 | 439 | 439 |
| FMS09 | 0 | 4 | 310 | 421 | 1 | 69 | 205 | 1 | 1 |
| FMS10 | 5891 | 8449 | 11 815 | 12 099 | 7347 | 10 642 | 11 293 | 8647 | 7347 |

| Problem | Active schedules with dispatch rules | | | | | | | Genetic algorithm | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Min | Min | Max | Average |
| FMS01 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FMS02 | 0 | 0 | 0 | 0 | 3 | 29 | 0 | 0 | 0 | 0 | 0 |
| FMS03 | 2008 | 2656 | 2418 | 3676 | 2158 | 2462 | 2733 | 2008 | 1577 | 1604 | 1581.6 |
| FMS04 | 202 | 454 | 349 | 295 | 159 | 706 | 131 | 131 | 40 | 40 | 40 |
| FMS05 | 446 | 1172 | 1014 | 829 | 679 | 702 | 614 | 446 | 144 | 169 | 155.5 |
| FMS06 | 210 | 436 | 0 | 27 | 37 | 85 | 0 | 0 | 0 | 0 | 0 |
| FMS07 | 3761 | 3079 | 3375 | 1793 | 2300 | 2664 | 4041 | 1793 | 476 | 630 | 563.5 |
| FMS08 | 1896 | 2518 | 1933 | 1634 | 755 | 2065 | 936 | 755 | 150 | 294 | 230.2 |
| FMS09 | 902 | 1126 | 654 | 396 | 186 | 731 | 0 | 0 | 0 | 0 | 0 |
| FMS10 | 11 041 | 14 153 | 13 321 | 9160 | 11 586 | 15 680 | 9309 | 9160 | 6159 | 6944 | 6564.5 |

According to the results in the first phase of the algorithm, significant results were obtained using the GA (see Fig. 7). Makespan was improved in 50% of the instances. In these instances, an average improvement of 4% was achieved. Total weighted tardiness improved in 70% of instances. In these instances, an average improvement of 53% was achieved. The weighted objective (Makespan 50% – TWT 50%) achieved an improvement in 70% of the instances. The average improvement in these instances was 20%. The initial scheduling sub-algorithm in PetNMA has improved the results in the larger instances by comparing them with the dispatch and bottleneck rules, because in small instances the same results can be achieved in the first cycles of the sub-algorithm.

## 5.3. Experimental tests of the reactive scheduling sub algorithm in PetNMA

To obtain the reactive schedules with the Reactive Sub-algorithm Scheduling in PetNMA, a design of experiments was carried out using the three selected problems. For each of the objectives was conducted Taguchi Robust Design to choose the best parameters for the GA considering factors that cannot be controlled when having a breakdown machine. The factors considered to assess the performance of algorithm: Control factors (A: Number of generations, B: Size of the population, C: Probability of crossing, D: Probability of mutation, E: Intensity of mutations, F: Elitism, G: Factor decremental) and Noise factors (H: Instance of the fault, I: Variation of repair time, J: Mean time to repair). Failure scenarios two forms were generated, depending on when the scheduling horizon in which the failure occurs, the percentage of the average processing time used for the repair time of the machine, and the time variation repair. The scenarios can be identified using the following coding: $B(C_x, \gamma_1, \gamma_2)$ where, $C_x$ corresponds to the medium and takes values from 1 to 2, $\gamma_1$ corresponds to the percentage of the average processing time and takes values 0.5 and 1.0, and $\gamma_2$ represents the variation

TABLE 10. Results of the initial sub-algorithm scheduling in PetNMA for minimization weighted global (Makespan (50%) and TWT (50%)).

| Problem | Dispatch rules | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Min |
| FMS01 | 218 | 232 | 218 | 232 | 218 | 218 | 232 | 218 |
| FMS02 | 188.5 | 173 | 173 | 188.5 | 162 | 162 | 180.5 | 162 |
| FMS03 | 1322.5 | 1641 | 1461.5 | 1517.5 | 1404.5 | 1381 | 1388 | 1322.5 |
| FMS04 | 342.5 | 466 | 372 | 409.5 | 312 | 637.5 | 304.5 | 304.5 |
| FMS05 | 351 | 570.5 | 431 | 431.5 | 428.5 | 483 | 373 | 351 |
| FMS06 | 166 | 256 | 178.5 | 187.5 | 176 | 176 | 174.5 | 166 |
| FMS07 | 1502 | 1546 | 1910.5 | 1217.5 | 1524.5 | 1785 | 1789 | 1217.5 |
| FMS08 | 554.5 | 1220.5 | 751 | 611 | 523 | 773.5 | 497 | 497 |
| FMS09 | 146 | 299 | 342 | 149 | 165 | 240 | 150 | 146 |
| FMS10 | 4642.5 | 6375 | 6429 | 4080 | 5777 | 6083 | 4713 | 4080 |

| Problem | Active schedules with dispatch rules (PetNMA) | | | | | | | | Genetic algorithm | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SPT | LPT | MRWT | EDD | CR | ATC | M SLACK | Min | Min | Max | Average |
| FMS01 | 240 | 233 | 206 | 240 | 228 | 228 | 228 | 206 | 218 | 218 | 218 |
| FMS02 | 184.5 | 178.5 | 163 | 188.5 | 172 | 198 | 180.5 | 163 | 162 | 162 | 162 |
| FMS03 | 1372 | 1705 | 1477 | 2264.5 | 1395 | 1562 | 1752 | 1372 | 1114.5 | 1123 | 1115.4 |
| FMS04 | 339 | 436.5 | 303 | 406.5 | 317.5 | 446.5 | 300.5 | 300.5 | 242.5 | 242.5 | 242.5 |
| FMS05 | 493.5 | 823.5 | 696.5 | 692 | 597 | 585.5 | 563 | 493.5 | 303.5 | 320 | 306.8 |
| FMS06 | 309 | 452 | 184 | 210.5 | 208.5 | 241.5 | 189.5 | 184 | 166 | 166 | 166 |
| FMS07 | 2526 | 2138.5 | 2269.5 | 1471 | 1729 | 1912 | 2682 | 1471 | 784.5 | 847.5 | 823.1 |
| FMS08 | 1289 | 1549.5 | 1210 | 1152.5 | 668.5 | 1349 | 814.5 | 668.5 | 358.5 | 404 | 376.7 |
| FMS09 | 648 | 740 | 470.5 | 396 | 252 | 529.5 | 145.5 | 145.5 | 128 | 130.5 | 128.7 |
| FMS10 | 5982.5 | 7565 | 7060.5 | 5077.5 | 6215.5 | 8339.5 | 5059.5 | 5059.5 | 3559.5 | 3735.5 | 3632.9 |

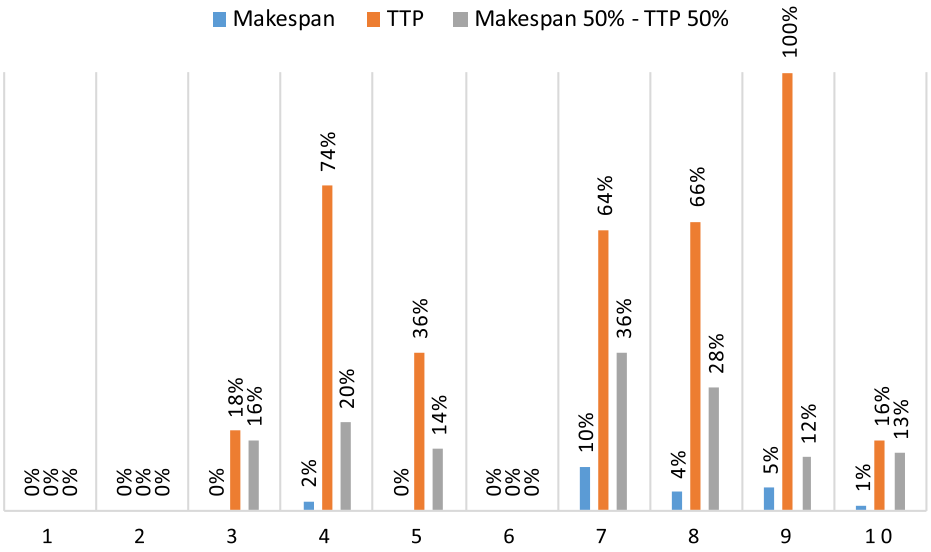



FIGURE 7. Variation of the objectives in the GA with respect to the dispatch rules. Note: TTP: Total Weighted Tardiness.

TABLE 11. Values for MA in sub-algorithm of reactive scheduling.

| | Makespan | | | TWT | | | Weighted objective | | |
|---|---|---|---|---|---|---|---|---|---|
| Parameters | S | M | B | S | M | B | S | M | B |
| Number of generations | 100 | 100 | 100 | 200 | 200 | 200 | 200 | 100 | 200 |
| Size of the initial population | 100 | 100 | 50 | 100 | 100 | 50 | 50 | 100 | 50 |
| Probability of crossing | 90 | 90 | 60 | 60 | 60 | 90 | 90 | 90 | 90 |
| Probability of mutation | 10 | 50 | 10 | 50 | 50 | 50 | 10 | 50 | 50 |
| Intensity of mutations | 1 | 2 | 1 | 2 | 2 | 2 | 4 | 4 | 1 |
| Elitism | 50 | 0 | 0 | 0 | 0 | 50 | 0 | 0 | 50 |
| Factor decremental | 0.1 | 0.01 | 0.01 | 0.1 | 0.1 | 0.01 | 0.1 | 0.01 | 0.01 |

**Notes.** S: Small; M: Medium; B: Big.

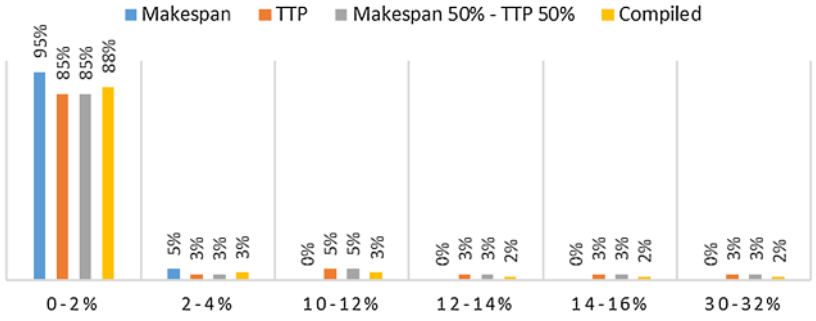



FIGURE 8. Percentage of improvement of the final objective with respect to the rules of dispatch.

with values of 0.02 and 0.05. According to the results obtained in the experimental design (see Appendix B) the following values have been selected for the memetic algorithm of the Reactive Scheduling Sub algorithm in PetNMA (Tab. 11).

