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

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. Furthermore, the assignment of human resources to maintenance activities can be done by favoring either the efficiency, the training or the equity. To solve this NP-hard scheduling problem, we propose a hybrid metaheuristic based on the guided local search method that uses a combination of intensifying / diversifying techniques. 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, then studied the impact of the human resource assignment strategy on scheduling performance.

1 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 [Chen, 2008], [Mashkani and Moslehi, 2016], [Luo *et al.*, 2015].

Besides, 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 latter 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]). In the maintenance scheduling, the human resource competence had been the subject of the majority of research works done on this field [Safaei *et al.*, 2011],[Tang *et al.*, 2007], [Marmier *et al.*, 2009]. 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.*, 2017],[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 where each maintenance activity is treated by a human resource characterized by a competence and some availability intervals. The contributions of this paper are twofold. First, we propose an integrated Guided Local Search (hereafter IGLS) to solve the studied problem. Moreover, to balance the intensification and diversification search effort we embedded a post optimisation procedure and a restart scheme. Moreoevre, we propose three human resource assignment scenarios. The first one favors the maintenance efficiency, it is done by choosing the humans having the highest competence that leads to a short maintenance time. The second one favors the training, it is achieved by involving the humans having the lowest competence aiming to improve the human competence. The third one favors the equity, it is acomplished by selecting the humans that do not work enough. This make a balance in the workload.

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.

2 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 $J = J_1, J_2, ...J_N$ 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 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.

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, $i \in \{N+1, ..N+Nb\_Occ\}$ occurring every T^* periods and each occurrence depends on the ones preceding it on the machine. The processing time $p^{'}$ of the maintenance activity is fixed and is nonnegative. 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 planed 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}. Since we seek scheduling over a production horizon, we do not perform a maintenance operation after the processing of the last job.

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. 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, $AI_r = \{[LB_{r1}, UB_{r1}], .., [LB_{rm}, UB_{rm}]\}$. 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 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.1).

$$f_p = \sum_{i=1}^{N} T_i \tag{1}$$

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 and the assignment strategy (equity, efficiency and trainning) (eq.2).

$$f_m = \sum_{i=N+1}^{N+Nb\_Occ} (E_i + T_i) \tag{2}$$

To optimize both production and maintenance criteria, we consider the global function defined as follows (eq.3):

$$\begin{cases} f = \alpha \times f_p + \beta \times f_m \\ \alpha + \beta = 1 \end{cases} \tag{3}$$

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 by [Du and Leung, 1990].

3 The proposed integrated guided local search meta-heuristic

Guided Local Search (GLS) [Voudouris and E.Tsang, 1995a] is a penalty-based metaheuristic algorithm that sits on top of other local search algorithms. The GLS associates a set of M features with every solution S, a feature f_{if} is determined by the *indicator function* I_{if}, $if \in \{1, ..., M\}$, which is equal to 1 if S has the property f_{if} and 0 otherwise. The GLS penalizes solution features during each iteration based on the value of a utility function: $util_{if}(S) = I_{if}(S) \times \frac{c_{if}}{1+pn_{if}}$, where c_{if} and a pn_{if} are respectively the cost and the penality associated to f_{if}. The penalty acts as a disturbance to an augmented objective function: $h(S) = f(S) + \lambda \sum_{if=1}^{M} (pn_{if} \times I_{if}(S))$ which is adjusted during each iteration, where λ represents the relative weight of penalties with respect to the cost of solution. 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 E.Tsang, 1995b], [Alsheddy and Tsang, 2011], [Nagata and I.Ono, 2018]. 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 charging to realize the maintenance activities are taken into account. These human resources are characterized by availability periods and competence levels. The characteristics 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 several local search heuristics (LS) instead of one. Indeed, In each iteration of IGLS, a specific LS heuristic is applied according to the feature witch corresponds to maximum utility.
- The introduction of post-optimization process in order to minimize both production and maintenance delays (intensifing). This process is performed at the end of each local search (section 3.3) ;
- The introduction of disruption process to generate a new solution after the solution stagnation (diversifing) (section 3.3).

The main steps of the proposed integrated GLS are depicted in Algorithm 1.

