

Exact and heuristic algorithms for post prognostic decision in a single multifunctional machine

Asma Ladj¹, Christophe Varnier², Fatima Benbouzid Si Tayeb³, and Nouredine Zerhouni⁴

^{1,3} *Laboratoire des Méthodes de Conception de Systèmes (LMCS)
Ecole nationale Supérieure d'Informatique (ESI)
BP 68M-16270 Oued Smar, Alger-Algérie*

*a_ladj@esi.dz
f_sitayeb@esi.dz*

^{2,4} *Institut FEMTO-ST UMR 6174, ENS2M, CNRS, Univ. Bourgogne Franche-Comté, F-25000 Besançon-France
christophe.varnier@ens2m.fr
zerhouni@ens2m.fr*

ABSTRACT

Prognostic and Health Management (PHM) benefits are strongly tied to the decision-making that follows the assimilation and interpretation of prognostics information. Hence, we deal in this study with the post prognostic decision making in order to improve system safety and avoid downtime and inopportune maintenance spending. We investigate the problem of scheduling production jobs in a single multifunctional machine subjected to predictive maintenance based on PHM results. For this reason, we propose a new interpretation of PHM outputs to define the machine degradation corresponding to each job. We develop a *Mixed Integer Linear Programming (MILP)* model to find the best integrated scheduling that optimizes the total maintenance cost. Unfortunately, the MILP is not able to compute the optimal solution for large instances. Therefore, we design a *Prognostic based Genetic Algorithm (Pro-GA)*. Computational results of different benchmarks setup show the efficiency and robustness of our scheme with an average deviation of about 0.2% over a newly proposed lower bound.

1. INTRODUCTION

In highly competitive environment, industrials are seeking to gain advantage with respect to cost, quality, and time. However, no matter how sophisticated manufacturing systems are, physical assets deteriorate over time due to functioning stress and load. For this reason, attention to the maintenance activity has rapidly increased as an inevitable reality in indus-

try. Maintenance becomes a major contributor to the improvement of the manufacturing system reliability and performance. In fact, maintenance strategies have extremely evolved over time and can be classified into two main policies: corrective and preventive maintenance (Duffuaa, Ben-Daya, Al-Sultan, & Andijani, 2001; CEN/EN, 2010). Traditionally, systems were repaired after failures, this is known as *corrective maintenance*. Each unexpected failure induces tremendous financial losses which consist of downtime cost (production loss, unavailability cost etc...) and repair cost (Berdinyazov, Camci, Sevkli, & Baskan, 2009). To overcome this problem and reduce failure risk for critical systems, concept of maintenance before failure has emerged to propose a new maintenance type called *preventive maintenance*. First, *systematic preventive maintenance* (also known as *time-based preventive maintenance*) was applied according to a periodic schedule where intervals were known and fixed in advance regardless of the system health state. The determination of maintenance interval is critical. However, two drifts can be observed with this latter type. The first occurs when the maintenance frequency is very high (small intervals), which induces an excessive cost due to useless interventions. The second occurs when the time between two successive maintenance interventions is very long, consequently failures cannot be avoided resulting in system shutdown. The solution is to observe the system health state in real time and recommend maintenance decisions based on the information collected through condition monitoring, this is called *Condition Based Maintenance (CBM)* (Lebold & Thurston, 2001).

To reduce maintenance costs noticeably while increasing equipments reliability and availability, a more efficient maintenance strategy must combine sophisticated methods, tools

Asma Ladj et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and techniques. Industrials show a growing interest in Prognostics and Health Management (PHM) which becomes a major research framework for the scientific community (Venkatasubramanian, 2005). It is nowadays recognized as a key feature applied to enhance a cost effective maintenance strategy and higher quality design (Brotherton, Jahns, Jacobs, & Wroblewski, 2000). Thereby, a new maintenance type is emerged, called predictive maintenance where representative features are obtained by condition monitoring and then forecasted in order to predict the evolution of degradation phenomena. Indeed, rather than understanding a failure which has just appeared to call for very expensive corrective interventions (diagnosis process), it seems more convenient to "anticipate" its occurrence (prognostic process) in order to resort to protective actions at the most appropriate time. Several definitions have been given in the literature about industrial prognostic (Lebold & Thurston, 2001; Byington, Roemer, & Galie, 2002; Muller, Suhner, & Iung, 2008). These definitions are then normalized by the International Organization for Standardization in which prognostic is defined as "*the estimation of time to failure and risk for one or more existing and future failure modes*" (ISO, 2004). The time to failure is commonly called *Remaining Useful Life (RUL)*. Developing sophisticated PHM tools represents a promising research area. On the other hand, PHM benefits are also strongly tied to the decision-making that follows the assimilation and interpretation of prognostics information, which refers to *Post Prognostic Decision* (Iyer, Goebel, & Bonissone, 2006). In fact, post prognostic decision seeks to improve safety, allow avoiding downtime and inopportune maintenance spending, plan successful missions, schedule maintenance, etc.

Besides maintenance optimization, mission scheduling, system control, post prognostic decision may appear as the integration of predictive maintenance planning in production schedule. This is a well known problem in the literature called "*production scheduling with availability constraints*" or "*integrated maintenance and production scheduling*" (Hadidi, Al-Turki, & Rahim, 2011). Works were previously proposed to investigate these problems. Two cases of consideration about the unavailability constraints can be found in the literature: (i) the deterministic case where intervals are known and fixed in advance and often correspond to systematic preventive maintenance operations (Ma, Chu, & Zuo, 2010), (ii) the dynamic case where unavailabilities periods are flexible and stand as decision variables. This is the case for instance when information about predictive maintenance are provided from prognostic.

In the latter case, we focus on making decisions integrating both production jobs and predictive maintenance operations. Relatively few works were proposed in this context. (Pan, Liao, & Xi, 2012) proposed a prognostics-based scheduling model incorporating production jobs and predictive maintenance operations for a single machine with the objective of

minimizing the maximum tardiness using mathematical programming formulation. Authors assumed that machine condition could be monitored and the machine RUL could be estimated. As the machine breakdown will result in process interruption and huge losses, a safety threshold is set before reaching the RUL. Hence, predictive maintenance operations are performed based on a new metric called the *Remaining Maintenance Life (RML)*. For a numerical example of nine jobs, the proposed scheduling model proves its efficiency on reducing tardiness as well as keeping machine in good operation condition when compared to three previous models: production scheduling model without maintenance planning, production scheduling model with periodic maintenance planning, and individual production scheduling and predictive maintenance planning. (Varnier & Zerhouni, 2012) proposed an original approach for solving flowshop scheduling problem with predictive maintenance where machines are able to switch between two production modes: nominal and sub-nominal. In the second mode, machine is slowed down to avoid early failures. As a consequence, the production tasks will be longer than expected but in counter part remaining useful life is increasing. They developed a mixed integer linear model that allows finding the best production and predictive maintenance scheduling optimizing the aggregated sum of makespan and maintenance delays. Obtained results show that for several cases the best solution is reached when some machines are switched in degraded mode. (Herr, Nicod, & Varnier, 2014) studied an interesting case of parallel machines. The platform can be running using different operating conditions corresponding to different production throughputs. Moreover, authors assumed that each machine is monitored and associated with a prognostics module that gives a RUL value depending on both its past and future usage. Three heuristics based on well known dispatching rules, were proposed to select the appropriate profile for each machine during the whole production horizon. A prognostic-based scheduling with the objective of maximizing the production horizon which is the period between two maintenance interventions is provided. A second objective is to minimize maintenance costs. Considering a multi-stack fuel cell system, the same problem was studied by (Chrétien, Herr, Nicod, & Varnier, 2015), where prognostics results in the form of RULs were used to maximize the global useful life of the system under service constraint. Convex optimization was used to cope with the scale of the whole production horizon. The scheduling provided defines the contribution of each stack to a global needed power output so as to reach the power demand as long as possible.