The small problem always gets the best result both for the makespan and for the Total weighted tardiness in the first generation because few operations must be programmed in a few machines, so the dispatch rules get good schedules. For the total weighted tardiness, both the small problem and the large one gets good answers, the medium problem has high delays because most of the jobs start in the same stations and therefore, they are bottled in them. For the weighted objective, good results are obtained in the small and large instances, the same inconvenience of the total weighted tardiness in the medium instances is had. The results and graphs of the designs of the experiments carried out can be seen in Appendix C. Next, the results of a run of each of the 10 problems studied are shown in the Tables 12–14.

Considering the results, we can present the following conclusions of the experimental tests developed:

- 95%, 85%, and 85 of the results in the rescheduling of production with PETNMA generated improvements between 0% and 2% with respect to the values obtained by the dispatch rules for the makespan. Total weighted tardiness and the objective weighted respectively. 12% of the instances evaluated have improvements greater than 2% (see Fig. 8).
- The algorithm was able to generate different variations in the different instances and rescheduling minimized those variations. On average 58% of the results had variations between 0% and 2% with respect to the initial scheduling. 33% of the instances had variations between 2% and 60%. While 10% exceeded 100% variations (see Fig. 9).

For the weighted objective, in scenarios 1 and 4, the machine failure occurs in the first half of the initial scheduling horizon shows that the variation of the final scheduling is greater than the initial scheduling.

TABLE 12. Results of the PetNMA for the evaluation of the Makespan.

| Instance | Failure scenario | Schedule initial | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Memetic algorithm | Variation (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FMS01_6_10_19 | 1 | 412 | 468 | 468 | 468 | 1566 | 1497 | 1578 | 1000 | 452 | 9.71 |
| | 2 | 412 | 413 | 413 | 413 | 826 | 751 | 810 | 630 | 413 | 0.24 |
| | 3 | 412 | 412 | 412 | 412 | 537 | 492 | 513 | 487 | 412 | 0.00 |
| | 4 | 412 | 412 | 412 | 412 | 1162 | 1074 | 1141 | 695 | 412 | 0.00 |
| FMS02_9_8_20 | 1 | 324 | 324 | 324 | 324 | 2215 | 2009 | 2217 | 591 | 324 | 0.00 |
| | 2 | 324 | 324 | 324 | 324 | 647 | 578 | 636 | 489 | 324 | 0.00 |
| | 3 | 324 | 324 | 324 | 324 | 533 | 505 | 529 | 400 | 324 | 0.00 |
| | 4 | 324 | 324 | 375 | 324 | 1467 | 1296 | 1454 | 636 | 324 | 0.00 |
| FMS03_10_5_10 | 1 | 536 | 632 | 624 | 536 | 1711 | 1388 | 1726 | 949 | 536 | 0.00 |
| | 2 | 536 | 623 | 623 | 623 | 623 | 623 | 623 | 623 | 623 | 16.23 |
| | 3 | 536 | 536 | 536 | 536 | 573 | 573 | 573 | 537 | 536 | 0.00 |
| | 4 | 536 | 719 | 711 | 623 | 1581 | 1264 | 1522 | 1001 | 623 | 16.23 |
| FMS04_10_5_12 | 1 | 394 | 406 | 426 | 394 | 1655 | 1471 | 1675 | 1136 | 394 | 0.00 |
| | 2 | 394 | 394 | 394 | 394 | 918 | 763 | 918 | 762 | 394 | 0.00 |
| | 3 | 394 | 394 | 394 | 394 | 873 | 763 | 868 | 736 | 394 | 0.00 |
| | 4 | 394 | 417 | 432 | 401 | 2224 | 1934 | 2247 | 1162 | 394 | 0.00 |
| FMS05_10_5_12 | 1 | 379 | 437 | 472 | 406 | 1665 | 1314 | 1672 | 856 | 406 | 7.12 |
| | 2 | 379 | 408 | 408 | 408 | 417 | 408 | 408 | 417 | 408 | 7.65 |
| | 3 | 379 | 420 | 420 | 420 | 420 | 431 | 431 | 431 | 420 | 10.82 |
| | 4 | 379 | 491 | 472 | 438 | 2222 | 2048 | 2198 | 807 | 438 | 15.57 |
| FMS06_10_7_18 | 1 | 332 | 332 | 349 | 332 | 1536 | 1279 | 1541 | 581 | 332 | 0.00 |
| | 2 | 332 | 404 | 439 | 367 | 1226 | 1040 | 1238 | 573 | 359 | 8.13 |
| | 3 | 332 | 332 | 335 | 332 | 1133 | 858 | 1117 | 566 | 332 | 0.00 |
| | 4 | 332 | 332 | 351 | 333 | 2314 | 2030 | 2278 | 632 | 332 | 0.00 |
| FMS07_10_10_20 | 1 | 1012 | 1032 | 1091 | 1061 | 3878 | 3704 | 3884 | 2780 | 1032 | 1.98 |
| | 2 | 1012 | 1015 | 1051 | 1015 | 1998 | 1806 | 1969 | 1573 | 1015 | 0.30 |
| | 3 | 1002 | 1002 | 1002 | 1002 | 1808 | 1592 | 1811 | 1401 | 1002 | 0.00 |
| | 4 | 1002 | 1007 | 1089 | 1007 | 3746 | 3502 | 3686 | 2660 | 1007 | 0.50 |
| FMS08_15_5_12 | 1 | 451 | 464 | 489 | 1477 | 1806 | 1471 | 1772 | 1073 | 464 | 2.88 |
| | 2 | 451 | 544 | 544 | 544 | 544 | 544 | 544 | 544 | 544 | 20.62 |
| | 3 | 456 | 583 | 526 | 493 | 1479 | 1351 | 1523 | 1012 | 493 | 8.11 |
| | 4 | 448 | 580 | 586 | 514 | 2842 | 2441 | 2871 | 1335 | 499 | 11.38 |
| FMS09_15_6_17 | 1 | 253 | 264 | 269 | 259 | 1736 | 1513 | 1754 | 861 | 259 | 2.37 |
| | 2 | 254 | 274 | 274 | 274 | 288 | 299 | 299 | 299 | 274 | 7.87 |
| | 3 | 254 | 271 | 271 | 271 | 286 | 299 | 299 | 299 | 271 | 6.69 |
| | 4 | 254 | 273 | 276 | 259 | 1578 | 1282 | 1578 | 930 | 259 | 1.97 |
| FMS10_20_6_13 | 1 | 753 | 758 | 767 | 753 | 1634 | 1422 | 1605 | 1478 | 753 | 0.00 |
| | 2 | 753 | 804 | 839 | 782 | 1274 | 1130 | 1274 | 1293 | 782 | 3.85 |
| | 3 | 753 | 813 | 813 | 813 | 813 | 813 | 813 | 813 | 813 | 7.97 |
| | 4 | 753 | 758 | 935 | 754 | 2722 | 2387 | 2698 | 2291 | 754 | 0.13 |
| | | | | | | | | | | Average variation | 4.21 |

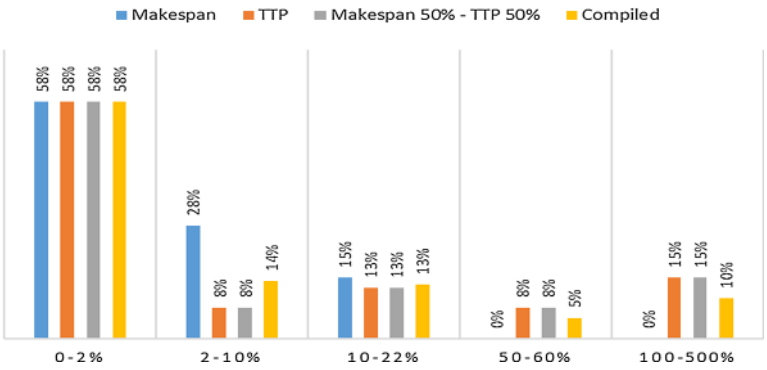



FIGURE 9. Percentage of variation of the final objective with respect to the initial scheduling.

TABLE 13. Results of the PetNMA for the evaluation of the TWT.

