Tailored Genetic Algorithm for Scheduling Jobs and Predictive Maintenance in a Permutation Flowshop

Asma Ladj*, Fatima Benbouzid-Si Tayeb * and Christophe Varnier †
* Laboratoire des Méthodes de Conception de Systèmes (LMCS)
Ecole nationale Supérieure d'Informatique (ESI)
BP 68M-16270 Oued Smar, Alger-Algérie, http://www.esi.dz
E-mails: {a_ladj,f_sitayeb}@esi.dz
†Institut FEMTO-ST UMR 6174, ENS2M, CNRS
Univ. Bourgogne Franche-Comté
F-25000 Besançon-France
E-mail: christophe.varnier@ens2m.fr

***Abstract*—We tackle in this paper the Permutation Flowshop Scheduling Problem (PFSP) with predictive maintenance interventions. The objective is to propose an integrated model that coordinates production schedule and predictive maintenance planning so that the total time to complete the schedule after predictive maintenance insertion is minimized. Predictive maintenance interventions are scheduled based on Prognostics and Health Management (PHM) results using a new proposed heuristic. To jointly establish an integrated scheduling of production jobs and predictive maintenance actions, we propose a tailored genetic algorithm incorporating properly designed operators. Computational experiments carried out on Taillard well known benchmarks, to which we add both PHM and maintenance data, show the efficiency of the newly proposed maintenance planning heuristic and genetic algorithm.**

I. Introduction

Production equipments reliability and availability has always been a major issue for manufacturers. Thus, attention to the maintenance activity has rapidly increased as an inevitable reality in industry. The earliest maintenance technique is corrective maintenance which takes place only after breakdowns. To ensure a satisfactory level of reliability and reduce failure risk, concept of maintenance before failure has emerged to propose a type called systematic preventive maintenance. However, maintenance efficiency is an important economical issue and a bad choice can lead to an over cost that is not acceptable. Two drifts can be observed in this case. The first occurs when the maintenance frequency is too high inducing an excessive cost of useless interventions. The second occurs when the time between two successive maintenance interventions is too long, consequently failures cannot be avoided resulting in system shutdown. Hence, maintenance decision must be taken according to the health status of the production system. Thereby, a more efficient maintenance approach called ”*predictive maintenance*“ was proposed to prevent equipment from risk of breakdowns while significantly reduce cost by reducing the number of unnecessary interventions. In this field, industrials show a growing interest in Prognostics and Health Management (PHM) thematic [1]. Indeed, rather than understanding a failure which has just appeared (diagnosis process), it seems convenient to anticipate its occurrence, i.e. *prognostic process*, in order to resort to protective actions [2]. Given the current machine condition and past operation profile, prognosis module is able to estimate time to failure called the *Remaining Useful Life (RUL)* [3].

PHM benefits are strongly tied to the decision making following the assimilation of prognostic outputs. By prognostic post decision, we mean the integration of the PHM results, i.e. the estimated RUL, in the system schedule. Post prognostic decision is still a novel axis of development and few works have been led. In fact, conflicts could occur between predictive maintenance planning and production scheduling if a maintenance action is programmed when equipment is used for production. Production and maintenance activities are inter-dependent. Hence, solving production scheduling and maintenance planning problems independently ignores these inherent conflicts. Hence, *integrated scheduling of production operations and predictive maintenance activities* must be properly established. Note that researches on predictive maintenance integration in the solution of scheduling problems of any type are still scarce. [4] proposed a mathematical programming for scheduling model of jobs and predictive maintenance on a single machine to minimize the maximum tardiness. A mixed integer linear model for solving flowshop scheduling problem with predictive maintenance was developed by [5] where machines are able to switch between two production modes: nominal and sub-nominal. [6] proposed three heuristics for parallel machines scheduling problem with predictive maintenance. The same problem was studied by [7] in order to maximize the global useful life of the system under service constraints. For scheduling jobs and predictive maintenance on a single deteriorating machine, [8] proposed a mixed integer linear programming and an integrated prognostics-based genetic algorithm with the objective of minimizing the total predictive maintenance cost. [9] proposed an integrated model in which the job scheduling, predictive maintenance, prognostic information and resource planning are proactively determined simultaneously for a single machine. A case study resolved using a genetic algorithm was used to demonstrate the proposed model effectiveness.

In this work, we deal with flowshop, one of the most extensively studied scheduling problem with a strong engineering background [10]. A study on flowshop scheduling problem with predictive maintenance was proposed by [5]. However, the defined mixed integer program allows to solve only small instances and is not able to compute the optimal solution for instances with a significant number of jobs and machines. To deal with larger instances, [11] proposed a variable neighbour search algorithm to obtain near optimal solutions. However, to take into account the several sources of uncertainty in the prognosis process, authors model PHM outputs using fuzzy logic. In this paper, we propose a population based metaheuristic based on genetic algorithm [12] to solve production and predictive maintenance scheduling problem. Moreover, to take into consideration the variable operating conditions of machines, we suppose in this study that, due to various deterioration levels, the corresponding RUL and degradation value for each machine depends on the job being processed.

