



HAL
open science

Exact and Metaheuristic Approaches for the Single-machine Scheduling Problem with Flexible Maintenance under Human Resource Constraints

Meriem Touat, Fatima Benbouzid, Belaid Benhamou

► **To cite this version:**

Meriem Touat, Fatima Benbouzid, Belaid Benhamou. Exact and Metaheuristic Approaches for the Single-machine Scheduling Problem with Flexible Maintenance under Human Resource Constraints. International Journal of Manufacturing Research, In press. hal-03167816

HAL Id: hal-03167816

<https://hal.science/hal-03167816>

Submitted on 18 Mar 2021

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.

Exact and Metaheuristic Approaches for the Single-machine Scheduling Problem with Flexible Maintenance under Human Resource Constraints

Meriem Touat*

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

E-mail: m_touat@esi.dz

Ecole Supérieure en Sciences Appliquées Alger (ESSA-Alger),
BP 474, Place des Martyrs, Alger 16001, 1er Novembre, Algérie

E-mail: m.touat@g.essa-alger.dz

*Corresponding author

Fatima Benbouzid-Si Tayeb

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

E-mail: f_sitayeb@esi.dz

Belaid Benhamou

Aix-Marseille Universié,
Laboratoire LSIS,
Domaine Universitaire de Saint-Jérôme, Batiment Polytech, Avenue
Escadrille Normandie-Niemen,
13397 MARSEILLE CEDEX 20, France
E-mail: belaid.benhamou@univ-amu.fr

Abstract: This paper tackles the scheduling problem of both production and flexible preventive maintenance activities on a single machine under human resource constraints. The considered human resources oversee the maintenance activities. They are characterized by a competence level and a timetabling that determines their availabilities. Our objective is to minimize a common and weighted objective function that involves both the tardiness and the earliness resulting from production and maintenance activities when considering maintenance workers. We first introduce a mathematical modeling for the studied problem that we implemented in Cplex in order to compute the optimal solutions of small instances of this problem. Secondly, we propose an improved Guided Local Search (GLS) metaheuristic to deal with relatively large instances of the problem. Cplex is used as a reference exact method to check the solution quality of the approached method GLS. The proposed methods are evaluated on a large number of randomly generated instances. The experimental results show that the studied problem is very hard to solve optimally, the approached method GLS performs well and is able to find good solutions to instances up to 700 jobs in a reasonable CPU time.

Keywords: Single machine scheduling; Flexible maintenance planning; Constraints programming; Guided Local Search; Human resource constraints.

Reference to this paper should be made as follows: Touat, M., Benbouzid-Si Tayeb, F. and Benhamou, B. (2020) 'Exact and Metaheuristic Approaches for the Single-machine Scheduling Problem with Flexible Maintenance under Human Resource Constraints', *Int. J. Manufacturing Research*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Meriem Touat received her Magister in Information, Communication Sciences and Technologies from the Ecole nationale Supérieure d'Informatique (ESI) of Algiers (Algeria), in 2014, and her PhD from ESI, in 2020. Her research interests include scheduling problems in manufacturing systems.

Fatima Benbouzid-Si Tayeb is a Professor at Ecole nationale Supérieure d'Informatique (ESI) d'Alger, Algeria. She received her MEng from ESI in 1992, and her PhD from Franche Comté University, France, in 2005. She is also a Researcher at the Laboratoire des Methodes de Conception de Systèmes (LMCS) of ESI. Her research interests are AI methods for scheduling, social networks and diagnostic approaches.

Belaid Benhamou is currently a research professor at Aix-Marseille University. He received his mathematical bachelor in 1983, his engineering degree in computer science in 1988 and his PHD in computer science in 1993 at the Aix-Marseille I University. He obtained an assistant professor job at the University of Aix-Marseille I in 1995 and in 2001 he passed his habilitation after which he obtained a national qualification for a full university professor. His research activities are in the field of artificial intelligence: constraint solving, logic and automated theorem proving, logic programming, non-monotonic reasoning, meta-heuristics and combinatorial optimization, and data mining. He published over 120 articles and has supervised several doctoral thesis on these areas. He organized several conferences and participated to over 35 program committees of conferences, 25 thesis committees, and several research projects.

1 Introduction

In the last years, there had been a considerable interest in models trying to integrate production and preventive maintenance in machine scheduling in order to take into account their mutual interdependencies. Whereas in the past, these aspects have been treated as separate problems. Production planning, typically concerns the determination of lot sizes and the evaluation of the capacity needs, in case of fluctuating demands. Both the optimal lot size and the capacity needs are affected by failures. On the other hand, maintenance prevents breakdowns and improves quality. Accordingly, they should be planned in an integrated way. There is two ways to integrate the preventive maintenance activities in the production scheduling. The first one considers scheduling with periodic maintenance. It is a deterministic approach where the maintenance periods are fixed in advance. The other alternative represents scheduling with flexible maintenance where the maintenance periods are decision variables (the intervals are not fixed).

Single machine problems are of fundamental character. They can be interpreted as building blocks for more complex problems. In the literature, several variants of the single-

machine problem with preventive maintenance had been investigated, depending on the used maintenance activities integration method. Adiri et al. (1989) were the first to introduce the single-machine problem with one deterministic unavailability period. They proved that the problem is NP-complete and that the sequence based on Shortest Processing Time (SPT) has a relative error bound of $\frac{1}{4}$. Based on these results, a simpler proof of NP-completeness of the problem had been established by Lee and Liman (1992). Besides, Liao and Chen (2003) studied for the first time the problem with periodic maintenance to minimize the maximum lateness. A branch and bound algorithm and a heuristic method were proposed to solve the problem. After that, surveys were presented to synthesize works before 2010 on scheduling with deterministic unavailability periods (see Schmidt (2000) and Ma et al. (2010)). Cui and Lu (2014) consider the problem with periodic maintenance and production jobs with release dates to minimize the makespan in both resumable and non resumable case. They proposed a mathematical model, a specific heuristic and a branch and bound algorithm. The same problem had been studied by Lee and Kim (2012) and Liu et al. (2015) to minimize the number of tardy jobs. A mixed integer programming formulation and a two-phase heuristic algorithm are proposed by Lee and Kim (2012). On the other hand, Liu et al. (2015) proved that the problem is NP-hard in the strong sense, then proposed a lower bound for the problem and a branch and bound algorithm. Recently, Yazdani et al. (2017) considered the problem with periodic maintenance to minimize the sum of maximum earliness and tardiness of jobs. They proposed a mathematical model and a variable neighborhood search to deal with large instances of the problem.

Yang et al. (2002) introduced the scheduling problem with flexible maintenance. They considered the problem with one flexible maintenance that must be done in previously arranged period with objective to minimize the makespan. The authors proved the NP-hardness of the problem and proposed a heuristic to solve it in practice. Yang et al. (2011) studied the single machine scheduling problem with a flexible maintenance activity with the objective to minimize the makespan. Both resumable and non-resumable cases are studied. To find the optimal solutions, a dynamic programming algorithm and a branch-and-bound method were proposed. Two mathematical models and a heuristic to solve the scheduling problem with flexible and periodic maintenance are proposed by Chen (2008). The objective, is to minimize the makespan. The same problem had been studied by Low et al. (2010) with the exception that the machine should stop to maintain after a periodic time interval or to change tools after a fixed amount of jobs processed. Authors proposed four efficient heuristics. Sbihi and Varnier (2008) proposed a branch and bound algorithm and a heuristic to solve this problem in non-resumable case with the objective to minimize the maximum tardiness. Besides, other kinds of scheduling with flexible maintenance are proposed. Mashkani and Moslehi (2016) proposed a bimodal periodic availability constraints where a machine must be maintained after working continuously for two possible periods of time. The maintenance duration is determined according to these periods. A generalization to several periods is made. The objective is to minimize the makespan. A mathematical model is given, then a heuristic and branch and bound algorithm were proposed. Luo et al. (2015) consider flexible maintenance that must start before a given deadline. The maintenance duration increases with its starting time. Authors proposed polynomial time optimal heuristics to minimize several classical objective functions.