| Instance | Failure scenario | Schedule initial | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Memetic algorithm | Variation (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FMS01_6_10_19 | 1 | 0 | 0 | 0 | 0 | 4452 | 5962 | 5086 | 549 | 0 | 0.00 |
| | 2 | 0 | 0 | 0 | 0 | 407 | 1179 | 720 | 160 | 0 | 0.00 |
| | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0.00 |
| | 4 | 0 | 0 | 0 | 0 | 4349 | 5532 | 4951 | 978 | 0 | 0.00 |
| FMS02_9_8_20 | 1 | 0 | 0 | 0 | 0 | 9700 | 9690 | 11 525 | 2997 | 0 | 0.00 |
| | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 |
| | 3 | 0 | 0 | 0 | 0 | 78 | 0 | 0 | 0 | 0 | 0.00 |
| | 4 | 0 | 0 | 0 | 0 | 10 433 | 8994 | 11 349 | 2003 | 0 | 0.00 |
| FMS03_10_5_10 | 1 | 1577 | 1602 | 1832 | 1728 | 10 215 | 9791 | 10 539 | 4415 | 1602 | 1.59 |
| | 2 | 1577 | 1721 | 1721 | 1721 | 1721 | 1721 | 1721 | 1721 | 1721 | 9.13 |
| | 3 | 1577 | 1625 | 1625 | 1625 | 1820 | 1644 | 1820 | 1761 | 1625 | 3.04 |
| | 4 | 1577 | 1814 | 2350 | 2273 | 18 336 | 11 920 | 15 360 | 9844 | 1751 | 11.03 |
| FMS04_10_5_12 | 1 | 40 | 58 | 58 | 58 | 4188 | 5518 | 6625 | 2439 | 40 | 0.00 |
| | 2 | 40 | 79 | 64 | 79 | 808 | 1115 | 1311 | 1008 | 64 | 60.00 |
| | 3 | 40 | 108 | 108 | 108 | 1461 | 2977 | 3262 | 1208 | 108 | 170.00 |
| | 4 | 40 | 211 | 660 | 480 | 19 068 | 19 116 | 18 957 | 4644 | 211 | 427.50 |
| FMS05_10_5_12 | 1 | 155 | 390 | 558 | 694 | 16 057 | 15 124 | 14 067 | 4848 | 390 | 151.61 |
| | 2 | 155 | 326 | 326 | 326 | 587 | 794 | 794 | 587 | 326 | 110.32 |
| | 3 | 155 | 392 | 392 | 392 | 392 | 392 | 392 | 392 | 392 | 152.90 |
| | 4 | 155 | 339 | 567 | 577 | 10 366 | 11 112 | 10 916 | 6662 | 339 | 118.71 |
| FMS06_10_7_18 | 1 | 0 | 3 | 10 | 0 | 17 552 | 18 933 | 20 755 | 970 | 0 | 0.00 |
| | 2 | 0 | 14 | 14 | 14 | 38 | 92 | 92 | 26 | 14 | 1400.00 |
| | 3 | 0 | 10 | 93 | 0 | 6386 | 5559 | 6605 | 1190 | 0 | 0.00 |
| | 4 | 0 | 10 | 93 | 0 | 6313 | 6363 | 7370 | 1486 | 0 | 0.00 |
| FMS07_10_10_20 | 1 | 580 | 900 | 1216 | 1433 | 30 793 | 35 793 | 37 894 | 11 996 | 900 | 55.17 |
| | 2 | 623 | 935 | 935 | 935 | 1113 | 1572 | 1572 | 1113 | 935 | 50.08 |
| | 3 | 623 | 653 | 653 | 653 | 4551 | 5703 | 9386 | 2295 | 653 | 4.82 |
| | 4 | 565 | 806 | 743 | 565 | 12 214 | 19 047 | 20 570 | 8494 | 565 | 0.00 |
| FMS08_15_5_12 | 1 | 6302 | 8390 | 8263 | 8764 | 20 039 | 17 132 | 22 439 | 15 851 | 7120 | 12.98 |
| | 2 | 6371 | 6415 | 6424 | 6415 | 6446 | 6484 | 6441 | 6446 | 6415 | 0.69 |
| | 3 | 6304 | 6308 | 6308 | 6308 | 6351 | 6370 | 6474 | 6364 | 6308 | 0.06 |
| | 4 | 6602 | 9351 | 9319 | 9903 | 22 234 | 26 654 | 31 504 | 17 629 | 7846 | 18.84 |
| FMS09_15_6_17 | 1 | 0 | 0 | 100 | 270 | 39 606 | 30 797 | 35 605 | 11 571 | 0 | 0.00 |
| | 2 | 0 | 0 | 0 | 0 | 258 | 728 | 728 | 258 | 0 | 0.00 |
| | 3 | 0 | 0 | 0 | 0 | 816 | 1216 | 1328 | 700 | 0 | 0.00 |
| | 4 | 0 | 12 | 0 | 12 | 8784 | 8577 | 11 671 | 3643 | 0 | 0.00 |
| FMS10_20_6_13 | 1 | 6187 | 8486 | 10 114 | 12 188 | 41 807 | 61 377 | 72 272 | 15 509 | 7501 | 21.24 |
| | 2 | 6660 | 6714 | 6771 | 6714 | 8752 | 8109 | 9562 | 8898 | 6714 | 0.81 |
| | 3 | 6465 | 6475 | 6475 | 6475 | 6475 | 6475 | 6475 | 6475 | 6475 | 0.15 |
| | 4 | 6420 | 8318 | 10 100 | 11 189 | 31 016 | 45 209 | 63 663 | 19 604 | 7454 | 16.11 |
| | | | | | | | | | | Average variation | 69.92 |

The variation of the time of the major machine failure is where the variation of the final scheduling is greater than the initial scheduling. For the Total Weighted Tardiness, in scenarios 3 and 4 with the longest failure time of the machine, the variation of the final scheduling is greater than the initial scheduling (see Fig. 10).

## 6. CONCLUSION

The model is a novel proposal that integrates the Petri net technique as an effective method for production scheduling. The innovative component of the proposed model has two key elements considering the state of the art. First integrate the Petri nets with a genetic algorithm to develop active initial scheduling, second integrate the use of Memetic algorithms with Petri nets for a production rescheduling environment. The integration of Petri net starts from the generation of active schedules that guarantees closer to good results, and their configuration when the machine failure occurs allows to validation of the busiest machines and the workflow to reduce the disturbance for rescheduling. This research is considered a pioneer in a memetic algorithm decoded by Petri nets in an environment of rescheduling when there is a failure or breakdown of machines. The proposed model establishes a modification of the libraries usually used to have different ranges of scenarios. The stages

TABLE 14. Results of the PetNMA for the evaluation of the weighted objective.

| Instance | Failure scenario | Schedule initial | SPT | LPT | MRWT | EDD | CR | ATC | M. SLACK | Memetic algorithm | Variation (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FMS01_6_10_19 | 1 | 206 | 222 | 232 | 222 | 2519 | 3459 | 3126 | 1376.5 | 213 | 3.40 |
| | 2 | 206 | 206 | 206 | 206 | 296 | 368 | 370.5 | 253.5 | 206 | 0.00 |
| | 3 | 206 | 209 | 209 | 209 | 227 | 246.5 | 246.5 | 223 | 209 | 1.46 |
| | 4 | 206 | 220 | 224 | 224 | 1814.5 | 1628 | 1733.5 | 844.5 | 220 | 6.80 |
| FMS02_9_8_20 | 1 | 162 | 162 | 162 | 162 | 6355.5 | 5001.5 | 7551.5 | 790 | 162 | 0.00 |
| | 2 | 162 | 162 | 162 | 162 | 1580 | 1461 | 2017.5 | 229.5 | 162 | 0.00 |
| | 3 | 162 | 162 | 162 | 162 | 452 | 729 | 907.5 | 240 | 162 | 0.00 |
| | 4 | 162 | 168.5 | 167 | 167 | 9826.5 | 7813.5 | 9952.5 | 478.5 | 167 | 3.09 |
| FMS03_10_5_10 | 1 | 1114.5 | 1559.5 | 1686.5 | 1569 | 5821.5 | 4393 | 4736.5 | 2535.5 | 1490.5 | 33.74 |
| | 2 | 1114.5 | 1338 | 1314.5 | 1300.5 | 3011 | 2103.5 | 3672 | 1674.5 | 1300.5 | 16.69 |
| | 3 | 1123 | 1123 | 1123 | 1123 | 1594.5 | 1290 | 1614 | 1240 | 1123 | 0.00 |
| | 4 | 1114.5 | 1284 | 1481.5 | 1404.5 | 12 349 | 10 229.5 | 10 847.5 | 4929.5 | 1284 | 15.21 |
| FMS04_10_5_12 | 1 | 242.5 | 313.5 | 348 | 351 | 9730 | 9588 | 10 172 | 4516 | 271 | 11.75 |
| | 2 | 242.5 | 242.5 | 242.5 | 242.5 | 894.5 | 890.5 | 1046 | 532 | 242.5 | 0.00 |
| | 3 | 242.5 | 242.5 | 242.5 | 242.5 | 259.5 | 295 | 295 | 270 | 242.5 | 0.00 |
| | 4 | 242.5 | 344.5 | 433.5 | 296.5 | 10 458 | 9714 | 10 139 | 4654 | 284.5 | 17.32 |
| FMS05_10_5_12 | 1 | 306.5 | 389.5 | 379 | 382 | 7171 | 6262 | 6309 | 3213 | 379 | 23.65 |
| | 2 | 307 | 374 | 374 | 374 | 432 | 432 | 432 | 432 | 374 | 21.82 |
| | 3 | 306.5 | 324 | 324 | 324 | 532 | 447.5 | 447.5 | 432.5 | 324 | 5.71 |
| | 4 | 307.5 | 409 | 522 | 497.5 | 6498 | 6651 | 6458.5 | 4973 | 393 | 27.80 |
| FMS06_10_7_18 | 1 | 166 | 171 | 174.5 | 171 | 6256.5 | 6059 | 6718.5 | 2239.5 | 171 | 3.01 |
| | 2 | 166 | 192 | 180.5 | 178.5 | 3736.5 | 4332.5 | 4431.5 | 877 | 178.5 | 7.53 |
| | 3 | 166 | 182 | 190 | 182 | 2026 | 2131.5 | 2662.5 | 658.5 | 182 | 9.64 |
| | 4 | 166 | 206.5 | 198 | 175.5 | 8319 | 9067.5 | 9477 | 936.5 | 175.5 | 5.72 |
| FMS07_10_10_20 | 1 | 833 | 1064 | 1292 | 1021.5 | 7489 | 8891.5 | 8768 | 3740 | 1021.5 | 22.63 |
| | 2 | 823 | 917.5 | 917.5 | 917.5 | 1119.5 | 1342.5 | 1342.5 | 1119.5 | 917.5 | 11.48 |
| | 3 | 849 | 888 | 882.5 | 888 | 1009.5 | 1415.5 | 1611 | 1263 | 882.5 | 3.95 |
| | 4 | 838 | 1008 | 1513.5 | 1167.5 | 17 187 | 17 997.5 | 19 797 | 9656 | 885 | 5.61 |
| FMS08_15_5_12 | 1 | 386.5 | 559 | 742 | 735.5 | 16 469 | 16 095 | 19 489.5 | 4278.5 | 559 | 44.63 |
| | 2 | 390 | 465.5 | 465.5 | 465.5 | 1852.5 | 2191 | 2723.5 | 940.5 | 465.5 | 19.36 |
| | 3 | 389 | 422 | 422 | 422 | 1057.5 | 908.5 | 908.5 | 669 | 422 | 8.48 |
| | 4 | 389 | 575 | 763 | 788.5 | 5443 | 6725 | 9196.5 | 2788 | 559.5 | 43.83 |
| FMS09_15_6_17 | 1 | 128 | 195.5 | 245.5 | 227.5 | 13 500.5 | 12 903 | 13 781 | 4510.5 | 147 | 14.84 |
| | 2 | 128 | 130 | 130 | 130 | 137.5 | 145.5 | 145.5 | 142 | 130 | 1.56 |
| | 3 | 128 | 175 | 164.5 | 161 | 2961 | 2712.5 | 3650 | 1085 | 154.5 | 20.70 |
| | 4 | 128 | 130.5 | 146 | 204 | 8533.5 | 8093.5 | 9545.5 | 3243.5 | 130 | 1.56 |
| FMS10_20_6_13 | 1 | 3707.5 | 4911.5 | 6044 | 6446.5 | 16 751.5 | 25 896.5 | 28 613.5 | 8555.5 | 4479.5 | 20.82 |
| | 2 | 3589 | 3665.5 | 3665.5 | 3665.5 | 3833.5 | 3696 | 3767 | 3833.5 | 3665.5 | 2.13 |
| | 3 | 3609.5 | 3806.5 | 3924.5 | 4158 | 8959 | 7162.5 | 9012 | 6029 | 3608.5 | −0.03 |
| | 4 | 3721 | 4661 | 5643 | 6199 | 22 585 | 35 414.5 | 36 346.5 | 10 281.5 | 4147.5 | 11.46 |
| | | | | | | | | | | Average variation | 11.18 |

of failure of the machine were considered in four instances that allow the introduction or simulation of faults in different stages of production.