The remaining content of the paper is organized as follows. In section 2, we present the integrated scheduling problem. Section 3 is devoted to describe the proposed genetic algorithm. Next, experiments results are discussed in section 4. Finally, we draw some concluding remarks and perspectives.

II. Problem statement

In this section, we describe first the integrated scheduling problem we are dealing with: the permutation flowshop scheduling problem with predictive maintenance. Next, objective function is defined.

A. Integrated scheduling problem

This paper addresses the Permutation Flowshop Scheduling Problem (PFSP), a more restricted version of the flowshop problem, under availability constraints where machines are subject to predictive maintenance operations. It consists in determining jointly the best sequencing of jobs and predictive maintenance actions to be processed for each machine in order to optimize the total time to complete the schedule C_{max} taking into account predictive maintenance operations planning. The problem can be formally defined as follows. A finite set $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ of n jobs must be processed on a finite set $\mathcal{M}$ of m machines $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. The processing order of operations on machines is the same for all jobs, i.e. all jobs are processed by all machines in the same order. All jobs are available and ready for processing at time zero. Each job $J_j, j \in \{1, \ldots, n\}$, requires a fixed and known amount of processing time on each machine $M_i, i \in \{1, \ldots, m\}$; represented by p_{ij}. Furthermore, preemption is disallowed, i.e., any started operation has to be continuously executed without a break.

In order to improve machines availability and avoid downtime and inopportune spending, predictive maintenance interventions must be planned using machines remaining useful life (RULs) and degradation values. Indeed, given the current machine health state and the operating environment, the associated PHM module is able to predict the evolution of degradation phenomena. As diverse jobs are being processed by the machines, every kind of job causes various levels of damage on each machine. When the accumulated degradation of a machine reaches a maximal threshold Δ, a predictive maintenance intervention must be performed to restore it to its initial state. Processing times of predictive maintenance operations are noted p_i^{PM}.

In our paper, contrary to the classical PFSP, machines are not continuously available due to predictive maintenance operations. The PFSP under availability constraints is NP-hard (Non deterministic Polynomial time hard), since the two-machine flow shop scheduling problem is NP-hard in the strong sense if two availability periods are considered as proved by [13].

The studied problem is modeled on the following assumptions :

- At any given time, each machine can handle at most one activity (production or predictive maintenance) and each production job can be processed by at most one machine;
- A deteriorating prognosis system provides RUL_{ij} corresponding to each job J_j when being processed by machine M_i. RUL_{ij} represents the period during which the "as good as new" machine M_i could achieve job J_j before failure. The PHM module is also able to provide an associated degradation value δ_{ij} (see Fig. 1) ;
- $\delta_{ij} \in]0;1[$ represents the wear and tear of the machine M_i when only job J_j is processed during the processing time p_{ij} (0 no degradation, 1 full degradation) ;
- $\delta_{ij} = f(p_{ij})$ where f characterizes the evolution of the machine degradation. To seek simplicity, we consider in our model that for each job J_{ij} is expressed by $\delta_{ij} = p_{ij}/RUL_{ij}$.
- We fix the full potential, i.e. the maximal degradation threshold, of the machine $\Delta = 1$. When exceeding this limit, the failure risk increases;
- During the planning horizon, at least one predictive maintenance operation is performed on each machine and no predictive maintenance operation is performed after the processing of last job;
- After a predictive maintenance operation, the machine is recovered to be "as good as new".

The resulting integrated scheduling is denoted $\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$ where π_i represents the corresponding integrated scheduling for machine M_i (see Fig. 2). π_i can be seen as a succession of several production blocks separated by predictive maintenance operations: $\pi_i = \{\mathcal{B}_{i1}, \mathcal{M}_{i1}, \mathcal{B}_{i2}, \ldots, \mathcal{M}_{i(l_i-1)}, \mathcal{B}_{il_i}\}$, where:

- $\mathcal{B}_{ik}$ is the k^{th} production block processed by machine M_i;
- $\mathcal{M}_{ik}$ is the k^{th} predictive maintenance operation performed in machine M_i;
- l_i is the number of blocks required to process all jobs by machine $M_i : \bigcup_{k=1}^{l_i} \mathcal{B}_{ik} = \mathcal{J}$.