The majority of the research works on scheduling with preventive maintenance assume a perfect environment in terms of resource availability. However, both production and maintenance services require agents to ensure the execution of activities. Generally, the role of production agents is limited to the control of production process. That is, they provide

a supervisory mission. On the other hand, the maintenance agent's mission is to ensure the proper functioning of machines. They should be versatile and be able to accomplish several interventions such as: installing, regularizing, controlling, maintaining and repairing the industry's production tools. This level of versatility is not always guaranteed. Indeed, maintenance agents 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 is then, necessary to consider the human resources in scheduling problems or even more as a scheduling problem separately. These problems had been widely studied in the literature. Some authors address only the human resource assignment problem and refer to it as the workforce scheduling problem or the employee timetabling problem (Yurtkuran et al. (2018), Gérard et al. (2016)). In (Lilly et al. (2007), Safaei et al. (2011), Tang et al. (2007)), authors consider the timetabling problem in the maintenance context. However, in (Sabar et al. (2009a), Sabar et al. (2009b), Yurtkuran et al. (2018)), authors treat this problem in the production one.

On the other hand, others researches deal with the dual resource scheduling problem in which the integrated scheduling of both tasks and human resource is considered. Ciro et al. (2015) studied the open shop scheduling problem where the objective is to minimize the total flow time. A task require a human resource with a specific skill to be executed. Indeed, workers are assigned to a task according to their skills and mastered abilities. The authors proposed a mathematical model and a genetic algorithm that considers different resource constraints related to personnel staff competences and their availability to execute a task. The integrated employee-timetable and job-shop-scheduling problem are studied by Guyon et al. (2014). It is a question of assigning employees to machines and shifts with the objective to minimize the cost assignment of operators to both machines and shifts. The employees have different competences. Authors proposed a mathematical model and an exact method, that is, an hybridization of a cut generation process with a branch and bound strategy. In the maintenance scheduling, the human resource competence had been the subject of the majority of research works done on this subject (Marmier et al. (2009), Bennour and Crestani (2007)). To our knowledge, research on the integrated production and maintenance scheduling problems under human resource constraints has been introduced recently (Bouzidi-Hassini et al. (2015), Touat et al. (2017a), Touat et al. (2017b), Touat et al. (2018), Touat et al. (2019)) and research works on it still very scarce.

In this paper, we deal with the single machine scheduling problem and flexible maintenance planning. Each maintenance activity should be carried out after an optimistic date and should not exceed a pessimistic date. It must be assigned to a human resource characterized by a competence and some availability intervals. The assignment is performed according to several strategies. Indeed, when more than one human resource is available, the one with the lowest competence is chosen when we aim to train the maintenance staff. The one with the highest competence is chosen, when we aim to minimize the processing time. Finally, if we aim to balance the workload, then the human resource who does not work sufficiently is chosen. However, if no strategy is adopted to assign the human resources, the first available resource is chosen. The contributions of this paper are twofold. First, we propose a mathematical modeling for this new scheduling problem based on the integer programming (IP) paradigm. We expressed the problem as a set of both linear and logical constraints that are implemented in ILOG OPL language. Cplex is then used as an exact method to find the best integrated schedule that optimizes both production and maintenance criteria for relatively small instances of the problem. On the other hand, we developed a

Guided Local Search (GLS) metaheuristic to solve large instances of the studied problem. The proposed integrated algorithm is supplemented with several local search heuristics based on specific knowledge of production and maintenance activities instead of only one in the original GLS. Moreover, it integrates a post-optimization procedure and a restart scheme aiming a more effective intensification and diversification of the search process. Cplex is used to show the hardness of solving the problem in practice and serves as a reference method to evaluate the proposed metaheuristic solution quality and its efficiency.

The remainder of this paper is organized as follows: The problem description and the IP modeling are given in Section 2. The proposed GLS is described in Section 3. Section 4 is devoted to the computational experiments. Section 5 concludes the work and gives some research perspectives.

2 Problem definition and modeling

In this section, we first describe the integrated single machine scheduling problem with flexible maintenance we are dealing with. Next both the human resource characteristics and the most salient aspects of the objective functions to optimize are defined. Finally, the IP formulation of the studied integrated problem is given.

2.1 The studied integrated single machine scheduling problem

We consider here, a single-machine scheduling problem subject to flexible and periodic maintenance. The problem consists in a set $J = J_1, J_2, \dots, J_N$ of N jobs to be processed on a single machine. All the jobs are available at time zero and 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.

Contrary to the classical single machine scheduling problem, here the machine is not continuously available due to preventive maintenance. Maintenance must be undertaken in order to ensure a high availability of the machine. In this paper, we consider a single flexible maintenance task M with multiple occurrences. Each occurrence $M_i, i \in \{N + 1, \dots, N + Nb_Occ\}$ depends on the tasks preceding it on the machine and must be preceded every T^* . The processing time p' of the maintenance activity is non negative and evaluated depending on the competence of the human resource assigned to it. A maintenance M_i must be completed within a time window $TI_i = [Tmin_i, Tmax_i]$ representing its tolerance interval (Fig. 1). 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) (Fig.1). Earliness and tardiness of the maintenance occurrence M_i are computed according to Eq. 1.

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

Where t'_i and c'_i are respectively the execution and the completion times of M_i .

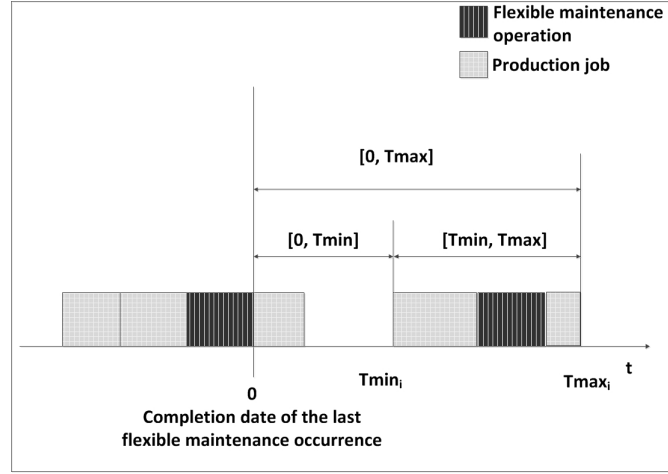


Figure 1: The flexible maintenance activity tolerance interval.

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} . The tolerance interval of the maintenance occurrence M_i is given by Eq. 2.

$$\begin{cases} Tmin_i = c'_{i-1} + T^* & , i = N + 1..N + Nb_Occ \\ Tmax_i = Tmin_i + (Tmax_{i-1} - Tmin_{i-1}) & , i = N + 1..N + Nb_Occ \end{cases} \quad (2)$$

The machine can handle at most one activity (a production job or a maintenance activity) at a time and cannot stand idle until the last job is finished. Moreover, since we seek scheduling over a production horizon, we do not perform a maintenance operation after the processing of the last job.

2.2 The human resource characteristics

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 such as: $ph_r = p' \div Comp_r$. Moreover, each resource HR_r has a timetabling which determines its availability. This is expressed by specifying for each resource HR_r a set $AI_r = \{AI_{r,l} : l = 1..m\}$ of m availability intervals (AI). More precisely, $AI_r = \{[LB_{r,1}, UB_{r,1}], \dots, [LB_{r,m}, UB_{r,m}]\}$. The symbols $LB_{r,l}$ and $UB_{r,l}$ denote respectively, the lower and the upper bounds of the availability interval $AI_{r,l}$ ($l = 1..m$). In our problem, we define the efficiency according to the executing times of maintenance activities ph_r . In addition, for each resource HR_r , we can select its total duration of work $durate_r$.

2.3 The objective functions

The problem we tackle consists in determining jointly the best sequencing of jobs and flexible maintenance occurrences to be processed by a single machine in order to optimize both production and maintenance criteria. The production objective f_p consists in searching

a permutation of N production jobs that minimizes the sum of tardiness T_i , when taking into account the maintenance activities. The production objective function f_p is computed as follows (Eq. 3):

$$\begin{cases} f_p = \sum_{i=1}^N T_i \\ T_i = \max(0, c_i - d_i), i = 1..N \end{cases} \quad (3)$$

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 representing the human availability and his competence level. This is done according to three strategies:

1. The efficiency strategy which favors the resource with the highest competence;
2. The equity strategy (workload balancing) which balances the hourly load between workers independently of their competence level;
3. The training strategy which favors the resource with the lowest competence in order to increase its qualification.