PetNMA is able to generate small variations in the different test instances of the algorithm. On average, 58% of the results obtained had variations in the performance measures, with respect to the initial scheduling between 0% and 2%. The initial scheduling sub-algorithm in PetNMA manages to improve the results considering the makespan and total weighted tardiness objectives, comparing them with the dispatch rules. The use of Simulated Annealing prevents the algorithm from falling into local optima. The low response variability in the rescheduling is a sample of the robustness and flexibility of the new method proposed.

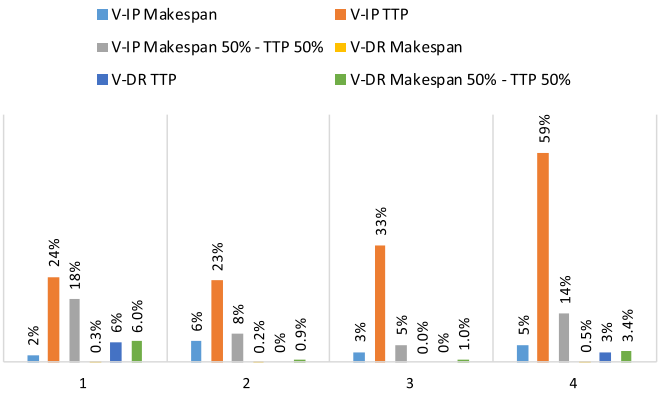



FIGURE 10. Variation of the final objective *vs.* initial scheduling and dispatch rules in machine failure scenarios.

### 6.1. Managerial insight and industrial application

There are key factors for the implementation and management of the proposed model for the reactive scheduling of production due to the occurrence of machine breakdowns in the FMS, which justify it:

- Instant when the fault occurs: when the failure occurs in the last quartile of the scheduling horizon, it is more difficult to find a good reactive production schedule. However, computational times are less when the failure happens at the beginning of production because the chromosome is more extensive.
- Repair time of the machine that fails: when the repair time is longer, the variation of the final objective with respect to the initial objective tends to be larger, because a machine is disabled for a longer time.
- The number of machines in the system: when the systems are larger, there is a greater probability of finding good reactive production schedules since the operations can be rescheduling in the other machines of the station where the machine with the fault is located.

### 6.2. Future research

As future research, the implementation of the algorithm in a real environment is proposed, due to the occurrence of machine failure. Tian *et al.* [57] describe that in practical applications of scheduling and rescheduling, optimization of multiple objectives often are involved. Another option is the inclusion of new parameters such as machine setup times depending on the sequence of operations and transport time between stations [52]. Consideration of other random events on the plant floor, such as order cancellations, new jobs, and change in processing priorities [42]. Use of new neighborhood structures for the Memetic Algorithm, as a strategy to explore the solution space of active schedules in an FMS environment. Such efforts would contribute to extend and enrich the body of knowledge in the areas of static and dynamic production scheduling in FMS [28].

*Acknowledgements.* The study is not funded by any agency. The authors do hereby declare that there is *no conflict of interest* in other works regarding the publication of this paper. This article does not contain any studies with human participants or animals performed by any of the authors.

## APPENDIX A. USED PROBLEMS

The following are the specifications of the problems used to evaluate the PetNMA algorithm with the following coding system:

FMSXX_J_S_M with: XX. Identification of the Problem Number (01, 02, ...).

J. Number of Jobs to Schedule.
S. Number of stations in the FMS.
M. Total number of machines in the FMS.

To represent the characteristics of the system and each of the works, the following structures are used:

– Vectors two positions to characterize the information of each station.

(Identification of the Station, Number of Machines in the Station)

– Three-position vectors to characterize the job information.

(Id, Time Availability, Delivery Date)

– Vectors two positions opposite each job to characterize the operations information. The order in which the operations are recorded in front of the works, represent their sequence by the stations.

(Station, Processing Time)

**FMS01_6_10_19**

Six (6) jobs, ten (10) stations and a total of nineteen (19) machines.

(W001, 2)(W002, 2)(W003, 1)(W004, 2)(W005, 2)(W006, 1)(W007, 3)(W008, 2)(W009, 1)(W0010, 3)

(1, 0, 397) (W008, 23)(W002, 44)(W009, 34)(W004, 48)(W001, 43)(W003, 28)(W006, 42)(W005, 44)(W010, 41)(W007, 50)
(2, 0, 325) (W001, 44)(W008, 39)(W010, 34)(W006, 36)(W005, 21)(W007, 24)(W004, 37) (W002, 35)(W009, 35)(W003, 20)
(3, 0, 373) (W003, 33)(W006, 35)(W002, 23)(W010, 44)(W009, 31)(W004, 41)(W007, 42) (W008, 49)(W001, 49)(W005, 26)
(4, 0, 313) (W009, 29)(W008, 39)(W003, 20)(W007, 32)(W005, 25)(W004, 27)(W010, 42) (W006, 45)(W001, 28)(W002, 26)
(5, 0, 294) (W006, 23)(W001, 47)(W008, 31)(W005, 20)(W010, 30)(W009, 20)(W003, 25) (W002, 31)(W007, 46)(W004, 21)
(6, 0, 390) (W005, 46)(W002, 41)(W006, 48)(W008, 24)(W001, 44)(W003, 48)(W007, 24) (W004, 26)(W009, 40)(W010, 49)

**FMS02_9_8_20**

Nine (9) jobs, eight (8) stations and a total of twenty (20) machines.

(W001, 2)(W002, 3)(W003, 2)(W004, 3)(W005, 2)(W006, 3)(W007, 3)(W008, 2)

(1, 0, 310) (W001, 40)(W006, 42)(W005, 31)(W003, 24)(W008, 42)(W007, 43)(W004, 38)(W002, 50)
(2, 0, 235) (W003, 27)(W001, 31)(W008, 23)(W007, 34)(W006, 28)(W005, 29)(W002, 40) (W004, 23)
(3, 0, 314) (W003, 42)(W001, 49)(W002, 42)(W008, 40)(W007, 31)(W004, 32)(W005, 28) (W006, 50)
(4, 0, 250) (W002, 25)(W004, 32)(W006, 21)(W005, 21)(W007, 48)(W008, 33)(W001, 22) (W003, 48)
(5, 0, 271) (W004, 31)(W002, 40)(W005, 38)(W001, 38)(W006, 28)(W003, 29)(W007, 33) (W008, 34)
(6, 0, 254) (W003, 29)(W005, 37)(W006, 20)(W008, 36)(W004, 29)(W007, 45)(W001, 24) (W002, 34)
(7, 0, 265) (W004, 50)(W002, 42)(W003, 35)(W007, 29)(W006, 21)(W008, 25)(W005, 35) (W001, 28)
(8, 0, 272) (W006, 35)(W004, 32)(W002, 34)(W003, 40)(W001, 46)(W005, 40)(W007, 24) (W008, 21)
(9, 0, 269) (W007, 38)(W008, 40)(W003, 35)(W004, 31)(W001, 32)(W005, 24)(W006, 25) (W002, 44)

**FMS04_10_5_12**

Ten (10) works, five (5) stations and a total of twelve (12) machines.

(W001, 3)(W002, 2)(W003, 2)(W004, 3)(W005, 2)

(1, 0, 231) (W001, 20)(W004, 87)(W002, 31)(W005, 76)(W003, 17)
(2, 0, 180) (W005, 25)(W003, 32)(W001, 24)(W002, 18)(W004, 81)
(3, 0, 280) (W002, 72)(W003, 23)(W005, 28)(W001, 58)(W004, 99)
(4, 0, 394) (W003, 86)(W002, 76)(W005, 97)(W001, 45)(W004, 90)
(5, 0, 180) (W005, 27)(W001, 42)(W004, 48)(W003, 17)(W002, 46)
(6, 0, 231) (W002, 67)(W001, 98)(W005, 48)(W004, 27)(W003, 62)
(7, 0, 180) (W005, 28)(W002, 12)(W004, 19)(W001, 80)(W003, 50)
(8, 0, 280) (W002, 63)(W001, 94)(W003, 98)(W004, 50)(W005, 80)
(9, 0, 394) (W005, 14)(W001, 75)(W003, 50)(W002, 41)(W004, 55)
(10, 0, 180) (W005, 72)(W003, 18)(W002, 37)(W004, 79)(W001, 61)

**FMS05_10_5_12**

Ten (10) jobs, five (5) stations and a total of twelve (12) machines.

(W001, 2)(W002, 3)(W003, 3)(W004, 2)(W005, 2)

(1, 0, 272) (W002, 23)(W003, 45)(W001, 82)(W005, 84)(W004, 38)
(2, 0, 159) (W003, 21)(W002, 29)(W001, 18)(W005, 41)(W004, 50)
(3, 0, 212) (W003, 38)(W004, 54)(W0005, 16)(W001, 52)(W002, 52)
(4, 0, 284) (W005, 37)(W001, 54)(W003, 74)(W002, 62)(W004, 57)
(5, 0, 297) (W005, 57)(W001, 81)(W002, 61)(W004, 68)(W003, 30)
(6, 0, 349) (W005, 81)(W001, 79)(W002, 89)(W003, 89)(W004, 11)
(7, 0, 230) (W004, 33)(W003, 20)(W001, 91)(W005, 20)(W002, 66)
(8, 0, 203) (W005, 24)(W002, 84)(W001, 32)(W003, 55)(W004, 8)
(9, 0, 220) (W005, 56)(W001, 7)(W004, 54)(W003, 64)(W002, 39)
(10, 0, 157) (W005, 40)(W002, 83)(W001, 19)(W003, 8)(W004, 7)

### FMS06_10_7_18

Ten (10) jobs, seven (7) stations and a total of eighteen (18) machines.

(W001, 2)(W002, 2)(W003, 3)(W004, 2)(W005, 3)(W006, 3)(W007, 3)
(1, 0, 272) (W007, 25)(W006, 44)(W005, 45)(W004, 42)(W001, 27)(W002, 49)(W003, 40)
(2, 0, 287) (W005, 40)(W007, 28)(W004, 46)(W006, 46)(W003, 41)(W002, 39)(W001, 47)
(3, 0, 247) (W003, 32)(W001, 49)(W002, 39)(W004, 43)(W005, 20)(W006, 35)(W007, 29)
(4, 0, 259) (W004, 26)(W005, 25)(W003, 35)(W001, 46)(W002, 50)(W007, 35)(W006, 42)
(5, 0, 244) (W004, 26) (W005, 29)(W006, 27)(W001, 43)(W002, 44)(W007, 37)(W003, 38)
(6, 0, 272) (W003, 48) (W007, 36)(W004, 21)(W001, 42)(W005, 36)(W002, 46)(W006, 43)
(7, 0, 217) (W005, 38) (W006, 21)(W004, 20)(W002, 40)(W003, 36)(W001, 36)(W007, 26)
(8, 0, 271) (W005, 39) (W007, 49)(W004, 37)(W006, 36)(W002, 42)(W001, 42)(W003, 26)
(9, 0, 271) (W004, 32) (W003, 39)(W007, 22)(W001, 46)(W005, 42)(W006, 40)(W002, 50)
(10, 0, 258) (W005, 46) (W007, 42)(W003, 50)(W002, 20)(W006, 20)(W004, 39)(W001, 41)

### FMS07_10_10_20

Ten (10) jobs, ten (10) stations and a total of twenty (20) machines.

