



HAL
open science

An Integrated Guided Local Search considering Human Resource Constraints for the Single-machine Scheduling problem with Preventive Maintenance

Meriem Touat, Fatima Benbouzid-Si Tayeb, Belaïd Benhamou, Lamia Sadeg, Salima Aklil, Meryem Karaoui

► To cite this version:

Meriem Touat, Fatima Benbouzid-Si Tayeb, Belaïd Benhamou, Lamia Sadeg, Salima Aklil, et al.. An Integrated Guided Local Search considering Human Resource Constraints for the Single-machine Scheduling problem with Preventive Maintenance. International Conference on Systems, Man and Cybernetics (SMC), Oct 2019, Bari, Italy. pp.3799-3804, 10.1109/SMC.2019.8914261 . hal-02450017

HAL Id: hal-02450017

<https://hal.science/hal-02450017>

Submitted on 21 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Integrated Guided Local Search considering Human Resource Constraints for the Single-machine Scheduling problem with Preventive Maintenance

Meriem Touat^{1,2,3} and Fatima Benbouzid-Si Tayeb² and Belaid Benhamou³ and Lamia Sadeg and Salima Aklil and Meryem Karaoui

Abstract—This work concerns the consideration of human resource constraints in the single machine scheduling problem of both production and flexible periodic maintenance activities. We assume that a maintenance activity requires the intervention of a human resource to be treated. These human resources are characterized by a competence level and availabilities considered as strong constraints allowing or not the maintenance activities' planning. To solve this NP-hard scheduling problem, we propose a guided local search metaheuristic that embeds a post-optimization process in order to minimize both production and maintenance delays. We implemented and experimented the proposed method on two series of benchmarks. The first one focuses on small size instances. The results show that the quality of the solutions obtained by the proposed method compared to an exact one is good, and even it reaches the optimal solution in some cases. In the second one, we applied the method on large instances to show its advantages and efficiency.

I. INTRODUCTION

Production scheduling and preventive maintenance planning are the most common and significant problems faced by the manufacturing industry. This paper focuses on a single-machine scheduling problem in which preventive maintenance activities should be performed on the machine. In the literature, several variants of the single-machine with preventive maintenance (PM) have been investigated. Preventive maintenance activities could be integrated in the production scheduling into two different ways. The former is the deterministic way where maintenance periods are fixed in advance [Ma *et al.*2010], [Liu *et al.*2015], [Yazdani *et al.*2017]. In the second, maintenance periods are decision variables, this is called scheduling with flexible maintenance [Chen2008], [Mashkani and Moslehi2016], [Luo *et al.*2015].

The majority of research works on scheduling with preventive maintenance assume a perfect environment in terms of resource availability. However, maintenance service requires agents to ensure the execution of activities. These agents should be versatile and be able to accomplish several interventions. Indeed, they are characterized by competence levels and

qualifications allowing them or not to execute the maintenance activities with different durations. In addition, these human resources could not be available permanently in the shop, but in specified intervals that assess the feasibility of the maintenance intervention. It may therefore be necessary to consider the human resources in scheduling problems or even more as a scheduling problem separately (the workforce scheduling or employee timetabling problems [Yurtkuran *et al.*2018], [Ciro *et al.*2015], [Guyon *et al.*2014]). To the best of our knowledge, research on the integrated production and maintenance scheduling problems under human resource constraints has been introduced recently [Touat *et al.*2017a], [Touat *et al.*2017b], [Touat *et al.*2018] and research works on it still very scarce.

This paper deals with the single machine scheduling problem and flexible maintenance planning, which presents a building block for more complex scheduling problem, where each maintenance activity is treated by a human resource characterized by a competence and some availability intervals. These unavailabilities are due to the fact that human resources are responsible to execute other tasks in the workshop. The contributions of this paper are twofold. First, we propose an integrated Guided Local Search (hereafter IGLS) to solve the studied problem favoring the maintenance efficiency by choosing the humans having the highest competence that leads to a short maintenance duration. Moreover, we embedded a post optimisation procedure to minimize both production and maintenance activities delays.

In the rest of the paper, we describe the studied problem, present the proposed resolution method, discuss the obtained results then concludes the work and gives some research perspectives.