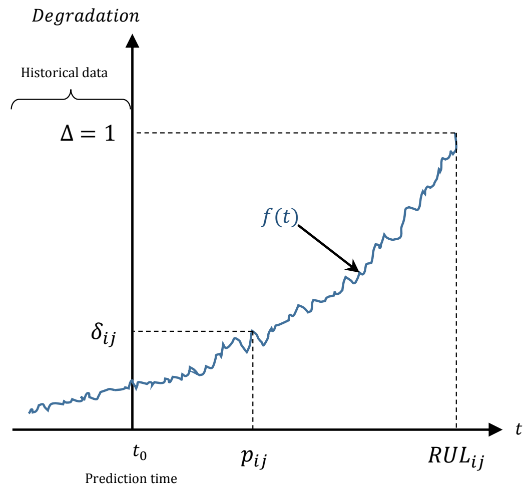

Fig. 1. Degradation estimation

B. Objective function

Our objective is to find an integrated sequence of production jobs and predictive maintenance operations that minimizes the completion time of the last processed job when the schedule also includes predictive maintenance operations, referred to as $makespan$ or C_{max}. Given a sequence $S = \{J_{s_1}, \ldots, J_{s_n}\}$ where J_{s_j} indicates the j^{th} job to be executed, taking into account predictive maintenance interventions planned on each machine, $C_{max} = C(J_{s_n}, m)$ where $C(J_{s_n}, m)$ is the completion time of the last job of the production sequence on the last machine M_m after predictive maintenance operations insertion.

III. The proposed Genetic Algorithm

Genetic Algorithms (*GAs*) were widely used to solve production scheduling problems [14]. They were next successfully applied to production scheduling with systematic preventive maintenance [15], [16], [17], [18]. We develop here a genetic algorithm for the integrated production and predictive maintenance scheduling problem with C_{max} minimization. The most significant characteristics of our proposed GA are:

1) The use of a unique representation of integrated solution regrouping both production and predictive maintenance data;
2) The proposition of a new predictive maintenance planning heuristic that inserts maintenance operations on each machine only when necessary to guarantee an efficient exploitation based on PHM results;
3) The design of well chosen operators and the sensitive calibration of its parameters;
4) The use of an embedded restart mechanism to avoid premature convergence of the search process.

A. The encoding scheme and fitness function

The proposed representation is an adaptation of the work of [16]. A solution π is coded by a two-field structure. The first field is a sequence S representing the execution order of production jobs by machines $S = \{J_{s_1}, \ldots, J_{s_n}\}$ where J_{s_j} indicates the j^{th} job to be executed. The second field is a binary matrix PM of size $m \times n$ that represents the scheduling of predictive maintenance operations on each machine. $PM[i, j] = 0$ indicates that no maintenance operation is planned on the i^{th} machine after the j^{th} production job in the sequence S and $PM[i, j] = 1$ indicates that there is one.

For the example in Fig. 3, given a sequence S of 10 jobs to be processed by 3 machines, $PM\ [3, 2] = 1$ indicates that a predictive maintenance intervention is planned on machine M_3 after the 2nd job in the sequence S, i.e. J_9.

To evaluate the quality of an individual in the population, GA needs a fitness function. Giving a candidate solution Sol, the objective function is to minimize $f(Sol) = C_{max}(Sol)$ and the fitness function is the reciprocal of $f(Sol)$:

$$Affinity(Sol) = 1/f(Sol) \quad (1)$$

B. The generation of the initial population

Instead of starting with an initial population randomly generated, it is more efficient to use special techniques to produce a higher quality initial population [14]. In this paper, we propose a two-step initialization procedure where an initial population of $PopSize$ individuals is generated as follows:

- Step 1. In this step, we generate three parts of the initial population:
  1) The first part contains the first production sequence generated using the well-known NEH (*Nawaz, Enscore and Ham*) heuristic [19] with C_{max} minimization for PFSP;
  2) The second part is made up the $\alpha\% PopSize$ production sequences generated using the modified NEH heuristic proposed by [14]. Thus, we ensure that these parts of the population is formed by fit members;
  3) The third part (the largest one) contains the remaining $(100 - \alpha\%) PopSize - 1$ randomly generated production sequences.
- Step 2. In this second step, the predictive maintenance operations are inserted in the production sequences that are generated in Step 1 using a new heuristic that we propose. Maintenance operations will be planned starting from the first machine M_1 until the last one M_n. The main idea of our proposed heuristic is to guaranty an efficient exploitation of machines by planning maintenance interventions only when necessary. The job sequence S is scanned and maintenance operations are inserted according to the current cumulative degradation δ with respect to the threshold $\Delta = 1$ (see Algorithm 1).
  - If $\delta < \Delta$, i.e. the full degradation has not been achieved yet, we move to the next job in the sequence S without inserting a predictive maintenance operation;
  - When $\delta > \Delta$, which means that the complete degradation is exceeded, we insert a maintenance