The objective f_m is given by the following equation (Eq. 4):

$$\begin{cases} f_m = \sum_{i=N+1}^{N+Nb_Occ} (E'_i + T'_i) \\ E'_i = \max(0, Tmin_i - t'_i) \quad i = N + 1..N + Nb_Occ \\ T'_i = \max(0, c'_i - Tmax_i) \quad i = N + 1..N + Nb_Occ \end{cases} \quad (4)$$

To optimize both production and maintenance criteria, we consider the global function f defined as follows (Eq. 5):

$$\begin{cases} f = \alpha \times f_p + \beta \times f_m \\ \alpha + \beta = 1 \end{cases} \quad (5)$$

Based on the classification given in Graham et al. (1979), we propose an extension that includes the human resource constraints. We add a subfield β_2 to the field β to represent the maintenance constraints as follows:

- β_1 concerns the production constraints, in our case, the only constraint is the due dates d_i .
- β_2 concerns the maintenance constraints, it is subdivided into two sub-constraints:
 - $\beta_{2,1}$ concerns the basic constraints of maintenance activities, especially, the number of maintenance activities to insert in the schedule. We note it by $\beta_{2,1,1} = \{M, M \leq k_0\}$. $\beta_{2,1,1} = M$ means that the number of maintenance activities is unknown. $\beta_{2,1,1} \leq k_0$ means that the number of maintenance activities could not exceed k_0 . The second subfield $\beta_{2,1,2}$ presents the temporal constraints, in our case, the tolerance interval $TI_i = [Tmin_i, Tmax_i]$.

- $\beta_{2,2}$ concerns the maintenance constraints related to human resource assignment, especially, the processing time of maintenance activity assigned to human resource. We denote it by $\beta_{2,2,1} = \{ph_i = g(i, r)\}$. It means that the processing time of the maintenance M_i is computed by a specific function which varies according to parameters i and r (the activity occurrence and the assigned human resource). The second sub-constraint $\beta_{2,2,2}$ represents the temporal constraints, in our case, the availability intervals $AI_r = \{[LB_{r1}, UB_{r1}], \dots, [LB_{rm}, UB_{rm}]\}$.

One could then denote the considered problem by $1/d_i, M, TI_i, g(i, r), AI_r/\alpha \sum T_i + \beta \sum (E_i + T_i)$. This problem is at least NP-hard since the basic one without maintenance activities $1/d_i/\sum T_i$ is NP-hard Du and Leung (1990).

2.4 The IP formulation of the integrated scheduling problem

In the following, we introduce an integer linear programming (IP) modeling to express the mathematical formulation proposed in Touat et al. (2018). The notations that will be used in the IP modeling are the following:

The problem parameters

- N : The number of jobs requiring processing at time zero.
- Nb_Occ : The number of maintenance activities computed in advance according to the production horizon.
- p_i : The processing time of the production job $J_i, i = 1..N$.
- d_i : The due date of the production job $J_i, i = 1..N$.
- p' : The basic processing time of the maintenance activity M .
- $TI_i = [Tmin, Tmax]_i$: The tolerance interval of the maintenance activity $M_i, i = 1 + N..N + Nb_Occ$.
- T^* : The optimal periodicity of the maintenance activity M .
- $Comp_r$: The competence of the human resource HR_r to execute $M, r = 1..R$.
- $AI_{rl} = [LB, UB]_{rl}$: The availability interval l of the human resource HR_r where LB is the Lower bound of the interval and UB its upper bounds, $r = 1..R, l = 1..m$.
- L : A very large number.

The decision variables

- i : The index of both production and maintenance jobs, $i = 1..N + Nb_Occ$.
- r : The index of human resources, $r = 1..R$.
- l : The index of human resource availability intervals, $l = 1..m$.
- J_i : The i^{th} production job, $i = 1..N$.
- M_i : The i^{th} maintenance activity occurrence, $i = 1 + N..N + Nb_Occ$.
- HR_r : The r^{th} human resource, $r = 1..R$.
- AI_{rl} : The l^{th} availability interval of the resource $r, r = 1..R, l = 1..m$.
- $EI_i = [t, c]_i$: The execution interval of the production job J_i or the maintenance activity $M_i, i = 1..N + Nb_Occ$.
- t_i : The starting time of the production job $J_i, i = 1..N$.
- c_i : The completion time of the production job $J_i, i = 1..N$.
- t'_i : The starting time of the maintenance activity $M_i, i = 1 + N..N + Nb_Occ$.
- c'_i : The completion time of the maintenance activity $M_i, i = 1 + N..N + Nb_Occ$.
- $TI_{N+1} = [Tmin, Tmax]_{N+1}$: The tolerance interval of the maintenance activity M_{N+1} .
- T_i : The tardiness of the production job $J_i, i = 1..N$.

T'_i : The tardiness of the maintenance activity M_i , $i = 1 + N..N + Nb_Occ$.
 E'_i : The earliness of the maintenance activity M_i , $i = 1 + N..N + Nb_Occ$, $i = 1 + N..N + Nb_Occ$.
 ph_i : The real processing time (according to the assigned human resource) of the maintenance activity occurrence M_i , $i = 1 + N..N + Nb_Occ$.
 x_{irl} : Is a binary variable that is equal to 1 when the availability interval AI_{rl} of the human resource HR_r is assigned to the maintenance activity occurrence M_i ; it is equal to 0 otherwise, $i = 1 + N..N + Nb_Occ$, $r = 1..R$, $l = 1..m$.
 $y_{ii'}$: Is a binary variable that is set to 1 when J_i precedes $J_{i'}$ (not necessarily immediately) and set to 0 otherwise, $1 \leq i \leq i' \leq N + Nb_Occ$.

The linear IP modeling that we propose for the considered problem is the following:
 Minimize:

$$f_{IP} = \alpha \times \sum_{i=1}^N (T_i) + \beta \times \sum_{i=N+1}^{N+Nb_Occ} (E'_i + T'_i) \quad (6)$$

Subject to:

$$T_i \geq c_i - d_i, i = 1..N \quad (7)$$

$$E'_i \geq Tmin_i - t'_i, i = N + 1..N + Nb_Occ \quad (8)$$

$$T'_i \geq c'_i - Tmax_i, i = N + 1..N + Nb_Occ \quad (9)$$

$$c_i = t_i + p_i, i = 1..N \quad (10)$$

$$(x_{irl} = 1) \Rightarrow (ph_i = p' \div Comp_r), i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (11)$$

We linearize constraint 11 by transforming the logical operators as follows (Eq. 12):

$$1 - x_{irl} + ph_i = p' \div Comp_r, i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (12)$$

$$c'_i = t'_i + ph_i, i = N + 1..N + Nb_Occ \quad (13)$$

$$Tmin_i = t'_{i-1} + T^* + ph_{i-1}, i = N + 2..N + Nb_Occ \quad (14)$$

$$Tmax_i = Tmin_i + (Tmax_{i-1} - Tmin_{i-1}), i = N + 2..N + Nb_Occ \quad (15)$$

$$\sum_{r=1}^R \sum_{l=1}^m x_{irl} = 1, i = N + 1..N + Nb_Occ \quad (16)$$

$$c_i \leq t'_i + L(1 - y_{ii'}), 1 \leq i \leq i' \leq N + Nb_Occ \quad (17)$$

$$c'_i \leq t_i + L(y_{ii'}), 1 \leq i \leq i' \leq N + Nb_Occ \quad (18)$$

$$(x_{irl} = 1) \Rightarrow (t'_i \geq LB_{rl}), i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (19)$$

We linearize constraint 19 as follows (Eq. 20):

$$1 - x_{irl} + t'_i \geq LB_{rl}, i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (20)$$

$$(x_{irl} = 1) \Rightarrow (c'_i \leq UB_{rl}), i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (21)$$