In previous studies, a unique RUL value (expressed in unit of time) was estimated and used as a threshold to perform predictive maintenance operations regardless of tasks being processed and without taking into account the variable operating conditions of machines. However, due to technical progress,

various kinds of powerful single machine have been designed in the field of factory production (e.g., intelligent machine tool) that regroups several operations in a single cell (e.g., turning, cutting, milling, drilling, sawing, lathing and grinding). Each operation requires some means of constraining the workpiece and provide a guided movement between the workpiece and the toolpath. The speeds and feeds used vary with respect to the desired mission. Thus, the wear and tear of the machine depends on the kind of operation being executed because different stresses and movements induce various degradation levels. Hence, we propose in this study a new interpretation of PHM results. We assume that a single multifunctional machine is subjected to many predictive maintenance interventions during the planning horizon. This equipment is supposed to be monitored continuously and a PHM module provides, due to various deterioration levels, the corresponding RUL for each kind of job. Moreover, in this paper, we introduced a novelty in our experiments and introduce a new metric to express the degradation of the machine when processing each kind of job. First, we develop a *Mixed Integer Linear Programming (MILP)* model to find the best integrated schedule of production and predictive maintenance that optimizes the total maintenance cost. However, due to the number of variables and constraints in the MILP, only small instances of the problem can be solved. Therefore, we develop in the other hand a *Prognostic based Genetic Algorithm* called *Pro-GA* to solve larger instances of the studied problem. Moreover, we propose a new lower bound to evaluate our algorithm performance for different setup cases.

The remaining content of the paper is organized as follows. In section 2, the tackled integrated scheduling problem is detailed. Optimal resolution approach which consists of a Mixed Integer Linear Programming (MILP) model is provided in section 3. After that, our algorithm *Pro-GA* is developed in section 4 as well as the integrated classical and newly proposed genetic operators. Finally, experiments results are discussed. A general conclusion of the work and the perspectives considered are given in the last section.

2. PROBLEM STATEMENT

We consider here, a single multifunctional machine scheduling problem subject to unavailability constraints due to predictive maintenance interventions. We assume that the machine is monitored and a prognostic module is able to provide significant information used to make post prognostic decision. Thus, the resulting prognostic based integrated scheduling incorporates both production jobs and predictive maintenance activities with the objective of minimizing the total maintenance cost. In this section, we describe first the integrated scheduling problem we are dealing with. We define next both PHM and predictive maintenance problems. The objective function of our model is also presented. An example of application case is finally given.

2.1. The integrated scheduling problem

The problem we face is known as the single machine scheduling problem under availability constraints. Considering a multifunctional machine that should process a job set $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ and subjected to predictive maintenance operations, the problem tackled here consists in determining what is the best sequencing of production jobs and the best emplacement of maintenance interventions that minimizes the total maintenance cost. There are several assumptions that are commonly made regarding this problem (Merten & Muller, 1972):

- All jobs $J_i \forall i \in \{1, \dots, n\}$ are available at time zero ;
- Each job J_i requires a given known, deterministic and non negative processing time, denoted $p_i \forall i \in \{1, \dots, n\}$;
- The machine is not continuously available due to predictive maintenance operations. When available, it can process only one job at a time.

Hence, the planning horizon can be divided into multiple production cycles separated by predictive maintenance interventions. First, machine *RULs* are predicted by the PHM module and a degradation value is associated to each kind of production job. Next, under the total maintenance cost minimization objective, the first block of jobs to be performed is generated respecting the predetermined constraint of machine maximal degradation. At the end of the first block, a predictive maintenance intervention is scheduled to recover the machine to its initial health state. The cost of this intervention is calculated based on the accumulated degradation of jobs assigned to the first block. After that, given the rest of jobs, a new block will be built and launched and the new maintenance operation cost is added to the total one. The same procedure will be iterated till all jobs are scheduled. The problem consists then in determining the jobs assignment for each production block to minimize the total cost required to process all maintenance interventions.

The resulting integrated scheduling can be seen as a succession of several production blocks separated by predictive maintenance operation, denoted by $\pi = \{\mathcal{B}_1, \mathcal{M}_1, \dots, \mathcal{M}_{l-1}, \mathcal{B}_l\}$, where :

- \mathcal{B}_i is the i^{th} production block of jobs;
- \mathcal{M}_i is the i^{th} predictive maintenance activity;
- l is the number of blocks required to process all jobs : $\bigcup_{i=1}^l \mathcal{B}_i = \mathcal{J}$;
- Each job J_i is included strictly in one production block.

Figure 1 shows an example of an integrated scheduling $\pi = \{\mathcal{B}_1, \mathcal{M}_1, \mathcal{B}_2, \mathcal{M}_2, \mathcal{B}_3\}$ for a set of $n = 10$ jobs with 3 production blocks and 2 predictive maintenance operations. This

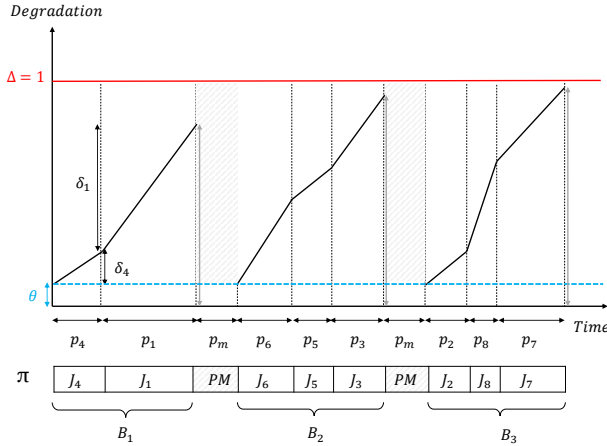


Figure 1. Integrated scheduling example

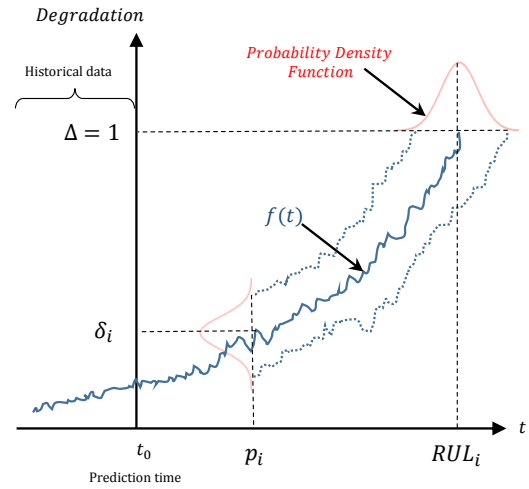
figure shows also the cumulative degradation level of machine during the horizon of the schedule.

To detail the integrated scheduling problem of production and predictive maintenance we are dealing with, we define in subsections below PHM context and predictive maintenance model.

2.1.1. The Prognostic Health Management problem

Reliability estimations is mandatory to substitute traditional maintenance concepts with new ones and prevent inopportune spending. In our work, we assume that prognostic tools and models are available from other studies (Liu, Zhang, Li, Lu, & Hu, 2014; Tobon-Mejia, Medjaher, & Zerhouni, 2012). In this context, the machine is supposed to be monitored continuously. Given the current machine health state, the operating environment and the observed condition monitoring, the associated PHM module is able to predict the deterioration evolution and estimate the Remaining Useful Life (RUL). Moreover, as diverse tasks are being processed on the multifunctional machine, the latter is subjected to a deterioration process that depends on the job being processed. Indeed, every kind of job requires specific functionalities that cause various levels of damage on the equipment. Hence, under these conditions, each job J_i has an associated remaining useful life value RUL_i as well as a degradation value δ_i . We consider the following assumptions:

- A deteriorating prognosis system provides RUL_i of the machine corresponding to a given Job $J_i \forall i \in \{1, \dots, n\}$;
- RUL_i represents the period during which the “as good as new” machine could achieve job J_i before failure;
- The PHM module is also able to provide an associated degradation value δ_i for each job J_i ;
- $\delta_i \in]0;1[$ represents the wear and tear of the machine


 Figure 2. Relationship between p_i , RUL_i , and δ_i

when only job J_i is processed during the processing time p_i (0 means no degradation committed, 1 a full degradation);

- $\delta_i = f(p_i)$ where f characterizes the evolution of the machine degradation (Figure 2).