II. THE PROBLEM DESCRIPTION

We consider here, a single-machine scheduling problem subject to flexible and periodic maintenance. Formally, the addressed problem can be described as follows: Let $J = J_1, J_2, \dots, J_N$ be a set of N jobs to be processed on a single machine. All the jobs are available at time zero. Each job J_i requires a given known deterministic and non negative processing time p_i , a completion time c_i and should be completed before a due date d_i . Preemption is not allowed and the machine can handle at most one job at a time and cannot stand idle until the last job is finished.

¹ Ecole Supérieure des Sciences Appliquées Alger (ESSA-Alger), BP 474, Place des Martyrs, Alger 16001, 1er Novembre, Algérie m.touat@g.essa-alger.dz

² Laboratoire des Méthodes de Conception des Systèmes (LMCS), Ecole nationale Supérieure d'Informatique (ESI), BP 68M 16309, Oued-Smar, Alger, Algérie f_sitayeb@esi.dz

³ Aix-Marseille Université, Laboratoire LSIS, Domaine Universitaire de Saint-Jerome, Batiment Polytech, Avenue Escadrille Normandie-Niemen, 13397 MARSEILLE CEDEX 20, France belaid.benhamou@univ-amu.fr

Besides, preventive maintenance must be undertaken in order to maintain a high availability of the machine. In this paper, we consider a single flexible maintenance M with multiple occurrences $M_i, i \in \{N + 1, \dots, N + Nb_Occ\}$ occurring every T periods and each occurrence depends on the ones preceding it on the machine. Each maintenance activity M_i requires a given known deterministic and non negative processing time p'_i and a completion time c'_i . Moreover, a maintenance M_i must be completed within a time window $TI_i = [Tmin_i, Tmax_i]$ representing its tolerance interval. It is achieved when the maintenance activity is more profitable and before the equipment loses its optimum performance. However, it can be planned before $Tmin_i$ and it is considered in advance (this Earliness is noted E'_i), or after $Tmax_i$ and it is considered late (this Tardiness is noted T'_i). We assume that the first time-window is arranged in advance. The i^{th} time window depends on the completion time of the $(i - 1)^{th}$ maintenance occurrence M_{i-1} (Eq. 1). Since we seek scheduling over a production horizon, we do not perform a maintenance operation after the processing of the last job.

$$\begin{cases} Tmin_i = c'_{i-1} + T \\ Tmax_i = Tmin_i + (Tmax_{i-1} - Tmin_{i-1}) \end{cases} \quad (1)$$

A maintenance M_i must be treated by one human resource. The maintenance service is composed of R human resources (HR). Each human resource HR_r ($r = 1..R$) is characterized by a competence level $Comp_r$ allowing to execute a maintenance task with a duration ph_r depending on the competence level, such as: $ph_r = p' \div Comp_r$. Note that higher is the competence level lower is the maintenance duration. Moreover, each resource HR_r has a timetabling which determines its availability. This is expressed by specifying for each resource HR_r a set $AI_r = \{AI_{rl} : l = 1..m\}$ of m availability intervals (AI). More precisely, $AI_r = \{[LB_{r1}, UB_{r1}], \dots, [LB_{rm}, UB_{rm}]\}$. The symbols LB_{rl} and UB_{rl} denote respectively, the lower and the upper bounds of the availability interval AI_{rl} ($l = 1..m$). In our problem, we define the efficiency according to the executing times of maintenance activities ph_r . The choice of the human resource to perform a maintenance activity does not depend on its availability only. It is possible in practice that several resources are available at a time. In this case, we choose the resource with the highest competence to perform the maintenance activity.

The production objective f_p is to find a permutation of N production jobs that minimizes the sum of tardiness T_i , when the schedule also includes maintenance activities (eq.2).

$$\begin{cases} f_p = \sum_{i=1}^N T_i \\ T_i = \max(0, c_i - d_i) \end{cases} \quad i = 1..N \quad (2)$$

The maintenance objective f_m consists in minimizing the sum of earliness/tardiness of all the occurrences of the maintenance activities with respect to the pre-specified maintenance intervals. The maintenance tasks are planned by taking into account the human resource constraints based on

their availabilities. i.e. the maintenance activity is assigned to the closest available resource (eq.3).