We linearize constraint 21 as follows (Eq. 22):

$$1 - x_{irl} + c'_i \leq UB_{rl}, i = N + 1..N + Nb_Occ, r = 1..R, l = 1..m \quad (22)$$

$$t_i \geq 0, c_i \geq 0, T_i \geq 0, i = 1..N \quad (23)$$

$$t'_i \geq 0, c'_i \geq 0, E'_i \geq 0, T'_i \geq 0, i = N + 1..N + Nb_Occ \quad (24)$$

$$x_{irl}, y_{ii'} \in \{0, 1\} r = 1..R, l = 1..m \quad 1 \leq i \leq i' \leq N + Nb_Occ \quad (25)$$

The objective function (Eq. 6) minimizes the sum of tardiness of production jobs and the sum of earliness/tardiness of the maintenance activities. The constraints 7, 8 and 9 determine the tardiness / earliness of each production job/ maintenance activity by means of its completion or its execution time. All of the completion times of production jobs, the real processing times of the maintenance occurrences according to the assigned human resource, the completion times of the maintenance activities and their tolerance intervals are respectively expressed by the constraints 10, 11, 13, 14 and 15. Constraint 16 ensures that each maintenance activity is treated by only one human resource and in only one availability interval. Constraints 17 and 18 meet the requirement that only one job can be processed at any time. Constraints 19 and 21 ensure that each maintenance activity starts and finishes in the same availability interval. Finally, constraints 23, 24 specify the non-negativity of $t_i, c_i, T_i, t'_i, c'_i, E'_i$ and T'_i , and constraint 25 establishes the binary restrictions of both x_{irl} and $y_{ii'}$. We will show in the sequel how this IP modeling is exploited by Cplex to find the optimal solution for small instances of the problem. To deal with large instances of the problem, we propose in the next section a metaheuristic method that could find good solutions for large instances in a reasonable CPU time.

3 The proposed integrated metaheuristic

Guided Local Search (GLS) (Voudouris and E.Tsang (1995a)) is a penalty-based metaheuristic algorithm that sits on top of other local search algorithms, with the aim to improve their efficiency and robustness. Based on the value of a utility function, the GLS penalizes solution features during each iteration. The penalty acts as a disturbance to an augmented objective function 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 scheduling problems (Voudouris and E.Tsang (1995b), Tischer and Carrión (2008), 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 metaheuristic 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 main features of the approach presently called IGLS for Integrated Guided Local Search can be summarized as follows:

- The use of an integrated representation of production and maintenance data which embeds human resource constraints.
- The definition of two types of features for both production and maintenance activities. The first is related to the advance and the other to the delay.
- The use of several local search heuristics (LS) instead of one as in the original GLS. Indeed, in each iteration of IGLS, a specific LS heuristic is applied according to the feature which corresponds to maximum utility.
- The use of post-optimization process in order to minimize both production and maintenance delays. This process is performed at the end of each local search (section 3.5) ;
- The introduction of a diversifying process to avoid entrapment in local optima after algorithm stall (section 3.6).

The main steps of IGLS are depicted in Algorithm 1. In IGLS, a set of M features is associated with every solution S . A feature f_{if} is determined by the *indicator function* I_{if} , $if \in \{1, \dots, M\}$, as follows:

$$I_{if}(S) = \begin{cases} 1 & \text{if } S \text{ has property } f_{if}, \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

In addition, IGLS associates each feature f_{if} with a cost c_{if} and a penalty pn_{if} . Thus, when the search is trapped in local optima, solutions with certain features are penalized. The selection is made according to the features' utility values:

$$util_{if}(S) = I_{if}(S) \times \frac{c_{if}}{1 + pn_{if}} \quad (27)$$

Given an objective function f , IGLS defines an augmented cost function that modifies f by penalizing the current solution with the worst features' values:

$$h(S_c) = f(S_c) + \lambda \sum_{if=1}^M (pn_{if} \times I_{if}(S_c)) \quad (28)$$

where λ represents the relative weight of penalties with respect to the cost of solution.

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}

```

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

```

The specifications of the IGLS components are discussed in detail in the following subsections.

3.1 The encoding scheme and solution representation

To effectively solve the problem, it is important to select a proper solution representation. A solution S is encoded by a two-field structure.

1. The first field is an integrated sequence S which represents the execution order of production jobs and maintenance activities. Production jobs are represented by the identifier 1 to N and the maintenance occurrences by the identifier $N + 1$ 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 2). $Z[r, l]$ 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 .

Figure 2 shows an example of solution encoding. We consider 3 maintenance occurrences and 6 production jobs. The production jobs are encoded from 1 to 6 and the maintenance occurrences from 7 to 9. Their execution times are respectively 30, 53 and 129. Production and maintenance activities are scheduled in the following sequence: $\{3, 6, 7, 1, 5, 8, 4, 9, 2\}$. Moreover, the human resource HR_1 is assigned to M_2 and execute it inside the availability interval AI_{12} , and HR_2 is assigned to respectively M_2, M_3 and execute them respectively inside the availability intervals AI_{21} and AI_{22} .

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

	AI_{r1}	AI_{r2}	AI_{r3}
HR_1	0	8	0
HR_2	7	9	0

Figure 2: An example of a solution encoding.

3.2 The generation of the initial solution

In this work, we propose a two-step initialization procedure as follows:

- **Step 1.** In this step, the maintenance operations are planned according to the human resource availabilities. Indeed, for planning a maintenance operation M_i , the available human resource is chosen according to the assignment strategy (equity, efficiency and training). Our aim is to minimize the maintenance' earliness/tardiness when the human resource availability intervals and the tolerance period do not overlap. The maintenance activity insertion process is outlined in Algorithm 2.
- **Step 2.** The maintenance sequence obtained in step 1 is scanned and production jobs are inserted one by one either randomly or sorted by using the dispatching rule EDD (Earliest Due Date) taking the maintenance planning as a strong constraint.

3.3 The proposed features

The features are defined to capture the properties of each solution S and then appropriately modify the objective function by adding penalties as described previously. IGLS associates four sets of features F_{if} , where $if \in \{1, 2, 3, 4\}$. Two of them are 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. Thus, F_1 has N features (the total number of production jobs) and F_3 has Nb_Occ features (the total number of maintenance occurrences). The feature F_2 (respectively F_4) represents the advance of a production (respectively maintenance) activity. There are $N + Nb_Occ$ features in F_2 and F_4 . It is worth noting that F_1, F_3 and F_4 are defined explicitly according to the objective function, since F_2 is defined in order to widen the search space to avoid trapped in local

Algorithm 2 The maintenance insertion procedure in the initial schedule.

Input: The maintenance data (p' , $TI = [Tmin, Tmax]$), the human resource data;

Parameter: The assignment strategy;

Output: The integrated initial solution S_0

```

1: for (Each maintenance activity  $M_i$ ) do
2:   if (There is an available resource  $HR_r$  with the availability interval  $AI_{rl}$  according
   the assignment strategy and which overlaps with  $TI_i$ ) then
3:     Schedule  $M_i$  in its tolerance interval;
4:   else
5:     Let  $AI^1$  and  $AI^2$  be the nearest availability intervals not yet assigned, respectively
   coming just before and just after  $TI_i$ ;
6:     Compute the maintenance objective function  $f_m$  when inserting  $M_i$  at  $AI^1$  and
    $AI^2$  and choose the one optimizing the best  $f_m$ ;
7:     Update the scheduling data;
8:     Compute the tolerance interval of the next maintenance activity occurrence  $TI_{i+1} =$ 
    $[Tmin_{i+1}, Tmax_{i+1}]$ ;
9:   end if
10: end for
11: Return  $S_0$ ;

```

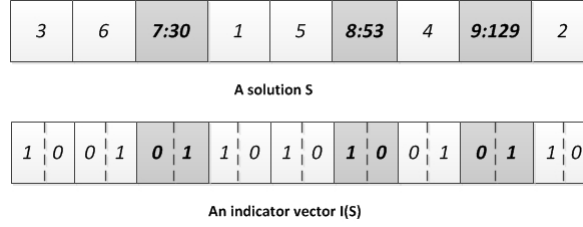


Figure 3: An example of an indicator vector for a solution S.

optima. Moreover, to every solution S , it is associated $2 \times (N + Nb_Occ)$ features. Each feature $F_{i,f,i}$ is associated with an indicator $I_{i,f,i}(S)$, a cost $c_{i,f,i}$ and a utility value $util_{i,f,i}$ such as: $i = 1..N + Nb_Occ$.