Algorithm 1 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}

```
k ⟵ 0;
Generate an initial solution S_0;
Set all penalties pn_if to 0, if = 0, ..., M;
S_best ⟵ S_0 ; S_c ⟵ S_0
while (Termination criterion is not met) do
    h(S) ⟵ f(S_c) + λ Σ_{if=1}^{M} pn_i × I_if;
    S_c ⟵ Local_Search(S_c, h(S), I(S));   ▷ perform the local search heuristic with the feature f_if corresponding the maximum utility util_if and the objective function h
    Post-optimization (S_c);   ▷ perform the post optimization procedure
    pn_i ⟵ pn_i + 1;   ▷ penalize features with maximum utility
    for (Each feature if) do   ▷ update features' utilities
        util_if(S_c) = I_if(S_c) × c_if / (1 + pn_if);
    end for
    if (f(S_c) < f(S_best)) then
        S_best ⟵ S_c;
    end if
    if (Stagnation) then   ▷ restart schema after stagnation
        Disruption (S_c);
    end if
end while
Return S_best;
```

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

3.1 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 assosiate to the integrated sequence an assignment matrix **Z** with size $R \times m$. Such **R** is the number of human rresources and **m** is the number of availability intervals. $Z[r, l]$ 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 when the human resource availability intervals and the tolerance period do not overlap. Then, the production jobs are scheduled based on to the maintenance activity planning according to their due dates.

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

$$\begin{array}{cc} & \begin{array}{ccc} AI_{r1} & AI_{r2} & AI_{r3} \end{array} \\ \begin{array}{c} HR_1 \\ HR_2 \end{array} & \begin{bmatrix} 0 & 8 & 0 \\ 7 & 9 & 0 \end{bmatrix} \end{array}$$

Figure 1: An example of a solution encoding

3.2 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 other ones to maintenance activities. Each feature in F_1 (respectively F_3) represents the delay of a production (respectively maintenance) activity T_i. Thus, F_1 has N features and F_3 has Nb_Occ features, the total number of production (respectively maintenance) activities. The second feature F_2 represents the advance of maintenance activities. There are Nb_Occ features in F_2. Consequently, to every solution S, it is associated $(2 \times N + Nb\_Occ)$ features. Each feature $F_{if,i}$ is associated with an indicator $I_{if,i}(S)$, a cost $c_{if,i}$ and a utility value $util_{if,i}$ such as: $i = 1..N + Nb\_Occ$.

Based on each feature, a specific local search (LS) heuristic is applied. Therefore, three LS heuristics are then proposed. 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 ramdomly an appropriated 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 2).

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 3).

Algorithm 2 The pseudo code of LS_1.

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

Let t_i^* be the desired start time of J_i $(d_i - p_i)$;
Let $i^{'}$ be the identifier of activity scheduled at t^*;
if ($J_i^{'}$ is a production job) **then**
  if (all activities scheduled in $[t^*, d_i]$ are production jobs) **then**
    Schedule J_i at t^*;
  else
    Let t_j be the stating time of maintenance activity scheduled at $[t^*, d_i]$;
    Schedule J_i at $[(t_j - p_i, t_j]$;
  end if
else
  if (J_i can be scheduled before $M_i^{'}$) **then**
    Schedule J_i at $[t_i^{'} - p_i, t_i^{'}]$;
  else
    Schedule J_i at $[t_i^{'}, t_i^{'} + p_i]$;
  end if
end if
$S_v \longleftarrow$ schedule of the production jobs in the same sequence that S_c;
if $(h(S_v) < h(S_c))$ **then**
  $S_c \longleftarrow S_v$;
end if
Return S_c;

3.3 Intensification and diversification strategies

It consists in reducing the idle time of the machine by advancing the production jobs, looking for the production objective function improvement. Then, 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 4.

Besides, we found that the penalties are not sufficient to overcome the constraints imposed by human resources in order to explore the entire research space. Hence, if after several iterations $Nb_{Improve}$, the objective function is not improved, we disrupt the scheduling in order to explore new solutions. It consists in rescheduling a randomly selected job at the end of the sequence which allows finding new schedules, what means of new search regions in the total search space.