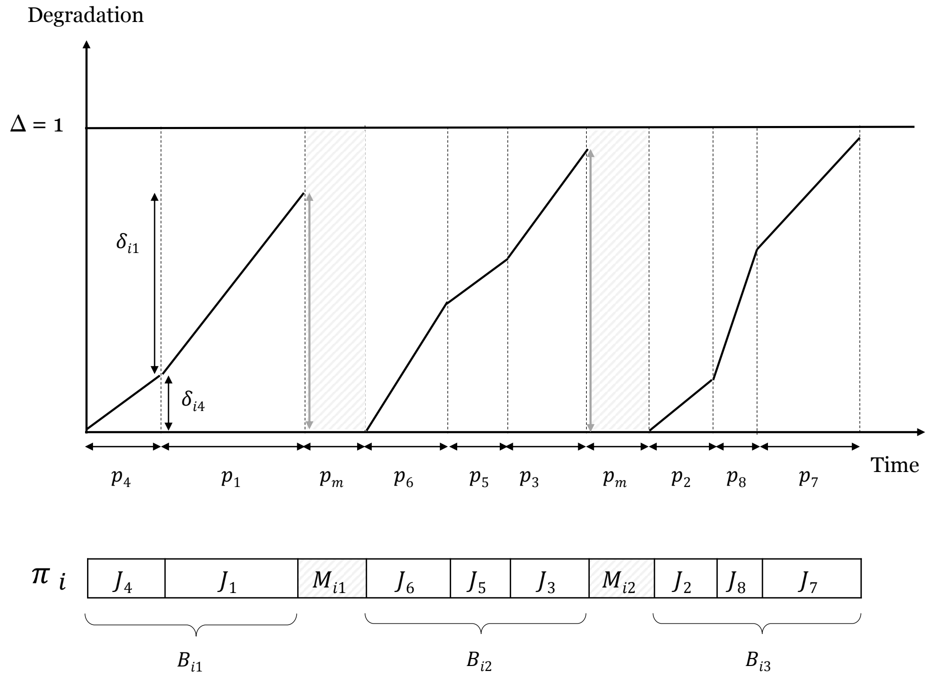

Fig. 2. Gantt diagram for integrated scheduling π_i of machine M_i

operation in the best position that optimizes C_{max}, i.e. before or after the last production job added.

Algorithm 1 Insert_PM(π)

```
1: $S = (J_{s_1}, \ldots, J_{s_n})$ // job sequence associated to $\pi$
2: $PM$ // maintenance matrix associated to $\pi$
3: for machine $M_i$ i=1 to m do
4:     $\delta$=0 // accumulated degradation
5:     for jobs $J_{s_j}$ j=1 to n do
6:         $\delta$=$\delta$+$\delta_{s_j,i}$
7:         if $\delta > \Delta$ then
8:             Compute $C_{max}$ when inserting PM before $J_{s_j}$ (early intervention) and after $J_{s_j}$ (tardy intervention) and choose the one optimizing the best $C_{max}$
9:             Update PM // $PM[i, s_{j-1}] = 1$ or $PM[i, s_j] = 1$
10:            $\delta$=0
11:        end if
12:    end for
13: end for
14: return $\pi$
```

We note that, the integrated solution obtained after predictive maintenance insertion in the sequence generated by NEH heuristic is called *I_NEH*.

C. The population improvement

The effectiveness of GAs greatly depends on the correct choice of the selection, crossover and mutation operators, as well as the probabilities by which they are applied. In the following subsections, we detailed the specification of the different genetic operators we designed.

$$S = [1\,9\,3\,8\,5\,6\,7\,4\,2\,0]$$

$$PM = \begin{bmatrix} 0\,1\,0\,0\,1\,0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \end{bmatrix}$$

M_1	p_{11}	p_{91}	p_{11}^{PM}	p_{31}	p_{81}	p_{51}	p_{21}^{PM}	p_{61}	p_{71}	p_{41}	p_{21}	p_{31}^{PM}	p_{01}

M_2	p_{12}	p_{92}	p_{32}	p_{12}^{PM}	p_{82}	p_{52}	p_{62}	p_{22}^{PM}	p_{72}	p_{42}	p_{22}	p_{02}

M_3	p_{13}	p_{93}	p_{13}^{PM}	p_{33}	p_{83}	p_{53}	p_{63}	p_{23}^{PM}	p_{73}	p_{43}	p_{23}	p_{03}

Fig. 3. Example of integrated solution representation

1) Selection operator: The first parent is selected using the *2-tournament* classical scheme [20]. The second parent is picked either by roulette wheel selection [20] during the first 40% iterations, or by random selection during the next iterations. This selection scheme ensures that at least one of the selected parents is a fit one. Furthermore, we seek by this strategy to accelerate the convergence of the research process when it starts, then its diversification is promoted using a random method.

2) Crossover operator: It generates new individuals by combining selected parents with a probability $CrossProb$. Many different general and specific crossover operators have been proposed for the PFSP [14]. Since we deal with integrated scheduling problem, our study include four types of crossover operators :

- Crossovers on production sequences. We use the classical *2X* [20] and the *SJOX (Similar Job Order Crossover)* scheme proved to be effective for the PFSP [14].

- Crossover on maintenance emplacements. We use the *1HX (1 point Horizontal crossover)* which deals with maintenance matrix [16]. The first child, respectively the second, inherits the job sequence of the first parent, respectively the second. First, a random horizontal cut point C is chosen (between 1 and m). Then, lines above the cut point (maintenance emplacements for machines $M_i, 1 \leqslant i < C$) of the first parent, respectively the second, are copied to the first child, respectively the second. In the same way, lines under the cut point (maintenance emplacements for machines $M_i, C \leqslant i \leqslant m$) of the first parent, respectively the second, are copied to the second child, respectively the first.