All the indicators form a vector \mathbf{I} associated to a solution \mathbf{S} of size $N + Nb_Occ$. Each vector component is composed of two sub-components representing the two features (advance and delay) associated to each activity in S. It is equal to 1 if the corresponding feature is present in the activity and to 0 otherwise. Figure 3 shows an example of an indicator vector. For instance, the production job J_3 is in advance since its corresponding sub-component $F_{2,3}$ is equal to 1. However, the maintenance activity M_9 is in delay since its corresponding sub-component $F_{3,9}$ is equal to 1. The proposed features are the important parameters in IGLS that impact the objective function. We explain this in the following:

(1) *The production tardiness ($F_{1,i}$)*

This first feature indicates whether the production job J_i is in delay according to its due date d_i . Its cost $c_{1,i}$ is expressed as its tardiness T_i according to its due date d_i (Eq. 3). Since we

aim to minimize the sum of production tardiness, the production tardiness becomes naturally a trivial feature of a solution S for the studied problem.

(2) *The production earliness ($F_{2,i}$)*

This feature indicates whether the production job J_i is in advance according to its due date d_i . Its cost $c_{2,i}$ is expressed as its earliness E_i according to its due date d_i (Eq. 29). This feature is used to perform a local search (a neighbourhood structure) even if it is not involved in the objective function. Thus, this feature is a perturbation process which helps the algorithm to escape from current local optima. F_2 is introduced because when we limit ourselves to considering only the tardy jobs, early ones compared to their due dates will rarely be shifted in the scheduling. However, if these jobs are delayed, the cost may be minimized.

$$E_i = \max(0, d_i - c_i), i = 1..N \quad (29)$$

(3) *The maintenance tardiness ($F_{3,i}$)*

This feature indicates whether the maintenance operation M_i is in delay according to its tolerance interval TI_i . That is, the maintenance operation is planned after its $Tmax_i$. Its cost $c_{3,i}$ is expressed by its tardiness (Eq. 4).

(4) *The maintenance earliness ($F_{4,i}$)*

This feature indicates whether the maintenance operation M_i is in advance according to its TI_i . This means that the maintenance operation is planned before its $Tmin_i$. Its cost $c_{4,i}$ is expressed by its earliness (Eq. 4).

3.4 Local search heuristics

A specific local search (LS) heuristic is applied in IGLS for each feature set. Therefore, four 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 randomly an appropriated neighborhood structure. In the following, we detailed the four proposed LS.

The local search LS_1

LS_1 is used when the activity with the maximum utility is a tardy job, aiming the production objective function improvement. Let J_i be this job. The objective is to reschedule J_i in the integrated sequence in order to minimize its delay. That is, its completion time c_i has to be as close as possible to its due date d_i . Two cases could occur (Algorithm 3).

1. The activity planned at the due date of J_i is a production job. In this case, we reschedule J_i at $c_i = d_i$.
2. The activity planned at the due date of J_i is a maintenance activity M_j . In this case:
 - If the processing time p_i of the job J_i is less than the duration of the interval separating the two maintenance occurrences M_j and M_{j-1} , then reschedule J_i before the maintenance activity M_j so that its completion time is equal to the execution time of M_j ($c_i = t_j$).

- Otherwise, we reschedule J_i at the completion time of the maintenance activity M_j ($t_i = c_j$). In this case, the job J_i remains in delay but it will be minimized.

Algorithm 3 LS_1 .

Input: The solution S , The activity J_i with d_i and p_i , $h(S)$, The scheduling data (production, maintenance and human resource);

Output: The solution S ;

- 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 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 just before the maintenance activity;
 - 9: **end if**
 - 10: Reschedule the production jobs in the same order than in S ;
 - 11: **if** ($h(S_{new}) < h(S)$) **then**
 - 12: Return S_{new} ;
 - 13: **else**
 - 14: Return S ;
 - 15: **end if**
-

The local search LS_2

LS_2 is performed when the activity with the highest utility is a early job which completes before its due date (Eq. 29). This local search is introduced because when we consider only the tardy jobs, early ones compared to their due dates will rarely be shacked in the schedule. However, it could be possible that the global cost is minimized when these jobs will be delayed. The principal of this local search is depicted in Algorithm 4.

Algorithm 4 LS_2 .

Input: The solution S , J_i the production job with the maximum utility, $h(S)$, The scheduling data (production, maintenance and human resource);

Output: The solution S ;

- 1: Reschedule J_i at the end of the schedule with advancing only the production jobs as possible;
 - 2: **if** ($h(S_{new}) < h(S)$) **then**
 - 3: Return S_{new} ;
 - 4: **else**
 - 5: Return S ;
 - 6: **end if**
-

The local search LS_3

LS_3 is used when the activity with maximum utility is a maintenance one in delay according to its tolerance interval. In LS_3 , we try to improve the maintenance objective function. Let M_i be this maintenance activity executed in the availability interval AI_{rl} . Two cases could occur aiming to reduce maintenance activity tardiness (Algorithm 5):

1. M_i could be advanced in the same availability interval AI_{rl} of the human resource assigned to it;
2. If this is not possible, reschedule M_i in the nearest available interval that is coming just before the interval AI_{rl} . We do this, even when this interval corresponds to another human resource.

Algorithm 5 LS_3 .

Input: The solution S , 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 ;

- 1: **if** (M_i can be advanced in AI_{rl}) **then**
 - 2: $S_{new} \leftarrow S$ with the new maintenance planning;
 - 3: **else**
 - 4: $S_{new} \leftarrow S$ with the planning of M_i at the nearest availability interval coming just before AI_{rl} and that does not disturb the planning of M_{i-1} ;
 - 5: **end if**
 - 6: Update the tolerance intervals TI_{i+1} of M_{i+1} ;
 - 7: Update the human resource availabilities;
 - 8: **if** ($h(S_{new}) < h(S)$) **then**
 - 9: Return S_{new} ;
 - 10: **else**
 - 11: Return S ;
 - 12: **end if**
-

The local search LS_4

LS_4 is used when the activity with the maximum utility is a maintenance in advance according to its tolerance interval. Let M_i be this maintenance activity executed in the availability interval AI_{rl} . Two cases could occur aiming to reduce maintenance activity earliness (Algorithm 6):

1. M_i can be delayed in the same availability interval of the human resource assigned to it;
2. If not, reschedule M_i in the nearest available interval that is just after its assigned one. This interval could correspond to another human resource.

Algorithm 6 LS_4 .

Input: The solution S , 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 ;

```

1: if ( $M_i$  can be delayed in  $AI_{rl}$ ) then
2:    $S_{new} \leftarrow S$  with the new maintenance planning;
3: else
4:    $S_{new} \leftarrow S$  with the planning of  $M_i$  at the nearest availability interval coming just
   after  $AI_{rl}$  and that does not disturb the planning of  $M_{i+1}$ ;
5: end if
6: Update the tolerance intervals  $TI_{i+1}$  of  $M_{i+1}$ ;
7: Update the human resource availabilities;
8: if ( $h(S_{new}) < h(S)$ ) then
9:   Return  $S_{new}$ ;
10: else
11:   Return  $S$ ;
12: end if

```

3.5 The post-optimization procedure

This procedure is performed after each local search. It leads to minimize the idle times between two successive activities. Reducing the idle time of the machine by advancing the production jobs leads to possible improvements of the production objective function. One could shake the maintenance activities to advance the production jobs. We propose here, a three step-optimization strategy (Algorithm 7):

- **Step 1.** Delay each maintenance activity M_i in the same availability interval of the human resource assigned to it;
- **Step 2.** Reschedule each production job planned after a maintenance M_i having an execution time less than or equal to the idle time between M_i and the activity scheduled before it.
- **Step 3.** The obtained solution replace the one issued from the local search when it is better.