$$\begin{cases} f_m = \sum_{i=N+1}^{N+Nb_Occ} (E'_i + T'_i) \\ E'_i = \max(0, Tmin_i - t'_i) & i = N + 1..N + Nb_Occ \\ T'_i = \max(0, c'_i - Tmax_i) & i = N + 1..N + Nb_Occ \end{cases} \quad (3)$$

To optimize both production and maintenance criteria, we consider the global function defined as follows (eq.4):

$$\begin{cases} f = \alpha \times f_p + \beta \times f_m \\ \alpha + \beta = 1 \end{cases} \quad (4)$$

The studied problem is NP-hard since the basic one without considering the maintenance activities denoted $N/1/d_i/\sum T_i$ based on the classification given in [Graham *et al.*1979], was proved NP-hard [Du and Leung1990].

III. THE PROPOSED APPROACH

Guided Local Search (GLS) [Voudouris and Tsang1995] is a penalty-based metaheuristic algorithm that sits on top of other local search algorithms. GLS associates a set of M features with every solution S , a feature f_{if} is determined by the *indicator function* I_{if} , $if \in \{1, \dots, M\}$, which is equal to 1 if S has the property f_{if} and 0 otherwise. GLS penalizes solution features during each iteration based on the value of an utility function $util_{if}(S)$. The penalty acts as a disturbance to an augmented objective function $h(S)$ which is adjusted during each iteration. This reduces the chance that the solution procedure will get stuck in a local optimum.

GLS algorithms were successfully applied to optimization problems [Voudouris and Tsang1997], [Alsheddy and Tsang2011], [Nagata and Ono2018]. However, to the best of our knowledge, no work considers the single machine scheduling problem either with flexible maintenance or under human resources constraints. In this section, we present the proposed GLS meta-heuristic for the integrated production and flexible maintenance scheduling problem where the human agents in charge of executing the maintenance activities are taken into account. These human resources are characterized by availability periods and competence levels. The main features of our IGLS compared to the standard one are the following:

- The use of an integrated representation of production and maintenance data which embeds human resource constraints.
- The definition of two types of features for each production and maintenance activity, one is related to the advance and the other to the delay.
- The use of three local search heuristics (LS) instead of one. In each iteration of IGLS, a specific LS heuristic is applied according to the feature which corresponds to maximum utility.
- The introduction of post-optimization process in order to minimize both production and maintenance delays (intensifying). This process is performed at the end of each local search.

3	6	7:30	1	5	8:53	4	9:129	2
---	---	------	---	---	------	---	-------	---

	AI ₁	AI ₂	AI ₃
HR ₁	0	8	0
HR ₂	7	9	0

Fig. 1: An example of a solution encoding

In the next sub-sections, we describe the components of the proposed meta-heuristic IGLS and then give its pseudo-code scheme in Algorithm 4.

A. THE ENCODING SCHEME AND INITIAL SOLUTION GENERATION

In IGLS, each solution is expressed by two substructures described in the following:

- 1) An integrated sequence which consists of two parts with different lengths as shown in Fig. 1. The first part is made up production jobs identified by numbers 1 to N and the second one of maintenance occurrences identified by numbers $N + 1$ to $L = N + Nb_Occ$. Moreover, to be able to compute the maintenance activities advance/delay, we add to the maintenance identifier the execution time t_i .
- 2) To take into account the human resource constraints, we associate to the integrated sequence an assignment matrix \mathbf{Z} with size $R \times m$. \mathbf{R} is the number of human resources and \mathbf{m} is the number of availability intervals of each resource (see section II). $Z[r, l]$ identifies the maintenance activity treated by resource HR_r during the interval AI_{rl} . $Z[r, l] = 0$ means that the resource HR_r does not treat any maintenance activity during the interval AI_l .

To generate an initial integrated solution, first, the maintenance occurrences are planned according to the human resource availabilities. Our aim is to minimize the maintenance' earliness/tardiness even the human resource availability intervals and the tolerance period do not overlap. Then, the production jobs are scheduled randomly based on the maintenance activity planning.