3.4 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.

Algorithm 3 The pseudo code of LS_2 (respectively LS_3).

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

Let $AI_{rl} = [LB_{rl}, UB_{rl}]$ be the availability interval in which M_i is planned and t_i its starting time;
if $(t_i > LB_{rl})$ (respectively $c_i < UB_{rl}$) **then**
  $t_i \longleftarrow LB_{rl}$ (respectively $c_i \longleftarrow UB_{rl}$);
  $S_v \longleftarrow S$ with the new maintenance planning;
else
  Let $AI_{r^{'}l^{'}}$ be the unoccupied availability interval coming just before (respectively after) AI_{rl} and that does not disruped the planning of M_{i-1} (respectively M_{i+1});
  Schedule M_i at $AI_{r^{'}l^{'}}$ such as: $t_i = UB_{r^{'}l^{'}} - p_i$ (respectively $t_i = LB_{r^{'}l^{'}}$) and $c_i = UB_{r^{'}l^{'}}$ (respectively $c_i = LB_{r^{'}l^{'}} + p_i$);
  $S_v \longleftarrow S_c$ with the new maintenance planning at $AI_{r^{'}l^{'}}$;
end if
Update the tolerance intervals TI_{i+1} for maintenance activities $i + 1 ... N + Nb\_Occ$;
Update the human resource availabilities;
$S_v \longleftarrow S_c$ with production job Scheduling in same order that S_c;
if $(h(S_v) < h(S_c))$ **then**
  $S_c \longleftarrow S_v$;
end if
Return S_c;

4 Experimental results

In this section, we present some experimental results to: firstly, show the effectiveness of the proposed IGLS in solving the integrated single-machine scheduling problem with flexible maintenance under the human resource constraints. In the second time, we will study the impact of the human resource assignment strategy on the obtained results. 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.

4.1 IGLS validation

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. 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 $(Comp_1, Comp_2)$. Both $Comp_1$ and $Comp_2$ are distributed uniformly in $]0, 2[$ $(U]0, 2[)$. 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 $ph_2 = p^{'} \div Comp_2$. The set AI_{rl} of availability intervals with $l \in \{1, 2\}$ is generated according to Algorithm presented in [Touat *et al.*, 2018]. We propose two types of availability inter-

Algorithm 4 The pseudo code of the post optimization procedure.

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

for (Each maintenance occurrence M_i, $i = N+1...N+Nb\_Occ$) **do**
  if (There is an idle period id between M_i and the previous production job J_j) **then**
    if (There is a production job $J_{i'}$ coming after M_i and having a processing times $p_{i'} < id$) **then**
      Advance $J_i^{'}$;
    else
      if (It is possible to delaying M_i in its AI_{rl} generating an idle time id) **then**
        if (There is a production job $J_i^{'}$ coming after M_i and having a processing times $p_{i'} < id$) **then**
          Delay M_i and Advance $J_{i'}$;
          Update the tolerance intervals TI_{i+1};
        end if
      end if
    end if
  else
    if (It is possible to delaying M_i in its AI_{rl} generating an idle time id) **then**
      if (There is a production job $J_i^{'}$ coming after M_i and having a processing times $p_i^{'} < id$) **then**
        Delay M_i and Advance $J_i^{'}$;
        Update the tolerance intervals TI_{i+1};
      end if
    end if
  end if
end for
$S \longleftarrow S_c$ with the new maintenance planning and production scheduling;
if $(f(S) < f(S_c))$ **then**
  $S_c \longleftarrow S$;
end if
Return S_c;

vals 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.