2.1.2. The predictive maintenance scheduling problem

Corrective maintenance is a curative reaction of repairing the equipment after the failure. On the other hand, systematic preventive maintenance aims to avoid failure by planning periodic interventions. To reduce the risk of machine failures while avoiding inopportune spending, predictive maintenance is proposed to focus on early detection and forecasting of failures. This strategy uses PHM outputs, in our case degradation value corresponding to each job, to schedule predictive maintenance interventions. After predictive maintenance strategy implementation, a predictive maintenance operation means intervention performed on the machine to recover it to its initial health state. For our model, we consider the following assumptions:

- Let Δ be the maximal authorized degradation of the machine. Beyond this threshold, a predictive maintenance task should be planned. In this study, we fix $\Delta = 1$. Hence, each degradation value $\delta_i < \Delta \forall i \in \{1, \dots, n\}$.
- The accumulated degradation of the machine between two consecutive maintenance operations should never exceed this threshold Δ .
- At the beginning of the planning horizon, the machine has a degradation equals to θ . After a predictive maintenance operation, the machine is recovered to its initial health state, i.e. its new accumulated degradation is reset to θ .
- During the planning horizon, at least one predictive maintenance operation is performed. Considering that, the machine

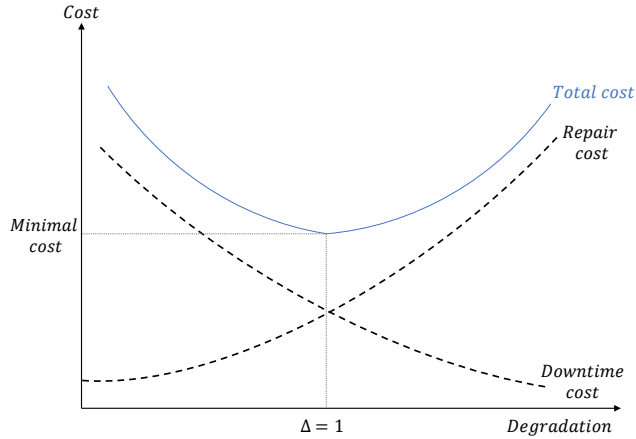


Figure 3. Predictive maintenance cost model

could not process all the jobs before maintenance operation.

- No predictive maintenance operation is performed after the processing of last jobs.
- If an accidental failure occurs during the production horizon, then the rest of jobs are rescheduled based on new data.

In this paper, we propose a new model to evaluate the cost of a maintenance operation. It is assumed that the cost of PHM is included. This cost is divided into two (2) parts: a repair cost and a downtime cost due to maintenance intervention as shown in Figure 3. When a maintenance action is performed due to a small degradation level, low repair cost and high downtime cost will be faced. As the machine degradation actions increase, the downtime cost is reduced and the repair cost is increased. It is natural that the repair cost is proportional to machine deterioration: the higher the machine damage is, the higher the cost is. The downtime cost can be explained by opportunity cost. Early repairing a workable machine when a small degradation level is observed, will lead to inopportune spending due to unnecessary intervention. Indeed, this intervention interrupts the whole production process while it is not necessary. There is an optimum point that minimizes the total maintenance cost. The minimum cost is spent when the machine reaches the maximum threshold of degradation Δ which means that the machine has been used at its full potential and all maintenance intervention have been scheduled at the right time. For simplicity reasons, it is assumed that additional costs (e.g., instrumentation cost, software cost, condition monitoring spending, ...) are constant and included in the repair cost.

2.2. An example of application framework

The study carried out here is generalized enough to deal with several application cases where the running machine is able to perform several tasks. To explain the application of our

model in industrial firms, we take the example of machine tool which plays a very important role in modern manufacturing systems. It is used for shaping or machining metal or other rigid materials. It is a multifunctional machine able to process various kind of operations: cutting, boring, grinding, shearing, or other forms of deformation. Hence, the wear and tear of the machine tool depends on its variable operating conditions (material type, hole shape and dept, machine feed and speed, ...). (Sardinas, Santana, & Brindis, 2006). Each desired shape requires some means of constraining the workpiece and provide a guided movement between the workpiece and the toolpath. There is a close relationship between machine tool parameters and its deterioration. Since no machining theory is available to predict the machine remaining useful life, variety of parameters can be detected and used to predict its RUL: temperature, current, acoustic emission and vibration. These conditions data are collected in order to predict the future trend of deterioration: this is known as the data-driven prognostic approach (Liu et al., 2014).

An effective post prognostic decision making is then required to avoid production loss and make full use of the machine. Early replacement of a workable machine or late replacement of a worn one may cause time and/or production loss (Aliustaoglu, Ertunc, & Ocak, 2009). Thus, it is important to determine the best time of when predictive maintenance interventions should be performed. An integrated schedule must be established to jointly plan production jobs processed by the machine tool and predictive maintenance intervention to improve its availability.

3. THE PROPOSED MILP FOR EXACT RESOLUTION

In this section, we propose an exact resolution to cope with small instances of the tackled problem. We model the integrated scheduling of production and predictive maintenance problem that optimize the total maintenance cost using a *Mixed Integer Linear Program*.

3.1. Notations

In the following, we will use the notation defined here:

- J_i : job number i ;
- n : number of jobs to be scheduled;
- p_i : processing time of job J_i ;
- RUL_i : RUL of machine when processing job J_i ,
- δ_i : machine degradation corresponding to job J_i ;
- Δ : maximum threshold of machine degradation;
- C_0 : minimum predictive maintenance cost spent when the machine reaches a full degradation Δ (Figure 4);
- C_f : maximum predictive maintenance cost (Figure 4);
- B_j : production block number j .

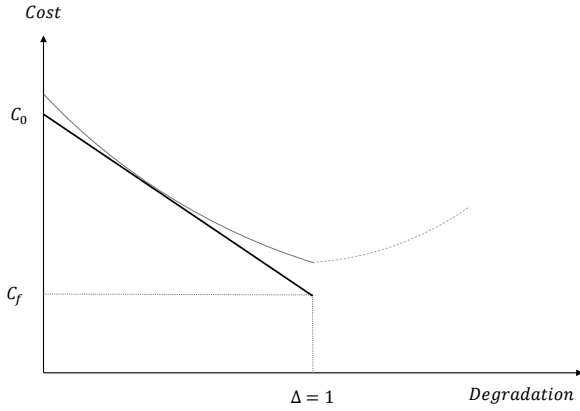


Figure 4. Predictive maintenance cost

To seek simplicity, we consider here for that each job J_i , the corresponding degradation δ_i is calculated as expressed in Eq. (1):

$$\delta_i = \frac{p_i}{RUL_i} \quad (1)$$

Moreover, the predictive maintenance cost discussed in Section 2.1.2 is supposed to be a linear function as shown in Figure 4.

3.2. Mixed Integer Linear Programming model

We propose to model the problem with mixed integer linear programming.