(W001, 2)(W002, 1)(W003, 2)(W004, 2)(W005, 3)(W006, 2)(W007, 3)(W008, 2)(W009, 2)(W010, 1)
(1, 0, 523) (W001, 72) (W002, 64)(W003, 55)(W004, 31)(W005, 53)(W006, 95)(W007, 11)(W008, 52)(W009, 6)(W010, 84)
(2, 0, 667) (W001, 61) (W004, 27)(W005, 88)(W003, 78)(W002, 49)(W006, 83)(W009, 91) (W007, 74)(W008, 29)(W010, 87)
(3, 0, 489) (W001, 86) (W004, 32)(W002, 35)(W003, 37)(W006, 18)(W005, 48)(W007, 91) (W008, 52)(W010, 60)(W009, 30)
(4, 0, 509) (W001, 8) (W002, 82)(W005, 27)(W004, 99)(W007, 74)(W006, 9)(W003, 33)(W0010, 20)(W008, 59)(W009, 98)
(5, 0, 476) (W002, 50) (W001, 94)(W006, 43)(W004, 62)(W005, 55)(W008, 487)(W003, 5)(W009, 36)(W010, 47)(W007, 36)
(6, 0, 407) (W001, 53) (W007, 30)(W003, 7)(W004, 12)(W002, 68)(W009, 87)(W005, 28) (W010, 70)(W008, 45)(W006, 7)
(7, 0, 502) (W003, 29) (W004, 96)(W001, 99)(W002, 14)(W005, 34)(W008, 14)(W006, 7) (W007, 76)(W009, 57)(W010, 76)
(8, 0, 695) (W003, 90) (W001, 19)(W004, 87)(W005, 51)(W002, 84)(W006, 45)(W010, 84)(W007, 58)(W008, 81)(W009, 96)
(9, 0, 617) (W003, 97) (W002, 99)(W005, 93)(W001, 38)(W008, 13)(W006, 96)(W004, 40)(W010, 64)(W007, 32)(W009, 45)
(10, 0, 524) (W003, 44) (W001, 60)(W009, 29)(W004, 5)(W007, 74)(W002, 85)(W005, 34)(W008, 95)(W010, 51)(W006, 47)

### FMS08_15_5_12

Fifteen (15) jobs, five (5) stations and a total of twelve (12) machines.

(W001, 3)(W002, 2)(W003, 2)(W004, 2)(W005, 3)
(1, 0, 258) (W002, 21)(W003, 34)(W005, 95)(W001, 53)(W004, 55)
(2, 0, 186) (W004, 52)(W005, 16)(W002, 71)(W003, 26)(W001, 21)
(3, 0, 222) (W003, 31)(W001, 12)(W002, 42)(W004, 39)(W005, 98)
(4, 0, 354) (W004, 77)(W002, 77)(W005, 79)(W001, 55)(W003, 66)
(5, 0, 237) (W005, 37)(W004, 34)(W003, 64)(W002, 19)(W001, 83)
(6, 0, 330) (W003, 43)(W002, 54)(W001, 92)(W004, 62)(W005, 79)
(7, 0, 413) (W001, 93)(W004, 69)(W002, 87)(W005, 77)(W003, 87)
(8, 0, 246) (W001, 60)(W002, 41)(W003, 38)(W005, 83)(W004, 24)
(9, 0, 233) (W003, 98)(W004, 17)(W005, 25)(W001, 44)(W002, 49)
(10, 0, 370) (W001, 96)(W005, 77)(W004, 79)(W002, 75)(W003, 43)
(11, 0, 241) (W005, 28)(W003, 35)(W001, 95)(W004, 76)(W002, 7)
(12, 0, 210) (W001, 61)(W005, 10)(W003, 95)(W002, 9)(W004, 35)
(13, 0, 271) (W005, 59)(W004, 16)(W002, 91)(W003, 59)(W001, 46)
(14, 0, 200) (W005, 43)(W002, 52)(W001, 28)(W003, 27)(W004, 50)
(15, 0, 221) (W001, 87)(W002, 45)(W003, 39)(W005, 9)(W004, 41)

### FMS09_15_6_17

Fifteen (15) jobs, six (6) stations and a total of seventeen (17) machines.

(W001, 3)(W002, 3)(W003, 2)(W004, 2)(W005, 3)(W006, 4)
(1, 0, 156) (W001, 24)(W004, 20)(W002, 13)(W003, 27)(W006, 38)(W005, 34)
(2, 0, 181) (W006, 32)(W004, 18)(W002, 37)(W001, 32)(W005, 23)(W003, 39)
(3, 0, 116) (W003, 15)(W004, 11)(W006, 10)(W002, 29)(W005, 13)(W001, 38)
(4, 0, 213) (W005, 27)(W006, 18)(W003, 47)(W002, 28)(W001, 50)(W004, 43)
(5, 0, 169) (W004, 39)(W005, 26)(W003, 14)(W002, 14)(W001, 31)(W006, 45)
(6, 0, 177) (W003, 32)(W001, 50)(W005, 23)(W006, 28)(W004, 19)(W002, 25)
(7, 0, 170) (W002, 49)(W003, 38)(W004, 23)(W005, 14)(W001, 32)(W006, 14)
(8, 0, 210) (W006, 29)(W001, 15)(W003, 48)(W002, 50)(W005, 36)(W004, 32)
(9, 0, 222) (W006, 44)(W005, 28)(W004, 45)(W001, 26)(W002, 47)(W003, 32)
(10, 0, 192) (W005, 23)(W004, 50)(W006, 36)(W003, 25)(W002, 30)(W001, 28)
(11, 0, 148) (W001, 40)(W002, 24)(W004, 18)(W005, 12)(W006, 17)(W003, 37)
(12, 0, 153) (W006, 21)(W002, 31)(W005, 12)(W003, 30)(W001, 27)(W004, 32)
(13, 0, 227) (W005, 35)(W003, 10)(W006, 50)(W001, 49)(W002, 35)(W004, 48)
(14, 0, 201) (W006, 45)(W004, 35)(W001, 30)(W002, 29)(W003, 31)(W005, 31)
(15, 0, 130) (W001, 22)(W003, 25)(W002, 13)(W004, 38)(W006, 17)(W005, 15)

### FMS10_20_6_13

Twenty (20) jobs, six (6) stations and a total of thirteen (13) machines.

(W001, 2)(W002, 1)(W003, 3)(W004, 3)(W005, 3)(W006, 1)
(1, 0, 178) (W005, 40)(W003, 33)(W006, 26)(W001, 22)(W002, 26)(W004, 31)
(2, 0, 239) (W004, 50)(W002, 34)(W005, 32)(W003, 46)(W001, 38)(W006, 39)
(3, 0, 248) (W005, 31)(W004, 24)(W003, 47)(W001, 49)(W006, 47)(W002, 50)
(4, 0, 218) (W001, 38)(W003, 43)(W004, 37)(W005, 36)(W006, 38)(W002, 26)
(5, 0, 204) (W005, 21)(W002, 50)(W001, 29)(W003, 35)(W006, 47)(W004, 22)
(6, 0, 191) (W003, 31)(W001, 27)(W005, 44)(W002, 21)(W004, 39)(W006, 29)
(7, 0, 206) (W004, 33)(W005, 22)(W001, 28)(W002, 45)(W006, 39)(W003, 39)
(8, 0, 211) (W001, 45)(W002, 21)(W005, 46)(W006, 33)(W004, 30)(W003, 36)
(9, 0, 171) (W001, 36)(W003, 33)(W005, 24)(W002, 21)(W006, 35)(W004, 22)
(10, 0, 196) (W004, 40)(W003, 30)(W002, 32)(W006, 36)(W005, 24)(W001, 34)

(11, 0, 241) (W006, 44)(W002, 42)(W001, 24)(W003, 36)(W005, 46)(W004, 49)
(12, 0, 170) (W004, 23)(W005, 30)(W001, 37)(W003, 30)(W006, 21)(W002, 29)
(13, 0, 223) (W005, 48)(W003, 47)(W004, 31)(W001, 26)(W006, 28)(W002, 43)
(14, 0, 210) (W006, 36)(W003, 33)(W004, 37)(W001, 23)(W005, 46)(W002, 35)
(15, 0, 193) (W003, 27)(W002, 40)(W004, 30)(W006, 42)(W001, 24)(W005, 30)
(16, 0, 245) (W001, 44)(W005, 36)(W006, 42)(W004, 40)(W002, 48)(W003, 35)
(17, 0, 223) (W003, 27)(W004, 32)(W002, 43)(W005, 37)(W001, 50)(W006, 34)
(18, 0, 174) (W003, 20)(W002, 25)(W001, 29)(W005, 26)(W004, 30)(W006, 44)
(19, 0, 204) (W001, 31)(W002, 23)(W004, 22)(W005, 38)(W006, 50)(W003, 40)
(20, 0, 188) (W005, 26)(W004, 22)(W001, 20)(W006, 43)(W002, 29)(W003, 48)

## Appendix B. Design of experiments sub algorithm initial scheduling

The experimental tests for the parameterization of the initial scheduling sub-algorithm were made in 3 previously selected problems. The small problem yielded the same results regardless of the combination of parameters, in which experiments in medium and large targets problems are studied. Design experiments were performed in Statgraphics Centurion software (Figs. B.1 and B.2, Tabs. B.1–B.6).