3.6 The restart scheme procedure

The local searches could be not sufficient to overcome the constraints imposed by human resources to explore efficiently the search space. Indeed, during the research process, IGLS could sometimes stall around a local optimum. Hence the interest in introducing a restart mechanism. If the best solution found is not improved after $Nb_{Improve}$ iterations, then we apply the LS to an activity (a production or a maintenance one) that is randomly selected among the $N + Nb_{Occ}$ activities. The goal is to break a good solution in order to explore a new search sub-space. Applying one of the neighborhood structures will generate a schedule that is usually more expensive but will allows to explore new search areas.

Algorithm 7 The post-optimization procedure.

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

Output: The solution S ;

```

1: for (Each maintenance occurrence  $M_i, i = N + 1 \dots N + Nb\_Occ$ ) do
2:   Delay  $M_i$  in  $AI_{r_i}$ ;
3:   Advance the production jobs coming after  $M_i$ ;
4:   Update the tolerance intervals  $TI_{i+1}$ ;
5: end for
6: if ( $h(S_{new}) < h(S)$ ) then
7:   Return  $S_{new}$ ;
8: else
9:   Return  $S$ ;
10: end if

```

3.7 The regulation parameter and stopping criteria

In IGLS, the local search process ensures intensification in the search space. To make a good balance with diversification we propose two approaches for setting the regulation parameter λ . It could be done in static way and its value does not change during all the iterations of the algorithm. The other way is to set the parameter dynamically. In this case, the parameter value is adjusted after a number of iterations to provide a good exploration of the research space and accelerate the search towards global optima. The IGLS process terminates after a maximum number Nb_{It} of iterations.

4 Experimental results

In this section, we present the results of the performed experiments to assess the proposed IGLS algorithm performances. 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 classified into three sets of tests. In the first set, we have undertaken a sensitivity analysis to tune IGLS parameters and evaluate the effect of multiple factors on the algorithm performances. The second set analyses the effectiveness of IGLS. We also compared IGLS to the exact method Cplex on small instance in order to see the solution quality of IGLS against the presented Cplex model. Finally, we study the impact of the human resource assignment strategy on the obtained results. The complete details are reported in the next sections.

The experiments are conducted by considering the same data than the one used in Touat et al. (2018). We used two types of data. The first one is related to production/maintenance, while the second is related to the human resources. Our test bed consists of a total of 150 instances of various sizes according to the number of jobs $N \in \{9, 10, 11, 12, 13, 20, 40, 60, 80, 100, 140, 200, 300, 500, 700\}$. These instances are grouped into 3 subsets according to N : 1) small size instances ranging from 9 to 13 jobs; 2) medium ones ranging from 20 to 100 jobs and 3) large ones ranging from 140 to 700 jobs.

The availability intervals of the human resources are generated according to the production horizon and all existing skills depend on the maintenance activity type. 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, using only one human resource in the workshop could lead to an overwork, whereas in practice, the assignment of human resources to tasks follows the workload regulation. On the other hand, considering three human resources or more leads to a high availability. Using two human resources looks to be the best compromise. This ensures that there is always an available resource which is not always at the needed time. Both $Comp_1$ and $Comp_2$ are distributed as $U]0, 2[$. In order to execute the occurrence M_i by a human resource, two different durations ph_1 and ph_2 are possible according to the competence level of the assigned human resource ($Comp_1$ and $Comp_2$). That is, $ph_1 = p' \div Comp_1$ and $ph_2 = p' \div Comp_2$. We used the method presented in Touat et al. (2018) to generate the set AI_r of availability intervals with $l \in \{1, 2\}$. Here, we propose two types of availability intervals according to their wideness:

- *The strict availability interval (SAI);*
- *The large availability interval (LAI).*

According to the human resource characteristics such as higher and lower 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.

The parameter λ is computed as follows:

- **Static approach:** We vary λ between 0 and 1 with a step of 0.2 between the different values.
- **Dynamic approach:** We vary λ during the execution of the algorithm in three different ways:
 1. The ratio between the cost of the best solution found so far " S_{best} " and the one of the current solution " S_c " according to Eq. 30.

$$\lambda = \frac{f(S_{best})}{f(S_c)} \quad (30)$$

2. The distance between the current solution S_c and the best one S_{best} according to Eq. 31.

$$D(S_c, S_{best}) = 1 - \frac{nb}{N + Nb_{Occ}} \quad (31)$$

nb is the number of activities moved in S_c when compared to S_{best} .

3. The coefficient of variation CV proposed in Ladj et al. (2016), a standardized measure that refers the dispersion degree of the population individuals. Since IGLS is a one-solution based metaheuristic, the population is composed of all intermediate solutions (S') generated at each iteration of the search process (Eq. 32).

$$CV = \frac{\text{standard_deviation}(\text{average}(S'), S_c)}{\text{average}(S')} \quad (32)$$

We recall that the objective function to optimize (Eq. 5) is related to both production and maintenance criteria.

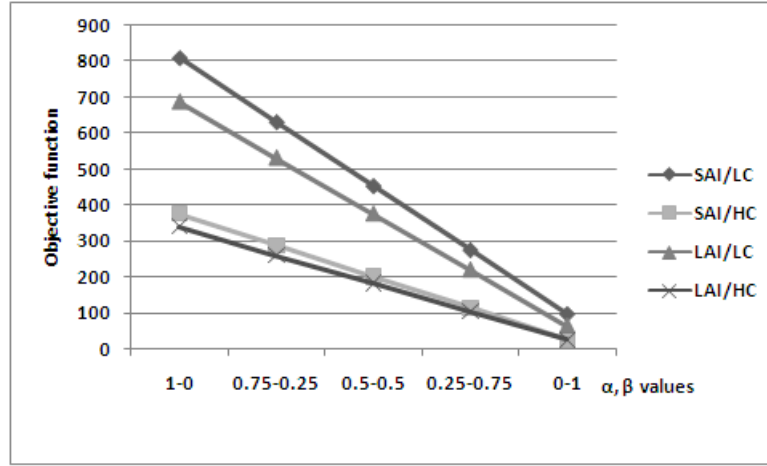


Figure 4: The value of the objective function w.r.t α and β variations.

4.1 IGLS Parameters setting

In this set of experiments, we have undertaken a sensitive analysis of performance for the proposed IGLS by varying different parameters. First, we measure the impact of weighting parameters α and β on the objective function. We use benchmark 13 instances and we generate the initial solution according to the EDD rule and a dynamic λ expressed as a distance.

Figure 4 shows the obtained results for the four classes of experiments computed with respect to the combinations of α, β taken in $\{0, 0.25, 0.5, 0.75, 1\}$.

The impact of production criterion is trivial from the figure for the four classes of experiments. It can be seen that when $\alpha = 1$ and $\beta = 0$, the values of the objective function are the highest. This corresponds to the extreme case where the global objective function is reduced to the production one (f_p in Eq. 3). The objective function values decrease when α decreases and β increases until reaching the other extreme situation where $\alpha = 0$ and $\beta = 1$ corresponding to the case where the global objective function is reduced to the maintenance one (f_m in Eq. 4).

For the rest of experiments, the values of the control parameters α and β of Eq. 5 are set to 0.5. The proposed common weighted global objective function will allow tackling the studied problem in a simplified way.

To calibrate the static value of λ , we used benchmark 13 instances and we generate the initial solution according to the EDD rule and λ is taken in $\{0, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$. Table 1 shows the obtained results for the four classes of experiments. The values of objective function is the one representing the average value for ten generated instances.

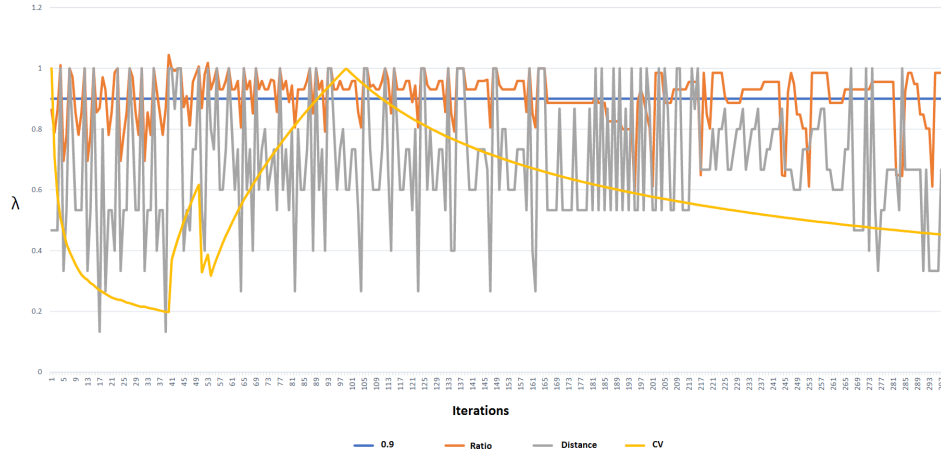


Figure 5: Variation of the different values of λ .