B. LOCAL SEARCH HEURISTICS

IGLS associates three sets of features F_{if} according to the objective function, where $if \in \{1, 2, 3\}$. One feature is related to production jobs and the two others to maintenance activities. Each feature in F_1 (respectively F_2) represents the delay of a production (respectively maintenance) activity. Thus, F_1 has N features and F_2 has Nb_Occ features, the total number of production (respectively maintenance) activities. The third feature F_3 represents the advance of maintenance activities. There are Nb_Occ features in F_3 . Consequently, to every solution S , it is associated $(N + 2 \times Nb_Occ)$ features. Each feature $F_{if,i}$ is associated with an indicator $I_{if,i}(S)$, $if \in \{1, 2, 3\}$, which is equal to 1 if S has the property $F_{if,i}$

and 0 otherwise. A cost $c_{if,i}$ and a utility value (eq.5):

$$util_{if,i}(S) = I_{if,i}(S) \times \frac{c_{if,i}}{1 + pn_{if,i}} \quad (5)$$

Where $pn_{if,i}$ is the penalty associated to $F_{if,i}$ and: $i = 1..N + Nb_Occ$. The augmented objective function is adjusted during each iteration (eq.6):

$$h(S) = f(S) + \lambda \sum_{i=1}^{i=3} \sum_{if=1}^M pn_{if,i} \times I_{if,i} \quad (6)$$

Three LS heuristics are proposed based on each feature. Each of them has a different neighborhood structure generated according to the feature having the highest utility. If several ones have the highest utility, we choose randomly an appropriate neighborhood structure.

THE LOCAL SEARCH LS_1

This LS is performed if the activity with maximum utility is a production job with the goal of production objective function improvement. Let J_i be this job. The objective is to insert J_i in the integrated sequence that minimizes best its delay, i.e. its completion time c_i is equal to its due date d_i , or it is as close as possible to it (Algorithm 1).

Algorithm 1 The pseudo code of LS_1 .

Input: The current solution S_c , The activity J_i , $h(S_c)$, The scheduling data (production, maintenance and human resource);

Output: The solution S_c ;

- 1: **if** (The activity scheduled at $d_i - p_i$ is a production job) **then**
 - 2: **if** (there is a maintenance activity planned in $[d_i - p_i, d_i]$) **then**
 - 3: Schedule J_i at the position coming just before this maintenance activity;
 - 4: **else**
 - 5: Schedule J_i at $d_i - p_i$;
 - 6: **end if**
 - 7: **else**
 - 8: Schedule J_i at the position coming just before the maintenance activity;
 - 9: **end if**
 - 10: $S_v \leftarrow$ schedule of the production jobs in the same sequence that S_c ;
 - 11: **if** ($h(S_v) < h(S_c)$) **then**
 - 12: $S_c \leftarrow S_v$;
 - 13: **end if**
 - 14: Return S_c ;
-

THE LOCAL SEARCH LS_2 (RESPECTIVELY LS_3)

LS_2 (respectively LS_3) is performed if the activity with a maximum utility is a maintenance occurrence in delay (respectively in advance) according to its tolerance interval with the goal of maintenance objective function improvement. Let M_i be this maintenance activity and let $[Tmin_i, Tmax_i]$ be its tolerance interval. We assume that M_i is executed in the availability interval AI_{lr} . In order to reduce the maintenance activity tardiness (respectively earliness), we try to advance (respectively delay) it in the same availability interval of the same assigned human resource if possible, or in the most nearest one (Algorithm 2).

Algorithm 2 The pseudo code of LS_2 (respectively LS_3).

Input: The current solution S_c , The activity M_i planned in the availability interval AI_{rl} , $h(S)$, The scheduling data (production, maintenance and human resource);

Output: The solution S_c ;

```
1: if ( $M_i$  can be advanced (respectively delayed) in  $AI_{rl}$ ) then
2:    $S_v \leftarrow S$  with the new maintenance planning;
3: else
4:    $S_v \leftarrow S_c$  with the planning of  $M_i$  at the unoccupied
   availability interval coming just before (respectively after)  $AI_{rl}$ 
   and that does not disturb the planning of  $M_{i-1}$  (respectively
    $M_{i+1}$ );
5: end if
6: Update the tolerance intervals  $TI_{i+1}$  for the maintenance activity
    $M_{i+1}$ ;
7: Update the human resource availabilities;
8:  $S_v \leftarrow S_v$  with production job scheduling in same order that
    $S_c$ ;
9: if ( $h(S_v) < h(S)$ ) then
10:   $S_c \leftarrow S_v$ ;
11: end if
12: Return  $S_c$ ;
```

Moreover, in order to reduce the idle times of the machine, looking for the production objective function improvement, for each maintenance activity M_i , we try to delay it in the same availability interval and advance the production ones scheduled after it, if possible, or advance M_i as possible. The solution S obtained after the post-optimization procedure is compared to the one issued from the local search S_c , if it is better, we replace S_c by S as explained in Algorithm 3.