Table B.1. Experiment designs (Makespan – Medium Problem – No. 7): analysis of variance improved.

| Source | Sum of squares | Df | Mean square | $F$-ratio | $P$-value |
|---|---|---|---|---|---|
| Main effects | | | | | |
| A: Intensity of mutation | 334.389 | 2 | 167.194 | 2.03 | 0.1494 |
| B: Percentage of elitism | 29.5556 | 2 | 14.7778 | 0.18 | 0.8366 |
| C: Probability of crossing | 100.0 | 1 | 100.0 | 1.21 | 0.2795 |
| D: Probability of mutation | 484.0 | 1 | 484.0 | 5.88 | 0.0218 |
| Interactions | | | | | |
| Residual | 2387.28 | 29 | 82.3199 | | |
| Total (Corrected) | 3335.22 | 35 | | | |

Table B.2. Experiment designs (Makespan – Big Problem – No. 9): analysis of variance improved.

| Source | Sum of squares | Df | Mean square | $F$-ratio | $P$-value |
|---|---|---|---|---|---|
| Main effects | | | | | |
| A: Intensity of mutation | 9.05556 | 2 | 4.52778 | 1.27 | 0.2969 |
| B: Percentage of elitism | 0.388889 | 2 | 0.194444 | 0.05 | 0.9472 |
| C: Probability of crossing | 13.4444 | 1 | 13.4444 | 3.76 | 0.0622 |
| D: Probability of mutation | 16.0 | 1 | 16.0 | 4.48 | 0.0431 |
| Interactions | | | | | |
| Residual | 103.667 | 29 | 3.57471 | | |
| Total (Corrected) | 142.556 | 35 | | | |

TABLE B.3. Experiment designs (TWT – Medium and Big Problem): analysis of variance improved.

| Source | Sum of squares | Df | Mean square | $F$-ratio | $P$-value |
|---|---|---|---|---|---|
| Main effects | | | | | |
| A: Intensity of mutation | 38 848.2 | 2 | 19 424.1 | 2.50 | 0.0998 |
| B: Percentage of elitism | 155 545 | 2 | 77 772.3 | 10.00 | 0.0005 |
| C: Probability of crossing | 2025.0 | 1 | 2025.0 | 0.26 | 0.6137 |
| D: Probability of mutation | 13 301.8 | 1 | 13 301.8 | 1.71 | 0.2012 |
| Interactions | | | | | |
| Residual | 225 483 | 29 | 7775.29 | | |
| Total (Corrected) | 435 203 | 35 | | | |

TABLE B.4. Experiment designs (TWT – Big Problem): analysis of variance improved.

| Source | Sum of squares | Df | Mean square | $F$-ratio | $P$-value |
|---|---|---|---|---|---|
| Main effects | | | | | |
| A: Intensity of mutation | 6.23389 | 2 | 3.11694 | 3.59 | 0.0304 |
| B: Percentage of elitism | 0.842222 | 2 | 0.421111 | 0.49 | 0.5205 |
| C: Probability of crossing | 2.77778 | 1 | 2.77778 | 3.20 | 0.0741 |
| D: Probability of mutation | 3.86778 | 1 | 3.86778 | 4.46 | 0.0335 |
| Interactions | | | | | |
| Residual | 25.1672 | 29 | 0.867835 | | |
| Total (Corrected) | 38.9 | 35 | | | |

TABLE B.5. Design of experiments for combination of the Weighted Objective ($\alpha$ Makespan + $\beta$ TWT) in the Medium Problem (7).

| Source/ Main effects | $P$-value 10–90 | $P$-value 20–80 | $P$-value 30–70 | $P$-value 40–60 | $P$-value 50–50 | $P$-value 60–40 | $P$-value 70–30 | $P$-value 80–20 | $P$-value 90–10 |
|---|---|---|---|---|---|---|---|---|---|
| A: Intensity of mutation | 0.0715 | 0.0108 | 0.7328 | 0.7271 | 0.1118 | 0.0311 | 0.0235 | 0.0000 | 0.0711 |
| B: Percentage of elitism | 0.0051 | 0.0255 | 0.0000 | 0.0262 | 0.0236 | 0.0449 | 0.0668 | 0.1574 | 0.6365 |
| C: Probability of crossing | 0.8857 | 0.3592 | 0.4780 | 0.0078 | 0.8120 | 0.9359 | 0.0114 | 0.0006 | 0.2208 |
| D: Probability of mutation | 0.5909 | 0.0622 | 0.0145 | 0.1377 | 0.8205 | 0.6596 | 0.9695 | 0.5104 | 0.6769 |

TABLE B.6. Design of experiments for combination of the Weighted Objective ($\alpha$ Makespan + $\beta$ TWT) in the Big Problem (9).

| Source/ Main effects | *P*-value 10–90 | *P*-value 20–80 | *P*-value 30–70 | *P*-value 40–60 | *P-value* 50–50 | *P*-value 60–40 | *P*-value 70–30 | *P*-value 80–20 | *P*-value 90–10 |
|---|---|---|---|---|---|---|---|---|---|
| A: Intensity of mutation | 0.0404 | 0.0629 | 0.2188 | 0.1948 | 0.2808 | 0.7184 | 0.9123 | 0.5951 | 0.1834 |
| B: Percentage of elitism | 0.6205 | 0.0354 | 0.0017 | 0.0611 | 0.0002 | 0.0122 | 0.0434 | 0.3536 | 0.8937 |
| C: Probability of crossing | 0.0841 | 0.0587 | 0.0614 | 0.3431 | 0.8013 | 0.4845 | 0.1843 | 0.0579 | 0.0024 |
| D: Probability of mutation | 0.0435 | 0.0805 | 0.0168 | 0.0173 | 0.8013 | 0.4078 | 0.1291 | 0.0089 | 0.5186 |

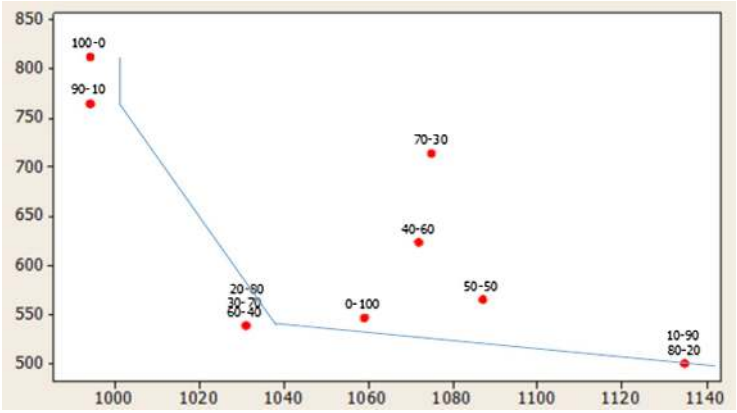



FIGURE B.1. Pareto diagram (Makespan ($x$) *vs.* TWT ($y$)) – Medium Problem.

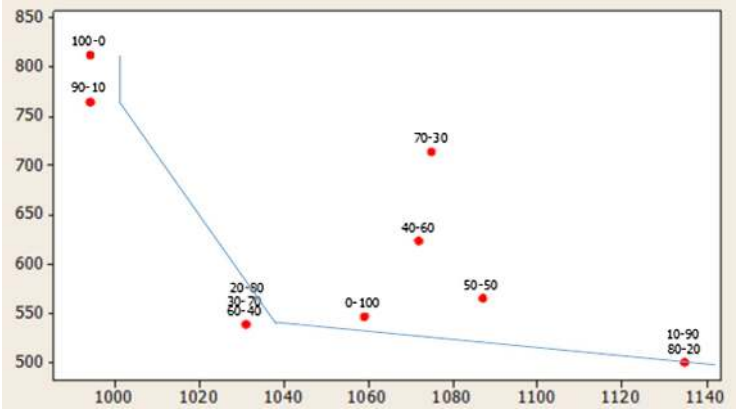



FIGURE B.2. Pareto diagram (Makespan ($x$) *vs.* TWT ($y$)) – Big Problem.

## Appendix C. Design of experiments of the reactive scheduling sub algorithm

The experimental tests for the parameterization of the reactive scheduling sub-algorithm were made in 3 previously selected problems used for the initial scheduling sub-algorithm tests. Design experiments were performed in Statgraphics Centurion software (Tabs. C.1–C.10).

Table C.1. Robust design parameters for rescheduling.

| Parameters of PetNMA | Nomenclature | Values |
|---|---|---|
| Number of generations | maxgen | 100; 200 |
| Size of the population | $P$ | 50; 100 |
| Probability of crossing | $Tc$ | 0.6; 0.9 |
| Probability of mutation | $Tm$ | 0.1; 0.5 |
| Intensity of mutations | $Im$ | 1; 2; 4 |
| Elitism | $Te_1$ | Off. On (50%) |
| Decrease factor | | 0.01; 0.1 |
| Instant of failure | | 1; 2 |
| Repair time variation | | 0.02; 0.05 |
| Average repair time | | 0.5; 1.0 |

Table C.2. Robust experiment design results (Taguchi) for the Makespan evaluation of the small problem.

| Noise factor | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| A | B | C | D | E | F | G | | | | | | | |
| 100 | 50 | 60 | 10 | 1 | 0 | 0.01 | 2.78 | 13.58 | 11.73 | 0.00 | 7.02 | 6.65 | −19.16 |
| 100 | 50 | 60 | 50 | 2 | 50 | 0.1 | 5.25 | 0.00 | 0.00 | 4.01 | 2.31 | 2.72 | −10.38 |
| 100 | 100 | 90 | 10 | 1 | 50 | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 100 | 100 | 90 | 50 | 2 | 0 | 0.01 | 5.25 | 0.00 | 0.31 | 0.00 | 1.39 | 2.58 | −8.39 |
| 200 | 50 | 90 | 10 | 2 | 0 | 0.1 | 0.00 | 0.00 | 6.17 | 5.56 | 2.93 | 3.40 | −12.37 |
| 200 | 50 | 90 | 50 | 1 | 50 | 0.01 | 0.00 | 4.94 | 0.00 | 0.00 | 1.23 | 2.47 | −7.85 |
| 200 | 100 | 60 | 10 | 2 | 50 | 0.01 | 0.00 | 0.00 | 2.78 | 0.00 | 0.69 | 1.39 | −2.85 |
| 200 | 100 | 60 | 50 | 1 | 0 | 0.1 | 6.79 | 7.10 | 5.56 | 0.00 | 4.86 | 3.31 | −15.03 |

TABLE C.3. Robust experiment design results (Taguchi) for the Makespan evaluation of the medium problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 1 | 0 | 0.01 | 9.08 | 7.06 | 0.00 | 6.18 | | 5.58 | 3.91 | −16.30 |
| 100 | 50 | 60 | 50 | 2 | 50 | 0.1 | 14.39 | 0.00 | 13.35 | 4.41 | | 8.04 | 6.98 | −20.05 |
| 100 | 100 | 90 | 10 | 1 | 50 | 0.1 | 9.58 | 0.00 | 2.92 | 12.07 | | 6.14 | 5.63 | −17.89 |
| 100 | 100 | 90 | 50 | 2 | 0 | 0.01 | 1.98 | 0.30 | 0.00 | 0.50 | | 0.69 | 0.88 | −0.26 |
| 200 | 50 | 90 | 10 | 2 | 0 | 0.1 | 7.37 | 0.00 | 0.00 | 0.00 | | 1.84 | 3.68 | −11.33 |
| 200 | 50 | 90 | 50 | 1 | 50 | 0.01 | 7.92 | 2.93 | 1.98 | 7.65 | | 5.12 | 3.11 | −15.24 |
| 200 | 100 | 60 | 10 | 2 | 50 | 0.01 | 0.98 | 6.08 | 4.86 | 2.80 | | 3.68 | 2.25 | −12.39 |
| 200 | 100 | 60 | 50 | 1 | 0 | 0.1 | 13.06 | 0.00 | 7.28 | 0.00 | | 5.09 | 6.33 | −17.48 |