Variables:

- x_{ij} : binary variable. $x_{ij} = 1$ if job J_i is assigned to production batch B_j , 0 otherwise;
- y_j : binary variable. $y_j = 1$ if batch $B_j \neq \emptyset$, 0 otherwise;
- Deg_j : machine degradation accumulated after processing batch B_j . It is calculated as follow:

$$Deg_j = \theta + \sum_{i=1}^n \delta_i \cdot x_{ij} \quad (2)$$

From Eq. (1) :

$$Deg_j = \theta + \sum_{i=1}^n \frac{p_i}{RUL_i} \cdot x_{ij} \quad (3)$$

- $Cost_j$: cost of the predictive maintenance operation performed after production batch B_j . From Figure (4) this variable is given by the following equation:

$$Cost_j = C_0 + (C_f - C_0)Deg_j \quad (4)$$

From Eq. (2) we can write the final equation :

$$Cost_j = C_0 + (C_f - C_0)(\theta + \sum_{i=1}^n \delta_i \cdot x_{ij}) \quad (5)$$

Constraints:

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, n\} \quad (6)$$

To process all jobs, we need to program at maximum n production blocks to process all jobs. Thus, a upper bound of the production blocks number is n .

$$y_j \in \{0, 1\} \quad \forall j \in \{1, \dots, n\} \quad (7)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \quad (8)$$

$$\sum_{i=1}^n \delta_i \cdot x_{ij} \leq (\Delta - \theta) \cdot y_j \quad \forall j \in \{1, \dots, n\} \quad (9)$$

Eq. (8) means that all jobs have to be produced by the machine exactly once. In other words, a job must be included in exactly one production block. While Eq.(9) ensures the maximum degradation constraints, i.e. the accumulated degradation of the machine after processing each production block does not exceed the maximum threshold Δ .

Objective function:

The resulted integrated scheduling must optimize the total cost required to process all predictive maintenance operations:

$$Cost = \sum_{j=1}^n Cost_j \cdot y_j \quad (10)$$

Our objective is then to minimize the total cost :

$$\min(\sum_{j=1}^n Cost_j \cdot y_j).$$

From Eq. (5) we can write :

$$\min(\sum_{j=1}^n (C_0 + (C_f - C_0)(\theta + \sum_{i=1}^n \delta_i \cdot x_{ij})) \cdot y_j)$$

One can note that this is not a linear function. We can transform this equation into a linear one as follow:

$$\min \sum_{j=1}^n Cost_j \cdot y_j = \min \sum_{j=1}^n Cost_j$$

Eq. (4) and Eq. (9) and $C_f \ll C_0 \Rightarrow (\min Cost_j \Leftrightarrow \max Deg_j)$

$$\begin{aligned}
maxDeg_j &\Leftrightarrow max \sum_{i=1}^n \delta_i \cdot x_{ij} \\
&\Rightarrow (min \sum_{j=1}^n Cost_j = max \sum_{j=1}^n \sum_{i=1}^n \delta_i \cdot x_{ij}) \\
&\Rightarrow (min \sum_{j=1}^n Cost_j \Leftrightarrow min(-\sum_{i=1}^n \sum_{j=1}^n \delta_i \cdot x_{ij})) \\
&\Rightarrow (min \sum_{j=1}^n Cost_j \Leftrightarrow min(\sum_{i=1}^n \Delta \cdot y_j - \sum_{i=1}^n \sum_{j=1}^n \delta_i \cdot x_{ij}))
\end{aligned}$$

Our new linear objective function is then :

$$min(\sum_{i=1}^n \Delta \cdot y_j - \sum_{i=1}^n \sum_{j=1}^n \delta_i \cdot x_{ij}) \quad (11)$$

In other word, if our goal is to minimize the total maintenance cost, we have to build as full as possible production blocks, so we have to minimize the gap between the accumulated degradation after processing each production block and the maximum authorized threshold Δ .

4. THE PROPOSED PROGNOSTIC BASED GENETIC ALGORITHM FOR APPROXIMATE RESOLUTION

As introduced above, the problem studied here is to create a prognostic based integrated scheduling of several jobs being processed by a single multifunctional machine with the objective of minimizing the total maintenance cost. Unfortunately, the MILP defined in the previous section is not able to compute the optimal solution for instances with an important number of jobs. To deal with larger instances, we propose a sub-optimal approach based on population based metaheuristic. It consists on a Prognostic based Genetic Algorithm, called *Pro-GA*. This choice is mainly adopted because Genetic Algorithms (GAs) (Goldberg, 1989) were widely used to solve production scheduling problems (Ruiz, Maroto, & Alcaraz, 2006). They were next successfully applied to production scheduling with systematic preventive maintenance (Ruiz, García-Díaz, & Maroto, 2007; Benbouzid-Sitayeb, Guebli, Bessadi, Varnier, & Zerhouni, 2011). Instead of implementing periodic preventive maintenance, the main feature of our *Pro-GA* is the predictive maintenance consideration. It is worth pointing out that to the best authors knowledge, our *Pro-GA* is the first metaheuristic proposed in the literature to solve the integrated scheduling problem of production and predictive maintenance. All previous works consist of either total enumeration methods (MILP) or heuristics. *Pro-GA* uses information about machine health state provided from PHM module in order to make the most appropriate post prognostic decision. Indeed, the estimated RULs and degradation levels give information about the machine health state during the production process. Thus, *Pro-GA* is able to incorporate these prognostic outputs in order to establish the most suitable production and predictive maintenance

integrated scheduling under the total maintenance cost minimization criterion. This can help both avoiding production loss and improving system availability.

GAs have gained considerable attention regarding their potential as an optimization technique for complex problems. Their main specific feature is their implicit parallelism, which is a result of the evolution and the hereditary-like process (Goldberg, 1989). In a classical GA, every individual is encoded into a structure. The set of individuals forms the population. The population undergoes a series of operations and evolves until some stopping criterion is met. At each generation, first a selection mechanism picks individuals of the current population. Then, the selected individuals mate and generate new offsprings: crossover process. Some offsprings might suffer a mutation (Goldberg, 1989). In our case, prognostic outputs are jointly taken into account with production data in all *Pro-GA* steps. This is guaranteed by considering a unique structure to represent individuals. Indeed, the "Pro" part is involved in initial population generation, individuals evaluation (objective function), and population improvement. For this reason, genetic operators (crossover, mutation) are adapted to deal with the integrated problem tackled here. The flowchart of our proposed *Pro-GA* is presented in Figure (5). The part framed in red represents the restart scheme we have designed and incorporated into classical GA implementation to stabilize the population convergence throughout the search. The most remarkable characteristics of our *Pro-GA* are:

- Use of an integrated representation of production, prognostic and maintenance data;
- Use of adapted heuristics rules to generate integrated individuals of the initial population;
- Use of a new proposed crossover to guarantee inheritance of good features from parents to offspring;
- Use of a restart scheme to provide a tactical balance between intensification and diversification of the research.

In the following subsections, we present first the integrated encoding scheme we propose. Next, we detailed the specification of genetic operators we designed. Finally, the proposed restart scheme is explained.