λ	<i>SAI/LC</i>	<i>SAI/HC</i>	<i>LAI/LC</i>	<i>LAI/HC</i>
0	530.85	228.00	487.70	233.00
0.1	517.55	228.20	483.00	233.35
0.3	518.30	223.90	485.65	241.95
0.5	525.70	224.75	484.95	232.45
0.7	512.10	228.60	486.15	232.05
0.9	509.05	229.35	480.20	227.50
1	519.80	229.35	486.45	236.90

Table 1 The results of IGLS obtained with static values of λ .

We can see that the setting $\lambda = 0.9$ gives, in general better solutions than the results of the other static values λ . For this reason, we take this value to represent the static values of λ for the future tests.

We launch the IGLS program on instance 10 of benchmark 13, for the different ways of variation, static and dynamic (ratio, distance and coefficient of variation), of the regulation parameter λ . We save the values of λ for each iteration. Figure 5 shows the variation of the different values of λ .

We notice that the λ values computed as *CV* vary continuously because each value depends on the set of solutions found at a precise iteration while the ratio and the distance only depend on the current solution and the best one found at all iterations, so a value at an iteration $i + 1$ does not depend on the value found at an iteration i .

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 here for the sake of concise presentation. These values are the following:

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

4.2 IGLS performance analysis on small size instances

The second set of experiments aims to evaluate the performance of the proposed IGLS compared to the optimal results obtained by Cplex (the ILOG OPL implementation) on small size instances corresponding to $N \in \{9, 10, 11, 12, 13\}$.

The time limit of Cplex is set to one hour. This study merely reports the problem size that the Cplex method can optimally solve within one hour CPU time. The relative percentage deviation (RPD) is used as an index to evaluate the solution quality and the performance of the proposed metaheuristic. RPD represents the deviation of the objective function f (Eq. 5) provided by IGLS with respect to the optimal solution f_{IP} given by Cplex (Eq. 6) (see Eq. 33).

$$RPD = \frac{(f - f_{IP})}{f_{IP}} \times 100 \tag{33}$$

Table 2 and Table 3 show the obtained results for an initial solution generated respectively according to the EDD rule and randomly. For each value of λ and each benchmark instances, we report the average value of the global objective function obtained by running 10 instances, and 10 repetitions for each instance, and the number of instances for which the proposed IGLS find the optimal solution "Best". It should be noted that we performed $5 \times 10 \times 4 \times 4 = 800$ tests for the case of an initial solution according to EDD rule and $5 \times 10 \times 10 \times 4 \times 4 = 8000$ executions for the case of an initial solution generated randomly.

λ	N	SAI							
		LC				HC			
		f_{IP}	f	RPD	Best	f_{IP}	f	RPD	Best
0.9	9	423.95	454.95	7.31%	10	134.95	143	5.97%	12
	10	404.55	450.55	11.33%		147.55	156.35	5.96%	
	11	441.25	465.55	5.51%		174.70	187.20	7.16%	
	12	507.10	545.80	7.63%		216.00	237.75	10.07%	
	13	418.25	456.75	8.21%		183.25	195.70	6.79%	
AVG							7.19%		
Ratio	9	423.95	459.75	8.44%	14	134.95	144.95	7.41%	12
	10	404.55	449.65	11.15%		147.55	157.65	6.85%	
	11	441.25	461.85	4.67%		174.7	189.9	8.7%	
	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%	
AVG			10.18%				8.43%		
Distance	9	423.95	459.85	8.47%	15	134.95	143.4	6.26%	13
	10	404.55	451.4	11.58%		147.55	159.65	8.2%	
	11	441.25	459	4.02%		174.7	188.35	7.81%	
	12	507.1	543.25	7.13%		216	237.45	9.93%	
	13	418.25	454.3	8.62%		183.25	200.25	6.8%	
AVG			7.96%				7.80%		
CV	9	423.95	460.05	8.52%	11	134.95	151.15	12%	8
	10	404.55	454.35	12.31%		147.55	163.3	10.67%	
	11	441.25	463.5	5.04%		174.7	190.8	9.21%	

λ	N	LC				HC			
		f_{IP}	f	RPD	Best	f_{IP}	f	RPD	Best
	12	507.1	554.15	9.28%		216	237.1	9.77%	
	13	418.25	464.3	11.01%		183.25	202.8	10.67%	
AVG				9.23%				10.46%	
		LAI							
		LC				HC			
		f_{IP}	f	RPD	Best	f_{IP}	f	RPD	Best
0.9	9	304.6	330.45	8.49%	5	106.8	120.35	12.69%	8
	10	284.65	311.65	9.45%		121.65	133.65	9.86%	
	11	356.3	396	11.14%		147.4	169.24	14.82%	
	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%	
AVG				9.96%				11.54%	
Ratio	9	304.6	330	8.44%	4	106.8	117.95	10.44%	8
	10	284.65	310	8.91%		121.65	136.25	12%	
	11	356.3	410.1	15.1%		147.4	166.85	13.2%	
	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%	
AVG				10.51%				10.34%	
Distance	9	304.6	327.75	7.6%	4	106.8	116.95	9.5%	9
	10	284.65	309.8	8.84%		121.65	129.75	6.66%	
	11	356.3	384.3	7.86%		147.4	162.95	10.55%	
	12	368.15	395.95	7.55%		176.55	196.4	11.24%	
	13	349.45	375.7	7.51%		172	181.95	5.87%	
AVG				7.87%				8.76%	
CV	9	304.6	330.3	8.44%	4	106.8	119.35	11.75%	9
	10	284.65	310	8.91%		121.65	139.15	14.39%	
	11	356.3	410.15	15.11%		147.4	170.55	15.71%	
	12	368.15	413.7	12.37%		176.55	196.4	11.24%	
	13	349.45	376.2	7.65%		172	181.95	5.78%	
AVG				10.49%				11.77%	

Table 2: IGLS results on small size instance according to an EDD based generated initial solution.

λ	N	SAI							
		LC				HC			
		f_{IP}	f	RPD	Best	f_{IP}	f	RPD	Best
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%		174.7	185.6	6.24%	
	12	507.1	548.35	8.13%		216	247.05	14.38%	
	13	418.25	457.95	9.49%		183.25	193	5.32%	
AVG				6.73%				7.56%	
Ratio	9	423.95	443.35	4.58%	9	134.95	144.85	7.34%	11
	10	404.55	448.15	10.78%		147.55	156.7	6.20%	
	11	441.25	462.80	4.88%		174.7	192.05	9.93%	
	12	507.1	537.35	5.97%		216	240.2	11.2%	
	13	418.25	455.75	8.97%		183.25	199.6	8.92%	
AVG				7.03%				8.71%	
Distance	9	423.95	444.85	4.93%	15	134.95	144.3	6.93%	15
	10	404.55	437.40	8.12%		147.55	155.45	5.35%	
	11	441.25	459	4.02%		174.7	187.45	7.30%	
	12	507.1	535.25	5.55%		216	230.45	6.68%	
	13	418.25	453.8	8.5%		183.25	196.3	7.12%	
AVG				6.22%				6.67%	
CV	9	423.95	445.65	5.12%	11	134.95	149.15	10.52%	11
	10	404.55	449.35	11.07%		147.55	160.7	8.91%	
	11	441.25	463.5	5.04%		174.7	191.6	9.67%	
	12	507.1	554.15	9.28%		216	241	11.57%	
	13	418.25	464.1	10.96%		183.25	199.85	9.06%	
AVG				8.29%				9.94%	
		LAI							
		LC				HC			
		f_{IP}	f	RPD	Best	f_{IP}	f	RPD	Best

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%		
Ratio	9	304.6	320	5.06%	3	106.8	115.75	8.38%	8
	10	284.65	303.25	6.53%		121.65	134.5	10.56%	
	11	356.3	384.6	9.94%		147.4	165.75	12.45%	
	12	368.15	404.95	10%		176.55	194.2	10%	
	13	349.45	377.4	8%		172	185.6	7.91%	
AVG			7.90%				9.86%		
Distance	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	197.70	11.98%	
	13	349.45	381.70	9.23%		172.00	184.45	7.24%	
AVG			9.12%				9.26%		
CV	9	304.60	330.30	8.44%	11	106.80	120.75	13.06%	9
	10	284.65	311.80	9.54%		121.65	136.05	11.84%	
	11	356.30	398.65	11.89%		147.40	168.6	14.38%	
	12	368.15	409.40	11.20%		176.55	199.10	12.97%	
	13	349.45	374.45	8.58%		172.00	184.65	7.35%	
AVG			9.93%				11.92%		

Table 3: IGLS results on small size instance according to a randomly generated initial solution.