TABLE C.4. Robust experiment design results (Taguchi) for the Makespan evaluation of the big problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 1 | 0 | 0.01 | 2.37 | 7.87 | 6.69 | 1.97 | | 4.73 | 3.00 | −14.63 |
| 100 | 50 | 60 | 50 | 2 | 50 | 0.1 | 5.20 | 3.59 | 9.45 | 4.00 | | 5.56 | 2.68 | −15.60 |
| 100 | 100 | 90 | 10 | 1 | 50 | 0.1 | 1.60 | 14.96 | 2.36 | 5.14 | | 6.02 | 6.15 | −18.10 |
| 100 | 100 | 90 | 50 | 2 | 0 | 0.01 | 4.02 | 16.54 | 12.20 | 3.92 | | 9.17 | 6.26 | −20.55 |
| 200 | 50 | 90 | 10 | 2 | 0 | 0.1 | 6.30 | 0.00 | 15.75 | 9.84 | | 7.97 | 6.59 | −19.83 |
| 200 | 50 | 90 | 50 | 1 | 50 | 0.01 | 7.48 | 16.21 | 9.84 | 2.36 | | 8.97 | 5.74 | −20.22 |
| 200 | 100 | 60 | 10 | 2 | 50 | 0.01 | 4.33 | 3.15 | 9.88 | 2.39 | | 4.94 | 3.39 | −15.19 |
| 200 | 100 | 60 | 50 | 1 | 0 | 0.1 | 3.60 | 11.76 | 14.57 | 1.97 | | 7.98 | 6.14 | −19.63 |

TABLE C.5. Robust experiment design results (Taguchi) for the TWT evaluation of the small problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 100 | 100 | 90 | 10 | 2 | 50 | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 200 | 50 | 90 | 50 | 2 | 50 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |
| 200 | 100 | 60 | 50 | 2 | 0 | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 |

TABLE C.6. Robust experiment design results (Taguchi) for the TWT evaluation of the medium problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 2 | 0 | 0.01 | 110.43 | 0.00 | 16.23 | 161.63 | | 72.07 | 77.04 | −39.84 |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 45.18 | 8.36 | 49.06 | 110.59 | | 53.30 | 42.37 | −36.22 |
| 100 | 100 | 90 | 10 | 2 | 50 | 0.1 | −3.07 | 47.11 | 7.35 | 235.01 | | 71.60 | 111.06 | −41.58 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 50.00 | 5.63 | 67.22 | 33.11 | | 38.99 | 26.24 | −33.09 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 120.65 | 16.73 | 0.00 | 79.87 | | 54.31 | 56.02 | −37.25 |
| 200 | 50 | 90 | 50 | 2 | 50 | 0.01 | 64.53 | 8.62 | 5.81 | 94.22 | | 43.30 | 43.40 | −35.17 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 30.47 | 0.55 | 3.75 | 99.67 | | 33.61 | 46.04 | −34.34 |
| 200 | 100 | 60 | 50 | 2 | 0 | 0.1 | 55.17 | 50.08 | 4.82 | 0.00 | | 27.52 | 29.13 | −31.44 |

TABLE C.7. Robust experiment design results (Taguchi) for the TWT evaluation of the big problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 2 | 0 | 0.01 | 2100 | 0 | 0 | 2100 | | 1050.00 | 1212.44 | −63.43 |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 8100 | 3300 | 8400 | 7500 | | 6825.00 | 2379.60 | −77.06 |
| 100 | 100 | 90 | 10 | 2 | 50 | 0.1 | 2100 | 0 | 5100 | 3900 | | 2775.00 | 2223.17 | −70.57 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 0 | 5400 | 0 | 4700 | | 2525.00 | 2929.59 | −71.08 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 0 | 0 | 0 | 1300 | | 325.00 | 650.00 | −56.26 |
| 200 | 50 | 90 | 50 | 2 | 50 | 0.01 | 0 | 0 | 0 | 0 | | 0.00 | 0.00 | 0.00 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 11400 | 0 | 6000 | 2100 | | 4875.00 | 5010.24 | −76.29 |
| 200 | 100 | 60 | 50 | 2 | 0 | 0.1 | 100 | 0 | 600 | 1200 | | 475.00 | 550.00 | −56.56 |

TABLE C.8. Robust experiment design results (Taguchi) for the Weighted Objective evaluation of the Small problem.

| Noise factor | | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 2 | 0 | 0.01 | 0.00 | 12.65 | 11.42 | 4.32 | | 7.10 | 5.99 | −18.88 |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 4.32 | 2.16 | 10.49 | 0.00 | | 4.24 | 4.52 | −15.23 |
| 100 | 100 | 90 | 10 | 2 | 50 | 0.1 | 0.00 | 0.00 | 12.65 | 0.00 | | 3.16 | 6.33 | −16.02 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 0.00 | 13.89 | 3.09 | 0.00 | | 4.24 | 6.59 | −17.04 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 0.00 | 0.00 | 0.00 | 3.09 | | 0.77 | 1.54 | −3.77 |
| 200 | 50 | 90 | 50 | 2 | 50 | 0.01 | 5.25 | 0.00 | 1.54 | 8.33 | | 3.78 | 3.75 | −13.95 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 0.00 | 0.00 | 12.65 | 0.00 | | 3.16 | 6.33 | −16.02 |
| 200 | 100 | 60 | 50 | 2 | 0 | 0.1 | 0.00 | 13.89 | 3.09 | 0.00 | | 4.24 | 6.59 | −17.04 |

TABLE C.9. Robust experiment design results (Taguchi) for the Weighted Objective evaluation of the Medium problem.

| Noise factor | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 2 | 0 | 0.01 | 11.28 | 1.31 | 4.36 | 87.61 | 26.14 | 41.19 | −32.91 |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 50.15 | 11.33 | 40.96 | 12.54 | 28.74 | 19.77 | −30.49 |
| 100 | 100 | 90 | 10 | 2 | 50 | 0.1 | 30.15 | 0.45 | 1.73 | 30.69 | 15.75 | 16.94 | −26.66 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 22.63 | 11.48 | 3.95 | 5.61 | 10.92 | 8.45 | −22.37 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 11.74 | 56.55 | 48.13 | 9.88 | 31.58 | 24.24 | −31.58 |
| 200 | 50 | 90 | 50 | 2 | 50 | 0.01 | 5.50 | 9.69 | 25.80 | 32.86 | 18.46 | 12.99 | −26.70 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 90.57 | 4.24 | 0.00 | 20.81 | 28.90 | 42.08 | −33.35 |
| 200 | 100 | 60 | 50 | 2 | 0 | 0.1 | 33.86 | 31.67 | 5.06 | 107.44 | 44.51 | 43.95 | −35.35 |

TABLE C.10. Robust experiment design results (Taguchi) for the Weighted Objective evaluation of the Big problem.

| Noise factor | | | | | | H | 1 | 2 | 2 | 1 | Mean | Standard deviation | Noise/Signal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | I | 0.5 | 0.5 | 1.0 | 1.0 | | | |
| | | | | | | J | 0.02 | 0.05 | 0.02 | 0.05 | | | |
| Control factor | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | | | | | $\bar{Y}$ | $S$ | $-10\log[\frac{1}{n}\sum_{i=1}^{n} Y_i^2]$ |
| 100 | 50 | 60 | 10 | 1 | 0 | 0.01 | 1.95 | 0.00 | 17.51 | 44.92 | 16.10 | 20.75 | −27.65 |
| 100 | 50 | 60 | 50 | 4 | 50 | 0.1 | 7.03 | 7.00 | 12.11 | 145.70 | 42.96 | 68.54 | −37.30 |
| 100 | 100 | 90 | 10 | 1 | 50 | 0.1 | 21.40 | 0.00 | 0.00 | 69.14 | 22.64 | 32.60 | −31.17 |
| 100 | 100 | 90 | 50 | 4 | 0 | 0.01 | 27.48 | 61.87 | 20.70 | 147.27 | 64.33 | 58.15 | −38.24 |
| 200 | 50 | 90 | 10 | 4 | 0 | 0.1 | 27.13 | 9.38 | 17.69 | 39.45 | 23.41 | 12.92 | −28.28 |
| 200 | 50 | 90 | 50 | 1 | 50 | 0.01 | 14.84 | 1.56 | 20.70 | 1.56 | 9.67 | 9.66 | −22.13 |
| 200 | 100 | 60 | 10 | 4 | 50 | 0.01 | 43.58 | 15.23 | 2.34 | 7.42 | 17.14 | 18.40 | −27.39 |
| 200 | 100 | 60 | 50 | 1 | 0 | 0.1 | 22.27 | 0.00 | 3.83 | 24.22 | 12.58 | 12.44 | −24.38 |

## REFERENCES


[1] J. Acevedo and G. Mejía, Programación reactiva y robusta de la producción en un ambiente sistema de manufactura flexible: llegada de nuevas órdenes y cambios en la prioridad de las órdenes de trabajo. Maestria, Departamento de Ingeniería Industrial, Universidad de los Andes (2006).

[2] N. Al-Hinai and T. El-Mekkawy, An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem. *Flexible Serv. Manuf. J.* **23** (2011) 64–85.

[3] Ö. Başak and Y.E. Albayrak, Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems. *Comput. Ind. Eng.* **86** (2015) 116–126.

[4] O.T. Baruwa, M.A. Piera and A. Guasch, Deadlock-free scheduling method for flexible manufacturing systems based on timed colored Petri nets and anytime heuristic search. *IEEE Trans. Syst. Man Cybern.* **45** (2014) 831–846.

[5] J.A. Chedid, K.S. Navarro, H.O. Mateus and A.S. Mercado, Reprogramación de producción en cadenas de suministro colaborativas: Una revisión de la literatura. *Espacio* **38** (2017) 23.

[6] F. Chen and J. Chen, Performance modelling and evaluation of dynamic tool allocation in flexible manufacturing systems using coloured Petri nets: an object-oriented approach. *Int. J. Adv. Manuf. Technol.* **21** (2003) 98–109.

[7] J. Chen and F.F. Chen, Adaptive scheduling in random flexible manufacturing systems subject to machine breakdowns. *Int. J. Prod. Res.* **41** (2003) 1927–1951.

[8] H. Cho, Petri net models for message manipulation and event monitoring in an FMS cell. *Int. J. Prod. Res.* **36** (1998) 231–250.

[9] C. Cotta, L. Mathieson and P. Moscato, Memetic algorithms. In: Handbook of Heuristics edited by R. Martí, P. M. Pardalos, M. G. C. Resende. Springer, Cham (2018) 607–638.