C. THE REGULATION PARAMETER AND STOPPING CRITERIA

In the IGLS algorithm, we propose two different ways to manage the regulation parameter λ : In a static way, where λ is assigned a fixed value or in a dynamic way where the value of λ varies during the resolution process. Finally, if a solution cannot be improved any more in consecutive $Nb_{Improve}$ iterations, the algorithm terminates, otherwise it stops after a Nb_{It} iterations.

IV. EXPERIMENTAL RESULTS

In this section, we present some experimental results to show the effectiveness of the proposed IGLS in solving the integrated single-machine scheduling problem with flexible maintenance under the human resource constraints. The tests are performed on a personal computer with an *Intel Core i7 2.70 GHz* CPU and *16 Gb* RAM memory under *Windows 7* operating system.

The experiments are conducted by considering the same data used in [Touat *et al.*2018]. Thus, we used two types of data. The first one related to production/maintenance data, while the second one is related to the human resource ones. We experimented both small size instances ranging from 9 to 13 jobs; and large ones ranging from 20 to 700 jobs with 10 instances for each considered benchmarks. Due to the lack on human resources benchmarks, the availability intervals are generated according to the production horizon.

Algorithm 3 The pseudo code of the post optimization procedure.

Input: The initial solution S_c , The scheduling data (production, maintenance and human resource);

Output: The solution S_c ;

```
1: for (Each idle period  $id$  between the maintenance occurrence
    $M_i$ ,  $i = N + 1 \dots N + Nb\_Occ$  and the previous production job
    $J_j$ ) do
2:   Advance the production jobs coming after  $M_i$  to the idle
   period as possible;
3:   Update the tolerance intervals  $TI_{i+1}$ ;
4:   if (It is possible to delay  $M_i$  in its  $AI_{rl}$  generating an idle
   time  $id$ ) then
5:     Delay  $M_i$  and advance the production jobs coming after
    $M_i$  to the new idle period as possible;
6:     Update the tolerance intervals  $TI_{i+1}$ ;
7:   end if
8:   if (No production job is advancing) then
9:     Advance  $M_i$  in  $AI_{rl}$  if it is possible;
10:    Update the tolerance intervals  $TI_{i+1}$ ;
11:   end if
12: end for
13:  $S_v \leftarrow S_c$  with the new maintenance planning and production
   scheduling;
14: if ( $f(S_v) < f(S_c)$ ) then
15:    $S_c \leftarrow S_v$ ;
16: end if
17: Return  $S_c$ ;
```

Thereby, for each integrated production and maintenance instance, a human resource one is specified. We consider two human resources (HR_1, HR_2) with respectively two competence levels ($Comp_1, Comp_2$). Indeed, the existence of only one human resource in the workshop could lead to overwork whereas in practice the assignment of human resources to tasks follows the workload regulation. As well, the existence of three human resources or more means a high availability. This is why we choose two human resources to ensure that there is always an available resource even if not always at the needed time. Both $Comp_1$ and $Comp_2$ are distributed uniformly in $]0, 2[$. In order to execute the occurrence M_i by a human resource, two different durations are possible: ph_1 and ph_2 . That is, $ph_1 = p' \div Comp_1$ and $ph_2 = p' \div Comp_2$. We propose two types of availability intervals according to their wideness: the strict availability interval (SAI), and the large availability interval (LAI). According to human resource characteristics: higher and lower human resource competences (respectively LC and HC) and strict and large availability intervals (respectively SAI and LAI), four classes of experiments (SAI/LC , SAI/HC , LAI/LC and LAI/HC) are performed.

To tackle the problem in a simplified way, the values of the control parameters α and β of equation 4 are set to 0.5. For the first set of experiments, we have undertaken a sensitive analysis of performance for the proposed algorithm by varying different parameters. The complete details are not reported for the sake of concise representation. We have chosen a full factorial design in which all possible combinations of the following factors are tested:

Algorithm 4 The pseudo-code of the proposed meta-heuristic IGLS.