4.1. The encoding scheme and the fitness function

The representation step specifies the mapping from the individuals (candidate solutions) into a set of genotypes. In our GA, a genotype is expressed by sequencing the job sets for all the production blocks. A set of jobs numbers in one block corresponds to a gene. For example, for an instance of problem $\mathcal{J} = \{J_1, J_2, \dots, J_{10}\}$, a candidate solution is $\pi = \{(5, 10)(2, 9, 1, 3)(6, 7, 8)(4)\}$. We decode this representation by scheduling the jobs of the first block (J_5 and J_{10}), then performing the first predictive maintenance opera-

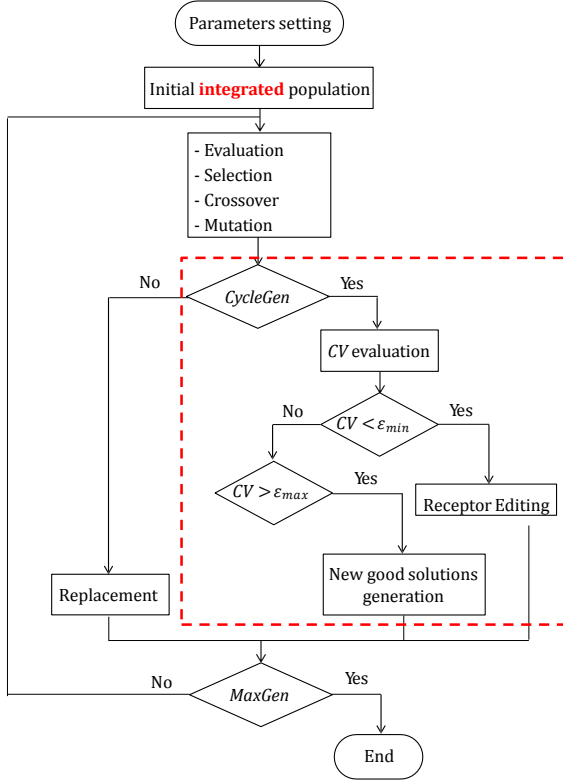


Figure 5. flowchart of the proposed Pro-GA

tion with a cost depending on assigned jobs. Next, we iterate the same process for the rest of blocks one by one.

GA needs a fitness function to evaluate the quality of an individual in the population. Giving a candidate solution $\pi = \{\mathcal{B}_1, \mathcal{M}_1, \dots, \mathcal{M}_{l-1}, \mathcal{B}_l\}$, the total predictive maintenance cost is equal to Eq. (12) (see Figure 4):

$$Cost(\pi) = \sum_{k=1}^{l-1} (C_0 + (C_f - C_0) \sum_{J_i \in \mathcal{B}_k} \delta_i) \quad (12)$$

The objective function is to minimize the total maintenance $Cost(\pi)$. Therefore, the definition of fitness function is just the reciprocal of this cost. The fitness of each chromosome π is calculated according to Eq. (13) as follows:

$$Affinity(\pi) = \frac{1}{Cost(\pi)} \quad (13)$$

It can be noticed that the lower the predictive maintenance cost is, the higher the affinity value is and so the better the solution is.

4.2. Population initialization

Instead of starting with an initial population randomly generated, it seems more efficient to use special techniques to produce a higher quality initial population (Reeves, 1995). We propose a two-step initialization procedure where an initial population of $PopSize$ individuals is generated as follows:

1. The first part of the initial population and the largest one ($\alpha\% \times PopSize$) is randomly generated. Its purpose is to ensure diversity of the research. First, a random permutation of all jobs is generated. Then First Fit heuristic (Coffman, Garey, & Johnson, 1984) is applied on this permutation in order to form a candidate solution by assigning jobs to production blocks (starting from empty production blocks, jobs are assigned one by one to the first available block).
2. The remaining part ($(100 - \alpha)\% \times PopSize$) is generated using the two common heuristics First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) heuristics (Coffman et al., 1984). It exploits the characteristics of these good solution to form other solutions by applying a series of permutations between jobs. Thus, we ensure that this part of the population is formed by fit members.

Since this initialization scheme uses a First Fit and Best Fit ordering, it naturally avoids generating invalid solutions. In other words, the maximal degradation constraints are respected and no feasibility check is needed.

4.3. The population improvement

Selection operator: for the sake of simplicity, we choose the 2-tournament classical selection scheme (Michalewicz & Hartley, 1996). It consists on randomly choosing two members from the current population and selecting the fittest one.

Crossover operator: since the considered objective is the total maintenance cost minimization, by analyzing the cost evolution model (Figure 6) we deduced that ideally, a preventive maintenance operation is planned when the accumulated degradation reaches the maximal threshold Δ . Thus, it is clear that we should make *full* use of the machine and then build as full as possible production blocks. To seek this goal, we propose a new crossover operator based on the one proposed by (Rohlfshagen & Bullinaria, 2010). This crossover produced a single offspring by copying the fullest bins from parents.

With a probability equals to $CrossProb$, our newly proposed crossover operator produces two offspring from two selected parents as follows:

1. In the first phase, blocks from both parents are sorted in the order of non-increasing degradation;

2. Next, starting from two empty offspring, we copy the fullest non-overlapping blocks from parents. In other words, a block is copied in both offspring only if it contains no duplicated job.
3. Finally, we need to represent the parents as job sequence (permutation of jobs). We scan the sequence of the first parent, respectively the second, from left to right, skipping jobs that are already contained and assign the rest of the jobs to the first offspring, respectively the second, using the First Fit rule.

Our crossover naturally avoids generating unfeasible solutions, where the accumulated degradation in a block exceeds the maximum capacity Δ . This can be explained by the use of the First Fit heuristic. This eliminates the time that would be spent cutting down unfeasibilities. Moreover, our crossover operator preserves good characteristics of parents (the fullest production blocks) and transfers them to offspring. Figure (6) shows an example of the crossover operator.

Mutation operator: we chose a simple mutation method inspired from the classical SWAP mutation (Michalewicz & Hartley, 1996). It consists on swapping, if possible, two randomly selected jobs from two different blocks. We only allow mutations that guarantee the feasibility of the obtained solutions. Thus, the maximal threshold Δ must be respected for each block. The mutation probability is set to $MutProb$. For each mutation operation, the number of permutations is randomly chosen between $(5\%n + 1)$ and $(15\%n + 1)$ where n is the number jobs.

4.4. Replacement

Individuals of the next generation are selected from the whole population formed by parents and newly created children. $\beta\%$ of the worst individuals are directly inserted in the new population. Then, we complete the rest of the population by the fittest members from parents and children.

4.5. Restart scheme

Intensification (exploitation) and diversification (exploration) are two major issues introduced to build effective search algorithms (Goldberg, 1989). Diversification generally refers to the ability to visit many and different regions of the search space, whereas intensification refers to the ability to obtain high quality solutions within those regions. In this study, we propose a new restart mechanism to provide a tactical balance between the exploitation and the exploration of the research, which are sometimes conflicting goals. We introduce a statistical metric called *coefficient of variation (CV)* (Everitt & Skronidal, 2010). It is a standardized measure that refers to the population dispersion degree. It is defined as the ratio of

the standard deviation σ to the mean μ of the population :

$$CV = \frac{\sigma}{\mu} \times 100\% \quad (14)$$

Where:

$$\mu = \frac{1}{PopSize} \sum_{\pi_i \in Pop} Cost(\pi_i) \quad (15)$$

$$\sigma = \sqrt{\frac{1}{PopSize - 1} \sum_{\pi_i \in Pop} (Cost(\pi_i) - \mu)^2} \quad (16)$$

The *CV* attempts to tune the search process by controlling, periodically in *CycleGen* iterations, the population dispersion degree to enhance, according to its value, either *diversification* or *intensification* to stabilize the population convergence throughout the search, as used by (Ladj, Benbouzid-Si Tayeb, & Varnier, 2016).

Populations with $CV < \varepsilon_{min}$, are considered to be of a low dispersion, i.e. individuals are of a great similarity and concentrated in small research space region. In this case, we apply an immune operator called Receptor Editing (De Castro & J., 2002). Its goal is lessening the risk of premature convergence by providing more widespread searching. It consists in eliminating a number ($Rst\% \times PopSize$) of worst individuals in the renewal population and replacing them by randomly created ones at the same number to cover other search regions. This mechanism allows us to find new schedules that correspond to new search regions in the entire search space.

On the other hand, populations with $CV > \varepsilon_{max}$ are considered to be high diversified, i.e. individuals cover distinct research space regions. In this case, we must promote promising regions exploitation. $Rst\% \times PopSize$ new individuals are generated by mutations of best solutions and are injected in the population to enhance its quality.