- Crossover on both production and maintenance. The previous crossover schemes deal with either production sequences or maintenance matrix exclusively. Hence, we propose a new crossover *(IntX* for *Integrated crossover)* which acts on both production and maintenance structures. We design a crossover operator so as to allow children to jointly inherit production and maintenance characteristics from parents. Job sequences of offspring are the results of 1 point crossover (*1X*) on parents' production sequences. Whereas their maintenance matrix are obtained using horizontal crossover (*1HX*) on parents maintenance matrix.

3) Mutation operator: We incorporate a mutation operator to avoid convergence to local optimum. Children suffer a mutation with a probability $MutProb$. Maintenance emplacements are retained whereas job sequence is updated by swapping two jobs that are randomly selected from either the same production block, i.e. no maintenance operation exists between them, or from different production blocks.

D. The population renewal

1) Population replacement: Individuals of the next generation are selected from a pool gene which combines the previous parent population and the newly created children. 40% of the best individuals are directly inserted in the new population, consequently, the best solutions are preserved to the next generation. Then, we complete the rest of the population by randomly chosen members from the pool gene. This allows a certain percentage of chromosomes with poor fitness values to be included in the population so that there will be a better chance of breaking out of local minima.

2) Restart scheme: A common problem of GAs is that the population could sometimes stall around a local optimum. To avoid that, a restart mechanism can be embedded into the GA [14]. If the best solution found doesn't improve after 10% of the algorithm iterations, we update the current population as follow so as to improve its diversity. The best $20\% \times PopSize$ individuals of the population are skipped, $20\% \times PopSize$ individuals from the remaining ones suffer a swap or shift mutation [20], $20\% \times PopSize$ individuals are generated using modified NEH heuristic and the rest ($40\% \times PopSize$) are replaced by newly randomly created members.

Our algorithm terminates after $MaxGen$ generations.

IV. Computational results

In this section, we present the results of computational experiments we conducted to test the newly proposed GA on a PC with Intel Xeon ® E5 and 28.00 GB RAM. We will first describe test data generation. Secondly, we present the calibration step results. Next, we analyze performance of our newly proposed algorithm. Finally, we discuss the computational time of the proposed algorithm.

A. Data generation

We use the well-known Taillard benchmarks [21] available in the literature to which we add both PHM and maintenance data. Our test bed consists of a total of 11 subsets, where each contains 10 instances with the same size $n \times m$ where $n \in \{20, 50, 100, 200\}$ and $m \in \{5, 10, 20\}$. From each partial instance we build a complete instance by adding PHM and maintenance data as follow.

We have three kinds of jobs :

- jobs inducing small machine deterioration where the degradation δ_{ij} is generated from a uniform distribution $U[1\%, 2\%]$;
- jobs inducing medium machine deterioration generated from a uniform distribution $U[2\%, 5\%]$;
- jobs inducing big machine deterioration generated from a uniform distribution $U[5\%, 10\%]$.

Moreover, we define three (03) maintenance modes:

- Mode 1 represents small maintenance interventions with processing time p_i^{PM} generated from a uniform distribution $U[1, 19]$.
- Mode 2 represents medium maintenance interventions with processing time p_i^{PM} generated from a uniform distribution $U[50, 99]$.
- Mode 3 represents long maintenance interventions with processing time p_i^{PM} generated from a uniform distribution $U[100, 200]$.

We have then 3 possible configurations which will be run on the 11 subsets of 10 instances. There is therefore a set of 330 different instances to perform the tests. For each instance, 5 executions are carried out, and in all 1650 executions for the proposed GA. We average the results for all the given instances.

For evaluation, we discuss two different metrics listed below:

- RPD : the relative percentage deviation of the C_{max} provided by each algorithm (with predictive maintenance interventions) C_{max}^{sol} with respect to the best known solution for Taillard instance C_{max}^{best} expressed as follows:

$$RPD = \frac{C_{max}^{sol} - C_{max}^{best}}{C_{max}^{best}} \times 100 \qquad (2)$$

- CPU : computation times of the proposed algorithm.

B. GA calibration

For the first set of experiments, we have undertaken a sensitive analysis of performance for the proposed algorithm by varying different parameters. We have chosen a full factorial design in which all possible combinations of the following factors are tested:

- Population size $PopSize$: 2 levels (100, and 150);
- Crossover type : 4 levels (*2X*, *SJOX*, *1HX*, and *IntX*);
- Crossover probability $CrossProb$: 4 levels (0.5, 0.6, 0.7, and 0.8);
- Mutation probability $MutProb$: 3 levels (0.05, 0.01, and 0.15);
- Stopping criteria $MaxGen$: 3 levels (100, 200, and 400).

All the cited factors result in a total of $2 \times 4 \times 4 \times 3 \times 3 = 288$ different combinations. Every combination is tested on a set of problem instances using procedure proposed by [21] where $n \in \{20, 50, 80, ..., 470, 500\}$ and $m \in \{5, 10, 15, 20\}$. We generates 02 instances for each combination $n \times m$. 10 replicates of each instance is executed. The response variable of the experiment is the RPD.