Input: The scheduling data (production, maintenance and human resource);

Parameter: The GLS parameter (λ);

Output: The final schedule S_{best}

```

1:  $k \leftarrow 0$ ;
2: Generate an initial solution  $S_0$ ;
3: Set all penalties  $pn_{if,i}$  to 0,  $if = 1, 2, 3$ ,  $i = 1, \dots, N + Nb\_Occ$ ;
4:  $S_{best} \leftarrow S_0$ ;  $S_c \leftarrow S_0$ 
5: while (Termination criterion is not met) do
6:    $h(S) \leftarrow f(S_c) + \lambda \sum_{i=1}^3 \sum_{if=1}^M pn_{if,i} \times I_{if,i}$ ;
7:    $S_c \leftarrow Local\_Search(S_c, h(S), I(S))$ ;  $\triangleright$  perform the
   local search heuristic with the feature  $F_{if,i}$  corresponding the
   maximum utility  $util_{if,i}$  and the objective function  $h$ 
8:   Post-optimization ( $S_c$ );  $\triangleright$  perform the post optimization
   procedure
9:   for (Each feature  $F_{if,i}$ ) do  $\triangleright$  update features' utilities
10:      $util_{if,i}(S_c) = I_{if,i}(S_c) \times \frac{c_{if,i}}{1+pn_{if,i}}$ ;
11:   end for
12:    $util_{j,j'} = max_{if=1,2,3,i=1,\dots,N+Nb\_Occ}(util_{if,i})$ ;  $\triangleright$ 
   Compute the maximum utilities
13:    $pn_{j,j'} \leftarrow pn_{j,j'} + 1$ ;  $\triangleright$  penalize features with maximum
   utility
14:   if ( $f(S_c) < f(S_{best})$ ) then
15:      $S_{best} \leftarrow S_c$ ;
16:   end if
17:   if (Stagnation) then  $\triangleright$  restart schema after stagnation
18:     Disruption ( $S_c$ );
19:   end if
20: end while
21: Return  $S_{best}$ ;

```

- We try the different values of static λ between 0 and 1 with a step of 0.2 between the different values. The value $\lambda = 0.9$ gives, in general, better solutions than the results of the other static values. For this reason, in the following tests we take this value to represent the static values of λ . According to the dynamic value, we vary λ during the execution of the algorithm by computing the ratio between the cost of the best solution found " S_{best} " and the one of the current solution " S_c " (eq.7):

$$\lambda = \frac{f(S_{best})}{f(S_c)} \quad (7)$$

- $Nb_{It} = \begin{cases} 300 & \text{for } N \in \{9, 10, 11, 12, 13\} \\ 500 & \text{for } N \in \{20, 40, 60, 80\} \\ 1000 & \text{for } N \in \{100, 140, 160, 200, 300\} \\ 2000 & \text{for } N \in \{500, 700\} \end{cases}$
- $Nb_{Improve} = 20$.

A. COMPARATIVE ANALYSIS WITH THE CPLEX IMPLEMENTATION

The objective of these set of experiments is to evaluate the performance of the proposed IGLS compared to the optimal results obtained by the exact method Cplex [Touat *et al.*2018] on some small size instances $N \in \{9, 10, 11, 12, 13\}$. The relative percentage deviation (RPD) is used as an index to evaluate the solution quality and the performance of the

proposed heuristic (eq.8).

$$RPD = \frac{(f - f_{CP})}{f_{CP}} \times 100 \quad (8)$$

where f is the global objective function defined in Equation 4 and f_{CP} is the global objective function of the CP modeling. We note that the execution time of small benchmarks vary in $[0, 0.16]$ seconds. We perform tests for a static and dynamic values of λ . Table I shows the obtained results. For each value of λ (static and dynamic (D) values) and each benchmark, we report the average value of the global objective function obtained by running 10 instances. The columns corresponding to B give the number of instances among those of each class that have been solved optimally by IGLS. It should be noted that we performed $5 \times 10 \times 4 \times 4 = 800$ tests.