4.6. Stopping criteria

In traditional GA, either computation time or the number of generations is selected as termination criterion. Our algorithm terminates after *MaxGen* generations.

5. COMPUTATIONAL RESULTS

In this section, we present the results of series of computational experiments, conducted to test both designed approaches (optimal and sub-optimal). They were tested on a PC with Intel® Core™ i3-2330M CPU @ 2.20 GHz and 2.00 GB RAM. The proposed MILP was coded in C++ and implemented using GUROBI optimization solver (GUROBI, 2014).

In the following, we will first describe how test data are gen-

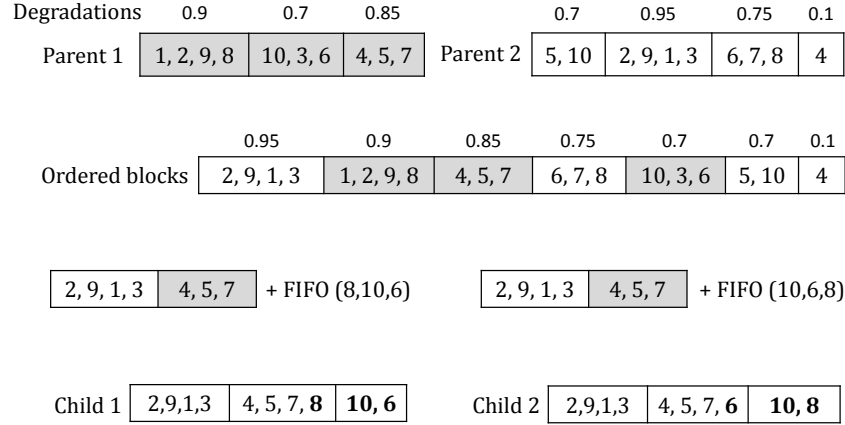


Figure 6. Crossover operator example

erated. Secondly, we analyze performance of our newly proposed *Pro-GA*. In this analysis phase experiments results are described : the calibration process, a comparison between *Pro-GA* and a standard GA (without restart scheme) as well as results of large size problems compared to a proposed lower bound. Next, a comparison between the *MILP* and *Pro-GA* is depicted for small problem instances. Finally, robustness analysis of *Pro-GA* is studied for a different benchmark.

5.1. Data generation

We generate a variety of random testing instances where:

- Size of problem instances $n \in [20, 300]$;
- Processing time of jobs is selected from a uniform distribution $p_i \in U[1, 50]$;
- RUL of each job is selected from a uniform distribution $RUL_i \in U[100, 150]$ for $n \in [20, 100]$, $RUL_i \in U[100, 200]$ for $n \in]100, 200]$, $RUL_i \in U[100, 250]$ for $n \in]200, 300]$;
- Initial machine degradation $\theta = 0$;
- Maintenance costs are set $C_f = 100$, $C_0 = 1000$.

10 instances are generated and tested for each problem size. We run 10 independent replicates of each instance in order to have a better view of the results. We average the results for all the given instances.

5.2. Pro-GA calibration

For the first set of experiments, we have undertaken a sensitive analysis of performance for our newly proposed algorithm *Pro-GA* by varying different parameters. These parameters are experimentally found to be good and robust for the

problems tested. 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 200);
- Initialization pourcentage α : 2 levels (70, and 80);
- Crossover probability $CrossProb$: 3 levels (0.7, 0.8, and 0.9);
- Mutation probability $MutProb$: 3 levels (0.01, 0.015, and 0.02);
- Replacement pourcentage β : 2 levels (20, and 30);
- Restart mechanism cycle $CycleGen$: 3 levels (10, 20, and 30);
- Low dispersion CV ε_{min} : 3 levels (10, 20, and 30);
- High dispersion CV ε_{max} : 3 levels (60, 70, and 80);
- Restart mechanism pourcentage Rst : 2 levels (15 and 25);
- Stopping criteria $MaxGen$: 3 levels (200, 300, and 400).

All the cited factors result in a total of $2 \times 2 \times 3 \times 3 \times 2 \times 3 \times 3 \times 3 \times 2 \times 3 = 11\ 664$ different combinations. Every combinaison is tested with a new set of problem instances randomly generated using the same data generation procedure described in the previous section where $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. 10 replicates of each problem size is executed which means 1 166 400 executions. The response variable of the experiment is the *Relative Percentage Deviation (RPD)* calculated using the following expression:

$$RPD = \frac{Cost_{sol} - Cost_{low}}{Cost_{low}} \times 100 \quad (17)$$

Where $Cost_{low}$ is a lower bound of the studied problem detailed in section 5.3.2.

The resulting experiment was analyzed by means of a multifactor analysis of variance (*ANOVA*) technique (Montgomery, 2008) 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. Figure (7) shows means plots for the three first greatest F-ratio parameters which are : $PopSize$, $MaxGen$, and $CrossProb$. The complete details are not reported for the sake of concise presentation.

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 : $PopSize = 200$; $MaxGen = 300$; $CrossProb = 0.7$; $MutProb = 0.015$; $\alpha = 80$; $\beta = 20$; $CycleGen = 20$; $Rst = 25$; $\varepsilon_{min} = 20$; $\varepsilon_{max} = 70$.

5.3. Performance analysis of Pro-GA

5.3.1. Comparison between Pro-GA and standard GA

Our newly proposed algorithm Pro-GA incorporates a restart scheme that seeks escaping from local optimum and stabilizes the population convergence throughout the search process. Thus, the second set of experiments are carried out to evaluate the effect of this mechanism on Pro-GA performance when compared to a standard genetic algorithm S-GA (without restart mechanism). Table 1 shows a comparison between the maintenance cost $Cost$ and the execution time CPU (in s) generated by Pro-GA and S-GA. Results show that, in all cases Pro-GA generates best solutions. S-GA yields a deviation of about 5% over $Cost$ obtained by Pro-GA. On the other hand, when observing the execution time CPU, we can find that Pro-GA is a little bit slower than S-GA due to the restart process, but it seems to be an acceptable compromise between solutions quality and execution time. Consequently, we can deduce that the embedded restart mechanism is well designed to guide the search process by enhancing its diversification and intensification throughout generations.

5.3.2. Comparison between Pro-GA and the proposed lower bound

Since no optimal solutions are known for the studied problem, we compare our GA results against a lower bound, denoted $Cost_{low}$. We note l_{low} the smallest number of blocks of capacity Δ required to process all jobs. In other words, if we suppose that all job blocks are completely full, i.e. their degradation is equal to Δ , then $l_{low} = \lceil \frac{1}{\Delta} \sum_{i=1}^n \delta_i \rceil$, and thus the total maintenance cost for all production batch, except the last one, will be fixed to cost C_f . Then, the low bound can be estimated by $Cost_{low} = (l_{low} - 1)C_f$.