For the rest of experiments, the values of the control parameters α and β of equation 3 are set to 0.5. The proposed common weighted global objective function will allow tackling the studied problem in a simplified way. In addition, the values of the control parameters of the proposed IGLS are fixed on the base of different trials of the proposed approach. The complete details are not reported for the sake of concise presentation. Their values are the following:

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

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.4).

$$RPD = \frac{(f - f_{CP})}{f_{CP}} \times 100 \qquad (4)$$

where f is the global objective function defined in Equation 3 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. 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": $\lambda = \frac{f(S_{best})}{f(S_c)}$. Table 1 shows the obtained results. For each value of λ (static and dynamique (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				*HC*			
		f_{CP}	f	*RPD*	*B*	f_{CP}	f	*RPD*	*B*
	9	423,95	454,95	7,31%		134,95	143	5,97%	
	10	404,55	450,55	11,33%		147,55	156,35	5,96%	
0.9	11	441,25	465,55	5,51%	10	174,7	187,2	7,16%	12
	12	507,1	545,8	7,63%		216	237,75	10,07%	
	13	418,25	456,75	8,21%		183,25	195,7	6,79%	
	9	423,95	459,75	8,44%		134,95	144,95	7,41%	
	10	404,55	449,65	11,15%		147,55	157,65	6,85%	
D	11	441,25	461,85	4,67%	14	174,7	189,9	8,7%	12
	12	507,1	544,55	7,39%		216	237,25	9,84%	
	13	418,25	456,2	9,07%		183,25	200,4	9,36%	
λ	*N*	*LAI*							
		LC				*HC*			
		f_{CP}	f	*RPD*	*B*	f_{CP}	f	*RPD*	*B*
	9	304,6	330,45	8,49%		106,8	120,35	12,69%	
	10	284,65	311,65	9,45%		121,65	133,65	9,86%	
0.9	11	356,3	396	11,14%	5	147,4	169,24	14,82%	8
	12	368,15	409,85	11,33%		176,55	198,35	12,35%	
	13	349,45	385,15	9,42%		172	185,8	8,02%	
	9	304,6	330	8,44%		106,8	117,95	10,44%	
	10	284,65	310	8,91%		121,65	136,25	12%	
D	11	356,3	410,1	15,1%	4	147,4	166,85	13,2%	8
	12	368,15	413,7	12,37%		176,55	196,85	11,5%	
	13	349,45	376,6	7,77%		172	179,85	4,56%	

Table 1: IGLS results on small size instance.

The obtained results show that for most of the tested instances, the RPD does not exceed 11% for all classes of experiments, except 20% of cases when λ is dynamic and 30% 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 LAI/HC class. It is even lower than 6% (5.51%) 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 classe of experiments.

It seems reasonable for as 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.

A stability test

In previous section, we used the ***RDP*** (eq.4) 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 mentionned in Equation 5. Then, we compute the standard deviation of each benchmark as mentionned in Equation 6:

$$Var_{ins} = \sqrt{(RPD_{ins} - RPD_{bench})^2} \tag{5}$$

$$Var_{bench} = \frac{\sum_{ins=1}^{10} \sqrt{(RPD_{ins} - RPD_{bench})^2}}{10} \tag{6}$$

Where RPD_{ins} is the deviation of one instance and RPD_{bench} is the deviation of one benchmark (the average of RPD_{ins}). Table 2 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 2: Standard deviation results.

We remark from Table 2 that the standard deviations do not exceed 9%. Moreover, all istances' deviations are in [0.01, 21.44] for all cases. Based on these results, one could remark that IGLS is relativelly stable on the considered benchmarks.

4.2 The impact of human resource assignment strategy on results

In this class, we will test the different strategies of human resource assignment to mesure their impact on the obtained results. Thus, when more than one resource is available at the same time, the assignment of human resource to maintenance activity is also based on the adopted strategy in maintenance service (efficiency, training and equity).

We generate production and maintenance data as instance size ranging from 20 to 700 production jobs. However, we generate a new instances for the human resource ones. We consider 4, 6 and 8 resources in the workshop with the competence generated ramdomly in]0,2[. The availability intarvals have a medium size and are generated accorrding the algorithm presented in [Touat *et al.*, 2018]. Table 3 shows results. For each benchmark, we present the global objective function average of ten instances according to the chosen strategy. We denote the objective functions by $f_{efficiency}$, $f_{trainning}$, f_{equity}. We remark from Table 3 that the strategy based on efficiency always gives the best results, contrary to the strategy based on training which gives the worst results for almost all benchmarks. Thus, the strategy based on efficiency is better than other ones since it involvs the human resources having a high competence level. Consequently, the operating time of maintenance activities is reduced and their tardiness too (eq. 2). Thus, the tardiness of all production jobs scheduled after the maintenance activity in question is reduced.