λ	N	SAI							
		LC			B	HC			B
		f_{CP}	f	RPD		f_{CP}	f	RPD	
0.9	9	423.95	445.50	5.08%	12	134.95	143.55	6.56%	12
	10	404.55	438.35	8.35%		147.55	155.40	5.32%	
	11	441.25	452.85	2.63%		174.70	185.60	6.24%	
	12	507.10	548.35	8.13%		216.00	247.05	14.38%	
	13	418.25	457.95	9.49%		183.25	193.00	5.32%	
AVG			6.73%				7.56%		
D	9	423.95	444.85	4.93%	15	134.95	144.30	6.93%	15
	10	404.55	437.40	8.12%		147.55	155.45	5.35%	
	11	441.25	459.00	4.02%		174.70	187.45	7.30%	
	12	507.10	535.25	5.55%		216.00	230.45	6.68%	
	13	418.25	453.80	8.50%		183.25	196.30	7.12%	
AVG			6.22%				6.67%		
λ	N	LAI							
		LC			B	HC			B
		f_{CP}	f	RPD		f_{CP}	f	RPD	
0.9	9	304.6	326.35	7.14%	10	106.8	115.55	8.19%	10
	10	284.65	307.95	8.19%		121.65	134	10.15%	
	11	356.3	377.7	6.01%		147.4	171.7	16.49%	
	12	368.15	405.85	10.24%		176.55	199.05	12.74%	
	13	349.45	372.85	6.7%		172	186.45	8.4%	
AVG			7.65%				11.19%		
D	9	304.6	331.75	8.91%	15	106.8	115.4	8.05%	10
	10	284.65	386.9	9.73%		121.65	129.9	6.78%	
	11	356.3	401.85	8.59%		147.4	165.5	12.28%	
	12	368.15	395.95	9.15%		176.55	197.7	11.98%	
	13	349.45	381.7	9.23%		172	184.45	7.24%	
AVG			9.12%				9.26%		

TABLE I: IGLS results on small size instance.

The obtained results show that for most of the tested instances, the RPD does not exceed 10% for all classes of experiments, except 6% of cases when λ is dynamic and 10% of cases when λ is static. In the average it varies between 5 and 10%. The best results of the IGLS are obtained for the *SAI/LC* class. It is even lower than 3% (2.63%) in the case of *SAI/LC* where $N = 11$ and λ is static.

For the 50 instances used for the tests, we remark that the proposed IGLS with dynamic value of λ succeeds to find more optimal solutions than in the case of static value (B columns). This is due to the adaptation of λ to each current solution. Thus IGLS finds the optimal solution in around 30% of the cases for all classes of experiments.

It seems reasonable for us to trust to the proposed IGLS since it usually succeeds to find good solutions and even the optimal solutions in some cases with CPU times that are very small comparing to the ones of the exact method Cplex.

Based on these results, the value of λ is computed dynamically for the rest of experiments.

B. STABILITY TEST

In previous section, we used the *RPD* (eq.8) to present the IGLS results. However, it could be interesting to present the dispersion too. This later is expressed according to the

standard deviation. Thus, first we compute for each instance its deviation to the average as mentioned in Equation 9. Then, we compute the standard deviation of each benchmark as mentioned in Equation 10:

$$Var_{ins} = \sqrt{(RPD_{ins} - RPD_{bench})^2} \quad (9)$$

$$Var_{bench} = \frac{\sum_{ins=1}^{10} \sqrt{(RPD_{ins} - RPD_{bench})^2}}{10} \quad (10)$$

Where RPD_{ins} is the deviation of one instance and RPD_{bench} is the deviation of one benchmark (the average of RPD_{ins}). Table II shows the obtained results.

Bench	SAI		LAI	
	LC	HC	LC	HC
9	8.56	6.80	5.91	7.33
10	8.23	5.58	5.98	5.66
11	4.90	5.29	5.14	7.32
12	7.67	4.88	3.70	5.73
13	8.65	5.08	4.41	4.54

TABLE II: Standard deviation results.

We remark from Table II that the standard deviations do not exceed 9%. Moreover, all instances' deviations are in [0.01, 21.44] for all cases. Based on these results, one could remark that IGLS is relatively stable on the considered benchmarks.

V. CONCLUSION AND FUTURE WORK

In this paper, we treated the single machine problem with flexible and periodic maintenance activities. We assume that each maintenance activity must be treated by a human resource characterized by a competence level and a timetabling precisizing their availabilities in the workshop. To solve the problem we proposed an adapted guided local search characterized by the application of an intensifying technique. The experimental results show that IGLS gives good solutions in reasonable CPU times. In future work, we aim to introduce the learning effect to the human resources allowing them to improve their competence level. Also, one could consider the studied problem as a multi-objective optimization one.