The resulting experiment was analyzed by means of a multifactor analysis of variance (*ANOVA*) technique [22] with the least significant difference (*LSD*) intervals (at the 95% confidence level). We focus on the *F-ratio*, the greater this ratio is, the more effective the parameter will be. Fig 4 shows means plots of the calibration parameters ordered by their F-Ratio. By analyzing different levels in a means plot for each factor, one by one respecting their F-ratio order, we fix each factor at its best level as follow :

- $PopSize = 150$;
- $MaxGen = 400$;
- Crossover type : *1HX* ;
- $CrossProb = 0.8$;
- $MutProb = 0.15$.

C. Performance analysis of the proposed algorithm

The first set of experiments were conducted to evaluate the efficiency of the newly proposed predictive maintenance planning heuristic (Algorithm 1). For this reason, we evaluate both C_{max} and the total sum of maintenance operations earliness/tardiness for maintenance mode 2.

According to the machine accumulated degradation at the moment of predictive maintenance intervention, 3 cases may appear:

1) The maintenance operation is ideally scheduled if the machine is fully exploited, i.e its accumulated degradation is $= \Delta$;
2) Otherwise, it is either early if the accumulated machine degradation is $< \Delta$;

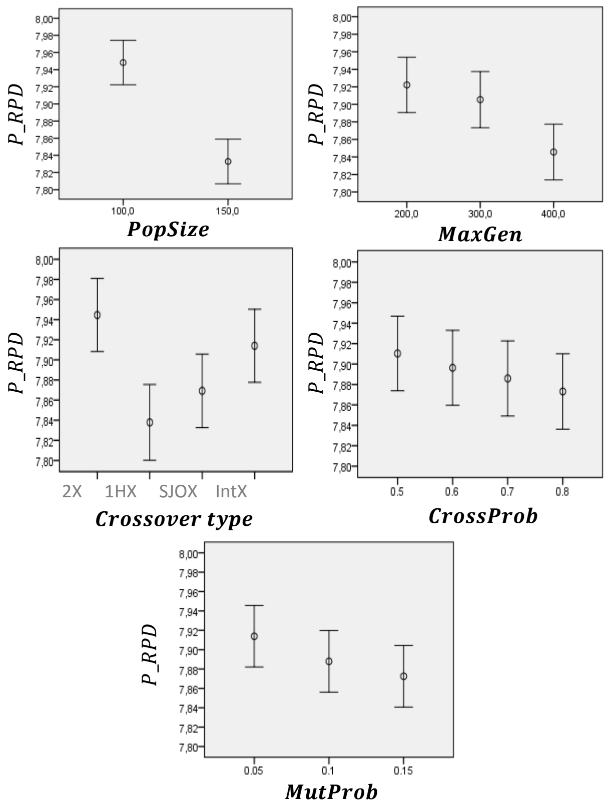

Fig. 4. Means plots for GA parameters calibration

3) Or tardy if the accumulated machine degradation is $> \Delta$.

Given the integrated scheduling of each machine $M_i : \pi_i = \{\mathcal{B}_{i1}, \mathcal{M}_{i1}, \mathcal{B}_{i2}, \ldots, \mathcal{M}_{i(l_i-1)}, \mathcal{B}_{il_i}\}$, the earliness/tardiness of intervention $\mathcal{M}_{ik}$ is calculated as follows:

$$ET(\mathcal{M}_{ik}) = |\Delta - \sum_{J_j \in \mathcal{B}_{i,k-1}} \delta_{ij}| \times 100 \qquad (3)$$

Then, the total sum of earliness/tardiness for all maintenance operations scheduled on all machines is expressed in Eq. 4

$$ET(\pi) = \sum_{i=1}^{m} \sum_{k=1}^{l_i} ET(\mathcal{M}_{ik}) \qquad (4)$$

Results reported in Table I present a comparison of ET and C_{max} of solutions obtained by our GA using the proposed maintenance insertion heuristic and the same GA using an insertion method that systematically schedules interventions before the maximal threshold Δ. We remark that, for several cases, results of ET of the proposed heuristic are better than those obtained by the systematic one. In the other cases, they are equivalent. However, when observing C_{max} results, we can easily note that the proposed maintenance planning heuristic provides the best solutions for all cases. In average, C_{max} deviation is decreased from 7.16% to 4.26%. Hence, we can deduce that the proposed heuristic is efficient since it allows to find good quality solutions regarding C_{max} minimization while scheduling interventions with small earliness/tardiness.