Table 2 shows the comparison between the maintenance cost

Table 1. Comparison of $Cost$ and CPU obtained by Pro-GA and S-GA

n	Pro-GA		S-GA	
	$Cost$	CPU (s)	$Cost$	CPU (s)
20	303.54	2.51	315.75	1.88
40	650.23	3.89	677.18	3.55
60	1021.11	5.75	1085.21	5.56
80	1402.53	6.64	1465.55	6.29
100	1692.47	7.22	2530.01	6.78
120	2072.06	8.34	2194.19	6.93
140	2443.98	7.43	2580.85	7.13
160	2662.57	6.77	2810.17	7.08
180	3083.03	7.51	3018.21	7.11
200	3493.14	7.70	3681.44	7.24
250	4354.08	7.82	4620.85	7.32
300	5615.45	7.92	5920.63	7.37

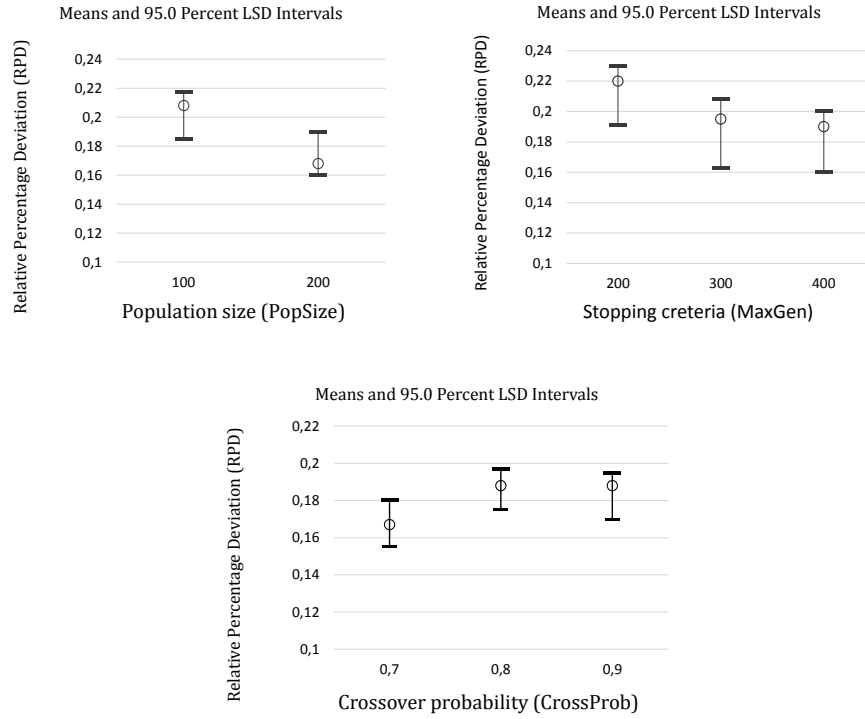
Table 2. Comparison of $Cost$ obtained by Pro-GA and the lower bound

n	$Cost$	$Cost_{low}$	% \nearrow
40	650.23	650	0.035
60	1021.11	1020	0.108
80	1402.53	1400	0.181
100	1692.47	1690	0.146
120	2072.06	2070	0.099
140	2443.97	2440	0.163
160	2662.57	2660	0.097
180	3083.02	3080	0.098
200	3493.14	3490	0.090
250	4354.08	4350	0.094
300	5615.44	5610	0.097

$Cost$ generated by Pro-GA and the lower bound $Cost_{low}$. One can easily observe that Pro-GA yields a very small deviation from the lower bound. In worst cases, predictive maintenance costs increase by less than 0.2%. Moreover, in several cases, this deviation is less than 0.1%. That confirms the efficiency of our GA to generate best solutions for all problem instances. This is argued by the correct parameters setting and the choice of appropriate genetic operators, especially the crossover that guarantees inheritance of good features through generations.

5.4. Comparative analysis of Pro-GA and MILP for small size problems

The third set of experiments reported in Table 3 were conducted to evaluate Pro-GA performance compared to the optimal results obtained by the MILP for small problem sizes $n \in \{5, 10, 12, 15, 18, 20\}$. We can see a comparison of the

Figure 7. Means plots for *PopSize*, *MaxGen*, and *CrossProb*

execution time *CPU* (in *s*) and the total maintenance cost *Cost*. These results are the average of 10 instances for each problem size. For $n \leq 12$, it is clear that *Pro-GA* is slowest since it manipulates a large set of individuals on which it applies greedy genetic operators during the whole process. Indeed, our GA must run $MaxGen = 300$ generations in all cases. This number of generation is very big for small problems. We can overcome this problem by using a different stopping criteria which is for example a *CPU* time limit fixed according the problem size. For $n > 12$, *Pro-GA* seems to be faster and its execution time becomes more efficient compared to *MILP*. For maintenance cost optimization, *Pro-GA* sub-optimal solutions yield a very small deviation over the optimal solutions given by *MILP*. In several cases, *Pro-GA* could find the optimal solutions. Moreover, in worst cases, this deviation is less than 0.1%.

5.5. Robustness analysis of Pro-GA for different setup

In order to know the claimed performance of our newly proposed Pro-GA is sensitive to how the data are generated, we run a second set of benchmarks generated differently. While PHM outputs (RULs) are generated with uniform distribution for the first experiments set, we use for this set of problem instances the 3-parameter Weibull distributions to obtain failure probabilities of the machine when processing each kind of

Table 3. Comparison of *CPU* and *Cost* between *MILP* and *Pro - GA*

<i>n</i>	<i>MILP</i>		<i>Pro - GA</i>		<i>Cost</i> % ↗
	<i>CPU(s)</i>	<i>Cost</i>	<i>CPU(s)</i>	<i>Cost</i>	
5	0.002	27.77	0,97	27.80	0.090
10	0.021	111.65	1.05	111.74	0.081
12	0.088	153.91	1.57	154.07	0.099
15	1.4	242.72	1.31	242.94	0.092
18	5.98	284.11	2.13	284.39	0.096
20	120.41	303.24	2.51	303.54	0.099

job. Weibull distribution is often used in the literature to estimate systems lifetime (Sidibe, Khatab, Claver, & Ait-Kadi, 2015; Khatab, Ait-Kadi, & Rezg, 2014; Berdinyazov et al., 2009). Probability density function of Weibull distribution is given in the Eq.(18).

$$f(t, k, \lambda, \theta) = \frac{k}{\lambda} * \left(\frac{t - \theta}{\lambda}\right)^{k-1} * e^{-\left(\frac{t - \theta}{\lambda}\right)^k} \quad (18)$$

For each job, machine degradation is estimated by the failure probability corresponding to its processing time (see Figure 8). The shape, scale and position parameters of Weibull

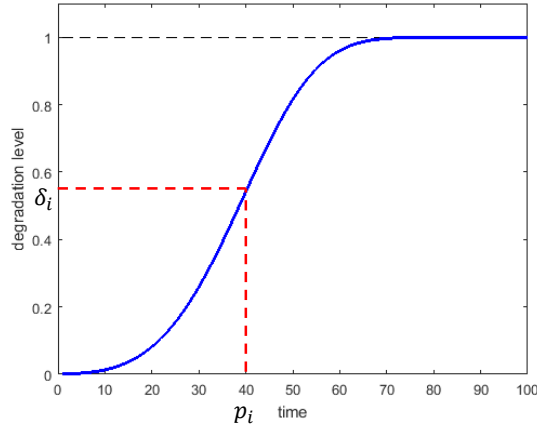


Figure 8. Example of failure probability estimation (Weibull distribution $k = 4$, $\lambda = 50$, $\theta = -8$)

distributions are selected as follow:

- Shape parameter : $k \in [2, 10]$;
- Scale parameter $\lambda \in [20, 50]$;
- Position parameter $\theta \in [-10, 0]$.

Results shown in Figure 9 and detailed in Table 4 compare deviations of maintenance cost obtained by Pro-GA over the proposed lower bound for both benchmarks sets. For each problem size n , PHM outputs are generated either by Uniform distribution (setup 1) or Weibull distribution (setup 2) for the same production data (processing times). When comparing the effect of data generation method on our Pro-GA effectiveness, we remark that deviations are almost equivalent. For both setups, Pro-GA yields a very small deviation over the lower bound. In worst case, this deviation is less than 0.19%. Consequently, we can deduce that the performance of Pro-GA remains stable for different setups thanks to the good operators choice and appropriate calibration. This prove the robustness of Pro-GA.