Human resources	Bench	$f_{efficiency}$	$f_{trainning}$	f_{equity}
4	20	1175,9	2134,7	1788,6
	40	4747,3	9799,8	7959,9
	60	7129,3	12451,4	24303,44
	80	15861,6	27957,8	31442,6
	100	18602,7	37778,5	36632,5
	140	27278,9	106814,2	65288,5
	200	52575,5	113510,3	109250,6
	300	90283,2	239302,7	183922,6
	500	215061,3	418284,9	412511,1
	700	395487,9	599921,2	618758,3
6	20	1062,7	2022	1899,9
	40	4990,4	8460,9	8123
	60	6726,7	20472,4	13047,5
	80	15088,9	24797	27419,5
	100	17990,6	54512,5	35707,3
	140	33717,6	70575,7	12982,4
	200	60336,7	115246,4	109104,1
	300	114116	256655,3	197197
	500	221788,4	356243,4	403858,2
	700	427389,5	1195366,5	642475,3
8	20	1096,5	2826,6	2101,8
	40	4810,8	8409,4	7848,2
	60	7105	26112,5	13454,3
	80	14182	26519,4	26723,3
	100	16436,3	35948,6	34374,3
	140	29177,4	68288,8	56756
	200	53391,8	141778,3	96346,9
	300	101257,5	235978,2	187679,5
	500	186284,9	374318	561955,9
	700	354529,4	704401,2	644433,6

Table 3: IGLS results according to the human resource assignment strategy.

5 Conclusion and future work

In this paper, we treated the single machine problem with flexible and periodic maintenance activities. We assume that each maintenance activitie must be treated by a human resource characterized by a competence level and a timetabling precising their availabilities in the workshop. In addition, the assignment of human resources is subjest to some rules related the workshop assignment strategy which could favor: the efficiency, when human resources with high competences are chosen. The equity, when all resources are assigned respecting the same workload. The training, when human resources having low competences are favored. To solve the problem we proposed an adapted guided local search characterized by the application of an intensifying / diversifying techniques. The experimental results show that IGLS gives good solutions in reasonable CPU times. We also studied the impact of the humain assignment strategy on the obtained results. In future work, we aim to introduce the learning effect to the human resources allowing them to improve their cometence level. Also, one could consider the studied problem as a multi-objective optimization one.

References

[Alsheddy and Tsang, 2011] A. Alsheddy and E-P. Tsang. Empowerment scheduling for a field workforce. *Journal of Scheduling*, 14:639–654, 2011.

[Chen, 2008] 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 Leung, 1990] 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.

[Marmier *et al.*, 2009] F. Marmier, C. Varnier, and N. Zerhouni. Proactive, dynamic and multi-criteria scheduling of maintenance activities. *International Journal of Production Research*, 47:2185–2201, 2009.

[Mashkani and Moslehi, 2016] 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 I.Ono, 2018] 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.

[Safaei *et al.*, 2011] N. Safaei, D. Banjevic, and A-K-S. Jardine. Workforce-constrained maintenance scheduling for military aircraft fleet: a case study. *Ann Oper Res*, 186:295–316, 2011.

[Tang *et al.*, 2007] H. Tang, E. Miller-Hooks, and R. Tomastik. Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research*, 43:591–609, 2007.

[Touat *et al.*, 2017] M. Touat, S. Bouzidi-Hassini, F. Benbouzid-Sitayeb, and B. 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.*, 2018] M. Touat, F. Benbouzid-Si Tayeb, and B. 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 E.Tsang, 1995a] C. Voudouris and E.Tsang. Fast local search and guided local search and their application to british telecom's workforce scheduling problem. *Department of Computer Science, University of Essex*, Technical Report:CSM–246, 1995.

[Voudouris and E.Tsang, 1995b] C. Voudouris and E.Tsang. Guided local search. *Department of Computer Science, University of Essex*, Technical Report:CSM–247, 1995.

[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 subassembly lines with limited buffers. *Applied Soft Computing*, 73:767–782, 2018.