TABLE I
PREDICTIVE MAINTENANCE INSERTION HEURISTIC EVALUATION

size	Proposed heuristic		Systematic heuristic	
	RPD	ET	RPD	ET
20×5	3.63	3.45	5.15	3.75
20×10	3.51	3.50	5.08	3.47
20×20	2.64	3.83	4.82	3.62
50×5	3.33	3.26	5.81	3.05
50×10	5.27	2.74	7.60	2.52
50×20	5.61	3.65	7.89	3.77
100×5	3.92	3.39	7.23	3.51
100×10	4.71	3.06	8.04	2.82
100×20	6.10	3.61	9.15	3.87
200×10	5.24	3.55	8.37	3.43
200×20	6.16	3.98	9.58	3.72
Average	4.26	3.46	7.16	3.41

TABLE II
RPD RESULTS FOR MAINTENANCE MODE 1,2 AND 3

size	Mode 1		Mode 2		Mode 3	
	GA	I_NEH	GA	I_NEH	GA	I_NEH
20×5	0.82	3.94	3.63	7.93	8.72	15.12
20×10	0.94	4.83	3.51	7.85	9.02	16.16
20×20	1.12	4.11	2.64	6.70	6.54	13.01
50×5	0.53	1.27	3.33	6.13	8.36	12.21
50×10	1.34	5.03	5.27	9.67	9.71	17.52
50×20	1.76	5.14	5.61	10.58	10.05	20.62
100×5	0.88	1.43	3.92	6.88	9.37	13.76
100×10	1.02	2.86	4.71	7.82	10.12	19.02
100×20	1.45	5.14	6.10	10.71	11.82	23.11
200×10	0.92	2.08	5.24	8.10	12.04	17.87
200×20	1.06	3.79	6.16	9.65	13.47	19.21
Average	1.08	3.60	4.56	8.37	9.93	17.06

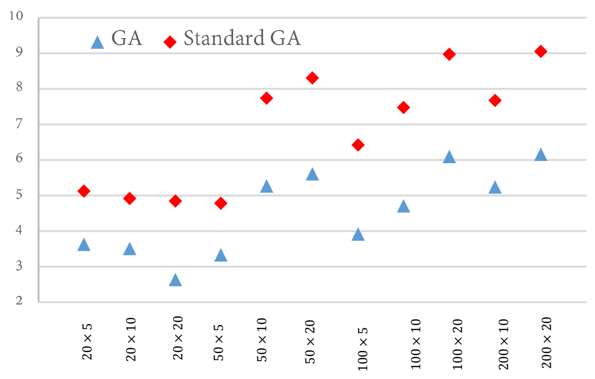

Fig. 5. Comparison between standard GA and our GA

The proposed genetic algorithm incorporates a restart scheme designed to enhance the diversification of the search process in case of stagnation. To study the influence of this mechanism, we show in Fig. 5 a comparison between the RPD obtained by our GA and a standard GA that uses the same operators without restart scheme for maintenance mode 2. We note that the standard GA parameters were also calibrated and the best combination was fixed. We remark that results obtained by our GA are better for all benchmarks. Deviations obtained by the standard GA are decreased by our GA by about 50%. Indeed, the standard GA converges quickly to a local optimum and then the search process wont be able to find new solutions of improved quality. On the other hand, our genetic algorithm is able to resume its search process and the best solution is once again improved after stagnation by exploring new promoting zones.

The third set of experiments reported in Table II were conducted to evaluate the proposed GA for C_{max} optimization when compared to results obtained by the *I_NEH* heuristic (NEH solution with predictive maintenance insertion as described in section III-B). Even if the NEH heuristic is the most powerful construction heuristic for C_{max} minimization [10], it fails to reach results obtained by our GA. That confirms the designed GA efficiency to find high quality solutions for all problem instances. This is argued by the design of appropriate GA operators, especially the crossover one, and the good parameters setting based on the sensitive calibration step. When comparing the effect of different maintenance modes on C_{max} deviations (average results for maintenance mode 1, 2 and 3), we note that the longest is the maintenance operations, the higher is the C_{max} deviation. This can be explicated by the insertion of predictive maintenance operations of more important processing time which yields higher deviations. An important note to report is that, high C_{max} deviations do not reflect bad solution quality since they are calculated relatively to Taillard best known solutions without maintenance operations. It is natural that the insertion of maintenance operations will certainly yield an increase of C_{max}.

The average computational times (CPU) grouped for each size of the problem are depicted in Table III. CPU times vary by the size of the problem. As the problem size increases, the CPU time becomes more important. Small instances, where $n \times m < 100 \times 20$, are resolved in less than a minute. When the problem size increases, the proposed GA becomes slower. This can be explain by the computational time spent in manipulating the solutions structures (job sequences and maintenance matrix) by different genetic operators.

V. CONCLUSION

We study the permutation flowshop scheduling problem with predictive maintenance activities under makespan minimization criterion. In our model, a PHM system is supposed to provide RULs and the degradation values for each machine when processing each kind of job. We proposed a new maintenance planning heuristic to guarantee an efficient exploitation of machines. Furthermore, we developed a tailored genetic algorithm including carefully tuned operators and parameters. The proposed algorithm incorporates a restart mechanism that allow diversification of the manipulated population when the

TABLE III
COMPUTATION TIMES (CPU)

size	CPU
20×5	02.13 s
20×10	04.96 s
20×20	11.81 s
50×5	06.24 s
50×10	15.23 s
50×20	40.02 s
100×5	10.14 s
100×10	0.52 min
100×20	2.03 min
200×10	1.37 min
200×20	6.04 min

search process is stagnated. We have carried out various experiments on well known Taillard benchmarks for PFSP. Results show the efficiency of the proposed genetic algorithm and maintenance insertion method.