ACKNOWLEDGMENT

This research is supported by the project PHC-Tassili 17MDU987.

REFERENCES

- [Alsheddy and Tsang2011] A. Alsheddy and E-P. Tsang. Empowerment scheduling for a field workforce. *Journal of Scheduling*, 14:639–654, 2011.
- [Chen2008] J-S. Chen. Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of the Operational Research*, 190:90–102, 2008.
- [Ciro et al.2015] G-C. Ciro, F. Dugardin, F. Yalaoui, and R. Kelly. Open shop scheduling problem with a multi-skills resource constraint: a genetic algorithm and an ant colony optimisation approach. *International Journal of Production Research*, page DOI: 10.1080/00207543.2015.1126371, 2015.
- [Du and Leung1990] J. Du and J-Y-T. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of Operations Research*, 15(3):483–495, 1990.
- [Graham et al.1979] R-L. Graham, E-L. Lawler, J-K. Lenstra, and A-R. Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.
- [Guyon et al.2014] O. Guyon, P. Lemaire, E. Pinson, and D. Rivreau. Solving an integrated job-shop problem with human resource constraints. *Annals of Operations Research*, 1(213):147–171, 2014.
- [Liu et al.2015] M. Liu, S. Wang, C. Chu, and F. Chu. An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research*, DOI:10.1080/00207543.2015.1108535, 2015.
- [Luo et al.2015] A. Luo, T-C-E. Cheng, and M. Ji. Single-machine scheduling with a variable maintenance activity. *Computers and Industrial Engineering*, 79:168–174, 2015.
- [Ma et al.2010] Y. Ma, C. Chu, and C. Zuo. A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58(2):199–211, 2010.
- [Mashkani and Moslehi2016] O. Mashkani and G. Moslehi. Minimising the total completion time in a single machine scheduling problem under bimodal flexible periodic availability constraints. *International Journal of Computer Integrated Manufacturing*, 29:323–341, 2016.
- [Nagata and Ono2018] Y. Nagata and I. Ono. Guided local search with iterative ejections of bottleneck operations for the job shop scheduling problem. *Computers & Operations Research*, 90:60–71, 2018.
- [Touat et al.2017a] Meriem Touat, Sabrina Bouzidi-Hassini, Fatima Benbouzid-Sitayeb, and Belaid Benhamou. A hybridization of genetic algorithms and fuzzy logic for the single-machine scheduling with flexible maintenance problem under human resource constraints. *Applied Soft Computing*, 59:556–573, 2017.
- [Touat et al.2017b] Meriem Touat, Fatima Benbouzid-Si Tayeb, Sabrina Bouzidi-Hassini, and Belaid Benhamou. A fuzzy genetic algorithm for single-machine scheduling and flexible maintenance planning integration under human resource constraints. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 551–558. IEEE, 2017.
- [Touat et al.2018] Meriem Touat, Fatima Benbouzid-Si Tayeb, and Belaid Benhamou. An effective heuristic for the single-machine scheduling problem with flexible maintenance under human resource constraints. *Procedia Computer Science*, 126:1395–1404, 2018.
- [Voudouris and Tsang1995] C. Voudouris and E. Tsang. Guided local search. *Department of Computer Science, University of Essex*, Technical Report, 1995.
- [Voudouris and Tsang1997] C. Voudouris and E. Tsang. Fast local search and guided local search and their application to british telecom's workforce scheduling problem. *Operations Research Letters*, 20(3):119–127, 1997.
- [Yazdani et al.2017] M. Yazdani, S-M. Khalili, M. Babagolzadeh, and F. Jolai. A single-machine scheduling problem with multiple unavailability constraints: A mathematical model and an enhanced variable neighborhood search approach. *Journal of Computational Design and Engineering*, 4:46–59, 2017.
- [Yurtkuran et al.2018] A. Yurtkuran, B. Yagmahan, and E. Emel. A novel artificial bee colony algorithm for the workforce scheduling and balancing problem in sub-assembly lines with limited buffers. *Applied Soft Computing*, 73:767–782, 2018.