6. CONCLUSION

In this paper we have proposed a Mixed Integer Linear Programming (*MILP*) model and new prognostic based genetic algorithm *Pro-GA* to solve the integrated production and predictive maintenance scheduling problem on a single machine under the total cost minimization criterion. Since each kind of production requires specific machine functionalities, we have assumed that a PHM system provides the RUL corresponding to each kind of production and then a relative degrada-



Figure 9. Comparison of deviations obtained by *Pro - GA* for Uniform and Weibull distributions

Table 4. Comparison of deviations obtained by *Pro - GA* for Uniform and Weibull distributions

n	setup 1 (Uniform) % ↗	setup 2 (Weibull) % ↗
40	0.035	0.054
60	0.108	0.110
80	0.181	0.165
100	0.146	0.123
120	0.099	0.101
140	0.163	0.186
160	0.097	0.082
180	0.098	0.130
200	0.090	0.077
250	0.094	0.087
300	0.097	0.107

tion value is calculated. A predictive intervention is scheduled whenever the maximal authorized threshold is reached. The designed *MILP* is able to compute optimal solutions for problem sizes $n \leq 20$. To deal with larger instances, *Pro-GA* include carefully designed operators in order to enhance the quality of the obtained solutions. We have conducted various experiments that showed the efficiency *Pro-GA* compared to a lower bound. Robustness of our algorithm has also been proved throughout different benchmark.

Further topics would be continued with regards to this results. The proposed integrated scheduling model can be extended to manage other typologies of production systems. Another work can deal with the uncertain character of the PHM outputs because it is important to rigorously deal with these uncertainties in order to create a robust scheduling using fuzzy

logic.

REFERENCES

- Aliustaoglu, C., Ertunc, H. M., & Ocak, H. (2009). Tool wear condition monitoring using a sensor fusion model based on fuzzy inference system. *Mechanical Systems and Signal Processing*, 23(2), 539–546.
- Benbouzid-Sitayeb, F., Guebli, S. A., Bessadi, Y., Varnier, C., & Zerhouni, N. (2011). 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*, 6(1), 30–48.
- Berdinyazov, A., Camci, F., Sevkli, M., & Baskan, S. (2009, Aug 2009). Economic analysis of maintenance policies for a system. In *2009 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives* (p. 1-5).
- Brotherton, T., Jahns, G., Jacobs, J., & Wroblewski, D. (2000). Prognosis of faults in gas turbine engines. In *Aerospace conference proceedings, IEEE* (Vol. 6, pp. 163–171).
- Byington, C. S., Roemer, M. J., & Galie, T. (2002). Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. In *2002 IEEE Aerospace Conference Proceedings* (Vol. 6, pp. 6–6).
- CEN/EN. (2010). *Maintenance - maintenance terminology* (Tech. Rep. No. EN13306:2010). European committee for standardization.
- Chrétien, S., Herr, N., Nicod, J. M., & Varnier, C. (2015, 22–25 Jun 2015). 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)* (pp. 1–7). TX, USA.
- Coffman, E. G., Garey, M. R., & Johnson, D. S. (1984). Approximation algorithms for bin-packing - an updated survey. In G. Ausiello, M. Lucertini, & P. Serafini (Eds.), *Algorithm design for computer system design* (pp. 49–106). Vienna: Springer Vienna.
- De Castro, L. T., & J. (2002). Artificial immune systems: A new computational intelligence approach. In *Algorithm design for computer system design*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Duffuaa, S., Ben-Daya, M., Al-Sultan, K., & Andijani, A. (2001). A generic conceptual simulation model for maintenance systems. *Journal of Quality in Maintenance Engineering*, 7(3), 207–219.
- Everitt, B., & Skrondal, A. (2010). *The Cambridge dictionary of statistics* (3th ed.).
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- GUROBI. (2014). *Optimizer reference manual*. Retrieved 2016-12-28, from <http://www.gurobi.com/documentation/>
- Hadidi, L. A., Al-Turki, U. M., & Rahim, A. (2011). Integrated models in production planning and scheduling, maintenance and quality: a review. *International Journal of Industrial and Systems Engineering*, 10(1), 21–50.
- Herr, N., Nicod, J. M., & Varnier, C. (2014, 18–22 Aug 2014). Prognostics-based scheduling in a distributed platform: Model, complexity and resolution. In *IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 1054–1059). Taipei, Taiwan.
- ISO. (2004). *Condition monitoring and diagnostics of machines, prognostics part 1: General guidelines* (Vol. ISO/IEC Directives Part 2; Tech. Rep. No. ISO13381-1). International Organization for Standardization.
- Iyer, N., Goebel, K., & Bonissone, P. (2006, 4–11 March 2006). Framework for post-prognostic decision support. In *2006 IEEE Aerospace Conference* (p. 10 pp.-). MT, USA.
- Khatab, A., Ait-Kadi, D., & Rezg, N. (2014). Availability optimisation for stochastic degrading systems under imperfect preventive maintenance. *International Journal of Production Research*, 52(14), 4132–4141.
- Ladj, A., Benbouzid-Si Tayeb, F., & Varnier, C. (2016, 24–29 July 2016). An integrated prognostic based hybrid genetic-immune algorithm for scheduling jobs and predictive maintenance. In *IEEE Congress on Evolutionary Computation* (pp. 2083–2089). Vancouver, BC, Canada.
- Lebold, M., & Thurston, M. (2001, 6–9 May 2001). Open standards for condition-based maintenance and prognostic systems. In *Maintenance and Reliability Conference (MARCON)* (Vol. 200). Gatlinburg, Tennessee, USA.
- Liu, S., Zhang, H., Li, C., Lu, H., & Hu, Y. (2014). Fuzzy reliability estimation for cutting tools. *Procedia CIRP*, 15, 62–67.
- Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58(2), 199–211.
- Merten, A. G., & Muller, M. E. (1972). Variance minimization in single machine sequencing problems. *Management Science*, 18(9), 518–528.
- Michalewicz, Z., & Hartley, S. J. (1996). Genetic algorithms+ data structures= evolution programs. *Mathematical Intelligencer*, 18(3), 71.
- Montgomery, D. C. (2008). *Design and analysis of experiments*. John Wiley & Sons.
- Muller, A., Suhner, M. C., & Iung, B. (2008). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering & System Safety*, 93(2), 234–253.

- Pan, E., Liao, W., & Xi, L. (2012). A joint model of production scheduling and predictive maintenance for minimizing job tardiness. *The International Journal of Advanced Manufacturing Technology*, 60(9), 1049–1061.
- Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing. *Computers & operations research*, 22(1), 5–13.
- Rohlfshagen, P., & Bullinaria, J. A. (2010). Nature inspired genetic algorithms for hard packing problems. *Annals of Operations Research*, 179(1), 393–419.
- Ruiz, R., García-Díaz, J. C., & Maroto, C. (2007). Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers & Operations Research*, 34(11), 3314–3330.
- Ruiz, R., Maroto, C., & Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, 34(5), 461–476.
- Sardinas, R. Q., Santana, M. R., & Brindis, E. A. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence*, 19(2), 127–133.
- Sidibe, I., Khatab, A., Claver, D., & Ait-Kadi, D. (2015). A maintenance optimization model for a second hand stochastically deteriorating system under different operating environments. In *Control, engineering & information technology (ceit), 2015 3rd international conference on* (pp. 1–5).
- Tobon-Mejia, D., Medjaher, K., & Zerhouni, N. (2012). Cnc machine tool's wear diagnostic and prognostic by using dynamic bayesian networks. *Mechanical Systems and Signal Processing*, 28, 167–182.
- Varnier, C., & Zerhouni, N. (2012, 23–25 May 2012). Scheduling predictive maintenance in flow-shop. In *Ieee conference on prognostics and system health management (phm)* (pp. 1–6). Beijing, China.
- Venkatasubramanian, V. (2005). Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities. *Computers & chemical engineering*, 29(6), 1253–1263.