Further topics would be continued with regards to this results. The proposed integrated scheduling model can be extended to manage other production systems typologies like jobshop. Another work can deal with the bi-objective character of the tackled problem in order to minimize both makespan and maintenance earliness/tardiness.

REFERENCES

[1] V. Venkatasubramanian, "Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities," *Computers & chemical engineering*, vol. 29, no. 6, pp. 1253–1263, 2005.

[2] T. Brotherton, G. Jahns, J. Jacobs, and D. Wroblewski, "Prognosis of faults in gas turbine engines," in *Aerospace Conference Proceedings, IEEE*, vol. 6, 2000, pp. 163–171.

[3] ISO, "Condition monitoring and diagnostics of machines, prognostics part 1: General guidelines," International Organization for Standardization, Tech. Rep. ISO13381-1, 2004.

[4] E. Pan, W. Liao, and L. Xi, "A joint model of production scheduling and predictive maintenance for minimizing job tardiness," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 9, pp. 1049–1061, 2012.

[5] C. Varnier and N. Zerhouni, "Scheduling predictive maintenance in flow-shop," in *IEEE Conference on Prognostics and System Health Management (PHM)*, Beijing, China, 23–25 May 2012 2012, pp. 1–6.

[6] N. Herr, J. M. Nicod, and C. Varnier, "Prognostics-based scheduling in a distributed platform: Model, complexity and resolution," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei, Taiwan, 18–22 Aug 2014 2014, pp. 1054–1059.

[7] S. Chrétien, N. Herr, J. M. Nicod, and C. Varnier, "A post-prognostics decision approach to optimize the commitment of fuel cell systems in stationary applications," in *IEEE Conference on Prognostics and Health Management (PHM)*, TX, USA, 22–25 Jun 2015 2015, pp. 1–7.

[8] A. Ladj, C. Varnier, F. Benbouzid-Si Tayeb, and N. Zerhouni, "Exact and heuristic algorithms for post prognostic decision in a single multifunctional machine," *International Journal of Prognostics and Health Management*, vol. 8, no. 2, 2017.

[9] Q. Liu, M. Dong, and F. Chen, "Single-machine-based joint optimization of predictive maintenance planning and production scheduling," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 238–247, 2018.

[10] V. Fernandez-Viagas, R. Ruiz, and J. M. Framinan, "A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation," *European Journal of Operational Research*, vol. 257, no. 3, pp. 707–721, 2017.

[11] A. Ladj, F. Benbouzid-Si Tayeb, C. Varnier, A. A. Dridi, and N. Selmane, "A hybrid of variable neighbor search and fuzzy logic for the permutation flowshop scheduling problem with predictive maintenance," *Procedia Computer Science*, vol. 112, pp. 663–672, 2017.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] W. Kubiak, J. Błażewicz, P. Formanowicz, J. Breit, and G. Schmidt, "Two-machine flow shops with limited machine availability," *European Journal of Operational Research*, vol. 136, no. 3, pp. 528–540, 2002.

[14] R. Ruiz, C. Maroto, and J. Alcaraz, "Two new robust genetic algorithms for the flowshop scheduling problem," *Omega*, vol. 34, no. 5, pp. 461–476, 2006.

[15] R. Ruiz, J. C. García-Díaz, and C. Maroto, "Considering scheduling and preventive maintenance in the flowshop sequencing problem," *Computers & Operations Research*, vol. 34, no. 11, pp. 3314–3330, 2007.

[16] F. Benbouzid-Sitayeb, S. A. Guebli, Y. Bessadi, C. Varnier, and N. Zerhouni, "Joint scheduling of jobs and preventive maintenance operations in the flowshop sequencing problem: a resolution with sequential and integrated strategies," *International Journal of Manufacturing Research*, vol. 6, no. 1, pp. 30–48, 2011.

[17] B. Naderi, M. Zandieh, and M. Aminnayeri, "Incorporating periodic preventive maintenance into flexible flowshop scheduling problems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2094–2101, 2011.

[18] F. Benbouzid-Si Tayeb and W. Belkaaloul, "Towards an artificial immune system for scheduling jobs and preventive maintenance operations in flowshop problems," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*. IEEE, 2014, pp. 1065–1070.

[19] M. Nawaz, E. E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.

[20] Z. Michalewicz and S. J. Hartley, "Genetic algorithms+ data structures= evolution programs," *Mathematical Intelligencer*, vol. 18, no. 3, p. 71, 1996.

[21] E. Taillard, "Benchmarks for basic scheduling problems," *european journal of operational research*, vol. 64, no. 2, pp. 278–285, 1993.

[22] D. C. Montgomery, *Design and analysis of experiments*. John Wiley & Sons, 2008.