[10] B.K. Dey, S. Pareek, M. Tayyab and B. Sarkar, Autonomation policy to control work-in-process inventory in a smart production system. *Int. J. Prod. Res.* (2020) 1–23. DOI: 10.1080/00207543.2020.1722325.

[11] J. Dorn, R. Kerr and G. Thalhammer, Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning. *Int. J. Human-Comput. Stud.* **42** (1995) 687–704.

[12] H.A. Elmaraghy and T.Y. Elmekkawy, Deadlock-free rescheduling in flexible manufacturing systems. *CIRP Ann.* **51** (2002) 371–374.

[13] M.R. Garey, D.S. Johnson and R. Sethi, The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1** (1976) 117–129.

[14] P. Garg, A comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. *Int. J. Network Secur. App.* **1** (2009) 34–42.

[15] A. Gholami, R. Sheikh, N. Mizani and S.S. Sana, ABC analysis of the customers using axiomatic design and incomplete rough set. *RAIRO:OR* **52** (2018) 1219–1232.

[16] R. Guchhait, B.K. Dey, S. Bhuniya, B. Ganguly, B. Mandal, R.K. Bachar, B. Sarkar, H.M. Wee and K. Chaudhuri, Investment for process quality improvement and setup cost reduction in an imperfect production process with warranty policy and shortages. *RAIRO:OR* **54** (2020) 251–266.

[17] L. Han, K. Xing, X. Chen, H. Lei and F. Wang, Deadlock-free genetic scheduling for flexible manufacturing systems using Petri nets and deadlock controllers. *Int. J. Prod. Res.* **52** (2014) 1557-1572.

[18] L. Han, K. Xing, X. Chen and F. Xiong, A Petri net-based particle swarm optimization approach for scheduling deadlock-prone flexible manufacturing systems. *J. Intel. Manuf.* **29** (2018) 1083–1096.

[19] S.K. Hasan, R. Sarker and D. Essam, Genetic algorithm for job-shop scheduling with machine unavailability and breakdowns. *Int. J. Prod. Res.* **49** (2011) 4999–5015.

[20] I. Hatono, K. Yamagata and H. Tamura, Modeling and online scheduling of flexible manufacturing systems using stochastic Petri nets. *IEEE Trans. Softw. Eng.* **17** (1991) 126–132.

[21] H. Hoos and T. Stutzle, Stochastic Local Search: Foundations and Applications. Morgan Kaufmann Publishers, San Francisco, CA (2005).

[22] B. Huang, Z. Cai, M. Zhou and J. Hao, Scheduling of FMS based on binary decision diagram and Petri net. In: IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). (2018) 1–6.

[23] G. Jamali, S.S. Sana and R. Moghdani, Hybrid improved cuckoo search algorithm and genetic algorithm for solving markov-modulated demand. *RAIRO:OR* **52** (2018) 473–497.

[24] A. Khanna, P. Gautam, B. Sarkar and C.K. Jaggi, Integrated vendor–buyer strategies for imperfect production systems with maintenance and warranty policy. *RAIRO:OR* **54** (2020) 435–450.

[25] Y.W. Kim, T. Suzuki and T. Narikiyo, FMS scheduling based on timed Petri Net model and reactive graph search. *Appl. Math. Modell.* **31** (2007) 955–970.

[26] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing. *Science* **220** (1983) 671–680.

[27] R.R. Kumar, A.K. Singh and M. Tiwari, A fuzzy based algorithm to solve the machine-loading problems of a FMS and its neuro fuzzy Petri net model. *Int. J. Adv. Manuf. Technol.* **23** (2004) 318–341.

[28] B.S. Kumar, G.J. Raju and G.R. Janardhana, Production planning in flexible manufacturing system by considering the multi-objective functions. In: Advances in Materials and Manufacturing Engineering. Lecture Notes in Mechanical Engineering. (ICAMME 2019) Springer, Singapore (2020) 343–352.

[29] H. Lei, K. Xing, L. Han and Z. Gao, Hybrid heuristic search approach for deadlock-free scheduling of flexible manufacturing systems using Petri nets. *Appl. Soft Comput.* **55** (2017) 413–423.

[30] C. Li, W. Wu, Y. Feng and G. Rong, Scheduling FMS problems with heuristic search function and transition-timed Petri nets. *J. Intell. Manuf.* **26** (2015) 933–944.

[31] C. Low and T.H. Wu, Mathematical modelling and heuristic approaches to operation scheduling problems in an FMS environment. *Int. Jo. Prod. Res.* **39** (2001) 689–708.

[32] D.R. Mahapatra, S. Panda and S.S. Sana, Multi-choice and stochastic programming for transportation problem involved in supply of foods and medicines to hospitals with consideration of logistic distribution. *RAIRO:OR* **54** (2020) 1119–1132.

[33] S.V. Mehta, Predictable scheduling of a single machine subject to breakdowns. *Int. J. Comput. Integr. Manuf.* **12** (1999) 15–38.

[34] G. Mejia and J. Acevedo, Reactive scheduling in FMS: an integrated approach based on petri nets, genetic algorithms and simulation. In: Third International Conference on Production Research, Americas' Region (2006).

[35] G. Mejía and K. Niño, A new Hybrid Filtered Beam Search algorithm for deadlock-free scheduling of flexible manufacturing systems using Petri Nets. *Comput. Ind. Eng.* **108** (2017) 165–176.

[36] U. Mishra, J.Z. Wu and B. Sarkar, A sustainable production-inventory model for a controllable carbon emissions rate under shortages. *J. Cleaner Prod.* **256** (2020) 120268.

[37] P. Moscato and C. Cotta, Una introducción a los algoritmos meméticos. *Intel. Artif. Rev. Iberoamericana Intel. Artif.* **7** (2003) 131–148.

[38] T. Murata, Petri nets: properties, analysis and applications. *Proc. IEEE* **77** (1989) 541–580.

[39] H.Z. Nabi and T. Aized, Performance evaluation of a carousel configured multiple products flexible manufacturing system using Petri net. *Oper. Manag. Res.* **13** (2020) 109–129.

[40] K.S. Navarro, J.A. Chedid, G.M. Arquez, W.F. Florez, H.O. Mateus, S.S. Sana and L.E. Cárdenas-Barrón, An EPQ inventory model considering an imperfect production system with probabilistic demand and collaborative approach. *J. Adv. Manage. Res.* **17** (2019) 282–304.

[41] K.S. Navarro, J.A. Chedid, W.F. Florez, H.O. Mateus, L.E. Cárdenas-Barrón and S.S. Sana, A collaborative EPQ inventory model for a three-echelon supply chain with multiple products considering the effect of marketing effort on demand. *J. Ind. Manage. Optim.* **16** (2020) 1613–1633.

[42] P.S. Pachpor, R. Shrivastava, D. Seth and S. Pokharel, Application of Petri nets towards improved utilization of machines in job shop manufacturing environments. *J. Manuf. Technol. Manage.* **28** (2017) 169–188.

[43] A.M. Patel and A.Y. Joshi, Modeling and analysis of a manufacturing system with deadlocks to generate the reachability tree using Petri net system. *Proc. Eng.* **64** (2013) 775–784.

[44] M. Pinedo, Scheduling. Springer, Cham (2012).

[45] M. Pinedo, Planning and Scheduling in Manufacturing and Services. Springer, New York, NY (2005).

[46] S.S. Sana, H. Ospina-Mateus, F.G. Arrieta and J.A. Chedid, Application of genetic algorithm to job scheduling under ergonomic constraints in manufacturing industry. *J. Ambient Intell. Humanized Comput.* **10** (2019) 2063–2090.

[47] S.S. Sankar, S. Ponnanbalam and C. Rajendran, A multiobjective genetic algorithm for scheduling a flexible manufacturing system. *Int. J. Adv. Manuf. Technol.* **22** (2003) 229–236.

[48] S.S. Sankar, S. Ponnambalam and M. Gurumarimuthu, Scheduling flexible manufacturing systems using parallelization of multi-objective evolutionary algorithms. *Int. J. Adv. Manuf. Technol.* **30** (2006) 279–285.

[49] B. Santosa, R. Damayanti and B. Sarkar, Solving multi-product inventory ship routing with a heterogeneous fleet model using a hybrid cross entropy-genetic algorithm: a case study in Indonesia *Prod. Manuf. Res.* **4** (2016) 90–113.

[50] N. Saxena, B. Sarkar and S. Singh, Selection of remanufacturing/production cycles with an alternative market: a perspective on waste management. *J. Cleaner Prod.* **245** (2020) 118935.

[51] B.K. Sett, S. Sarkar and B. Sarkar, Optimal buffer inventory and inspection errors in an imperfect production system with preventive maintenance. *Int. J. Adv. Manuf. Technol.* **90** (2017) 545–560.

[52] M. Souier, M. Dahane and F. Maliki, An NSGA-II-based multiobjective approach for real-time routing selection in a flexible manufacturing system under uncertainty and reliability constraints. *Int. J. Adv. Manuf. Technol.* **100** (2019) 2813–2829.

[53] A.A. Taleizadeh, L.E. Cárdenas-Barrón and B. Mohammadi, A deterministic multi product single machine EPQ model with backordering, scraped products, rework and interruption in manufacturing process. *Int. J. Prod. Econ.* **150** (2014) 9–27.

[54] Y. Tanimizu, T. Sakaguchi, K. Iwamura and N. Sugimura, Evolutional reactive scheduling for agile manufacturing systems. *Int. J. Prod. Res.* **44** (2006) 3727–3742.

[55] E.S. Tashnizi, S. Farahani and A.F. Nahrekhalaji, Production process optimization in flexible manufacturing system using Petri nets. In: Proceedings of the World Congress on Engineering and Computer Science (2008) 22–24.

[56] M. Tayyab, J. Jemai, H. Lim and B. Sarkar, A sustainable development framework for a cleaner multi-item multi-stage textile production system with a process improvement initiative. *J. Cleaner Prod.* **246** (2020) 119055.

[57] S. Tian, T. Wang, L. Zhang and X. Wu, Real-time shop floor scheduling method based on virtual queue adaptive control: algorithm and experimental results. *Measurement* **147** (2019) 106689.

[58] F. Tüysüz and C. Kahraman, Modeling a flexible manufacturing cell using stochastic Petri nets with fuzzy parameters. Expert Syst. App. **37** (2010) 3910–3920.

[59] M. Ullah and B. Sarkar, Recovery-channel selection in a hybrid manufacturing-remanufacturing production model with RFID and product quality. *Int. J. Prod. Econ.* **219** (2020) 360–374.

[60] G.E. Vieira, J.W. Herrmann and E. Lin, Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *J. Schedul.* **6** (2003) 39–62.

[61] Z. Zhao, G. Zhang and Z. Bing, Scheduling optimization for FMS based on Petri net modeling and GA. In: IEEE International Conference on Automation and Logistics (ICAL) (2011) 422–427.