From Tables 2 and 3 we can see that the obtained results show that for most of the tested instances, the RPD does not exceed 11% for the four classes of experiments. Except 20% when the EDD rule is performed to generate the initial solution and 8% when the initial solution is generated randomly. In the average RPD varies between 5 and 10%, (65%) of cases for the EDD rule and (74%) of cases for a random initial solution. The best results of the IGLS are obtained for the *LAI/HC* class. It is even lower than 3% (2.63%) in the case of *LAI/HC* where $N = 11$ and the initial solution is generated randomly.

For the 50 instances used for the tests, we can see that the proposed IGLS with a dynamic value of λ allows to find the maximum number of the obtained optimal results (columns 6 and 10 of Tables 2 and 3). This is due to the adaptation of the regulation parameter to each obtained solution. Thus, IGLS finds the optimal solution in around 30% of the cases when the initial solution is generated randomly for the four classes of experiments.

We report on Figure 6 the CPU times in seconds of the IP model on small instances. One can notice that the exact method is able to solve optimally instances having 13 jobs or less. This confirms the hardness of solving in practice the studied problem. We can remark also that the exact method Cplex spent more time to find the optimal solution in the case of *LAI/LC* and vary in [3.34, 634.32] (see Figure 6). This can be explained by the fact that Cplex explores more solutions in this case, since the search space is large when considering large availability intervals (LAI). Besides, when the competences are low the maintenance duration becomes large and then it becomes difficult to find an available resource. We do not report the execution times of IGLS on small benchmark instances since they are very close and less than 0.16 seconds. Moreover, computational times of Cplex increase when the size of the instance increases.

We present in Table 4 a comparative study with the heuristic proposed in Touat et al. (2018). The objective function of this heuristic is denoted by f_H and represented in both columns 3 and 4. We retain the best results obtained by the IGLS in both cases when the initial solution is generated randomly or according to the EDD rule. The RPD is computed

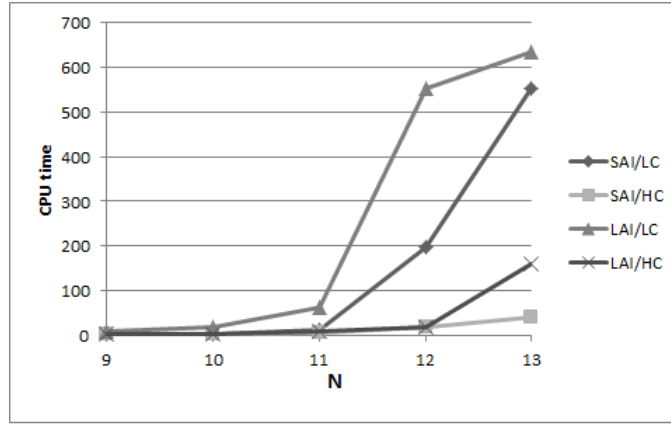


Figure 6: Computational times in second of the IP model on small size problems.

with respect to Eq. 33 where f_{IP} is replaced by f_H and its values are given in columns 5 and 8.

Initial_sol	N	SAI					
		LC			HC		
		f_H	f	RPD	f_H	f	RPD
EDD	9	476.20	459.85	-3.43%	146.60	143.00	-2.45%
	10	448.40	451.40	0.67%	161.70	156.35	-3.30%
	11	475.45	459.00	-3.45%	181.65	187.20	3.05%
	12	538.20	543.25	-3.45%	234.80	237.75	1.25%
	13	456.00	454.30	-0.37%	193.75	195.70	1.00%
AVG				-2.00%			-0.09%
Random	9	476.20	444.85	-6.58%	146.60	144.30	-1.56%
	10	448.40	437.40	-2.45%	161.70	155.45	-3.87%
	11	475.45	459.00	-3.45%	181.65	187.45	3.19%
	12	538.20	535.25	-0.54%	234.80	230.45	-1.85%
	13	456.00	453.80	-0.48%	193.75	196.30	1.31%
AVG				-2.70%			-0.55%
Initial_sol	N	LAI					
		LC			HC		
		f_H	f	RPD	f_H	f	RPD
EDD	9	321.40	327.75	1.97%	113.80	116.95	2.76%
	10	321.80	309.80	-3.72%	127.40	129.75	1.84%
	11	395.10	384.30	-2.73%	155.65	162.95	4.69%
	12	412.70	395.95	-4.05%	191.60	196.40	2.50%
	13	365.40	375.70	2.81%	177.55	181.95	2.47%
AVG				-1.14%			2.85%
Random	9	321.40	326.35	1.54%	113.80	115.40	1.40%
	10	321.80	307.95	-4.30%	127.40	129.90	1.96%
	11	395.10	377.70	-4.40%	155.65	165.50	6.32%
	12	412.70	405.85	-1.65%	191.60	197.70	3.18%
	13	365.40	372.85	2.03%	177.55	184.45	3.88%
AVG				-1.35%			3.34%

Table 4: IGLS results on small size instances.

We remark from Table 4 that IGLS performs better than the heuristic proposed in Touat et al. (2018) in $\frac{3}{4}$ of the cases. This is true when the initial solution is generated randomly and when it is computed according to the EDD rule.

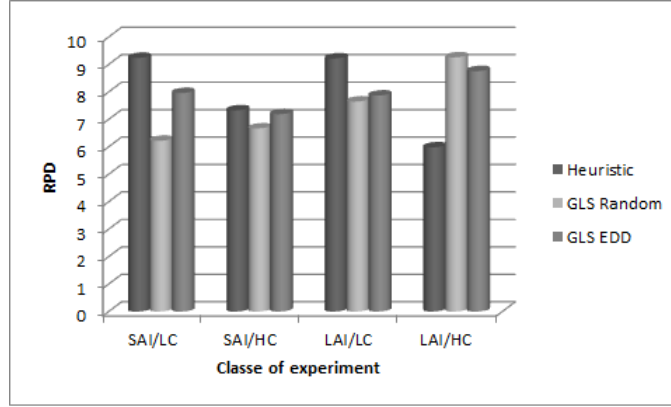


Figure 7: The RPD values of both IGLS and the heuristic proposed in Touat et al. (2018).

We present in Figure 7 the RPD of the IGLS method and the one of the heuristic proposed in Touat et al. (2018) for both cases of the initial solution generation. The RPD values are picked up from Tables 2 and 3.

We remark that IGLS gives the best results for the three cases of experiments: SAI/LC, SAI/HC and LAI/LC.

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

In previous tests, we used the RPD (Eq. 33) to present the IGLS results. However, it could be interesting to present the dispersion too. This later is expressed according to the standard deviation. First, we compute for each instance its deviation from the average as mentioned in Eq. 34, then the standard deviation of each benchmark instances as mentioned in Eq. 35:

$$Var_{ins} = \sqrt{(RPD_{ins} - RPD_{bench})^2} \quad (34)$$

$$Var_{bench} = \frac{\sum_{ins=1}^{10} \sqrt{(RPD_{ins} - RPD_{bench})^2}}{10} \quad (35)$$

where RPD_{ins} is the deviation of one instance and RPD_{bench} is the deviation of one benchmark instances (the average of RPD_{ins}). Table 5 shows the obtained results.

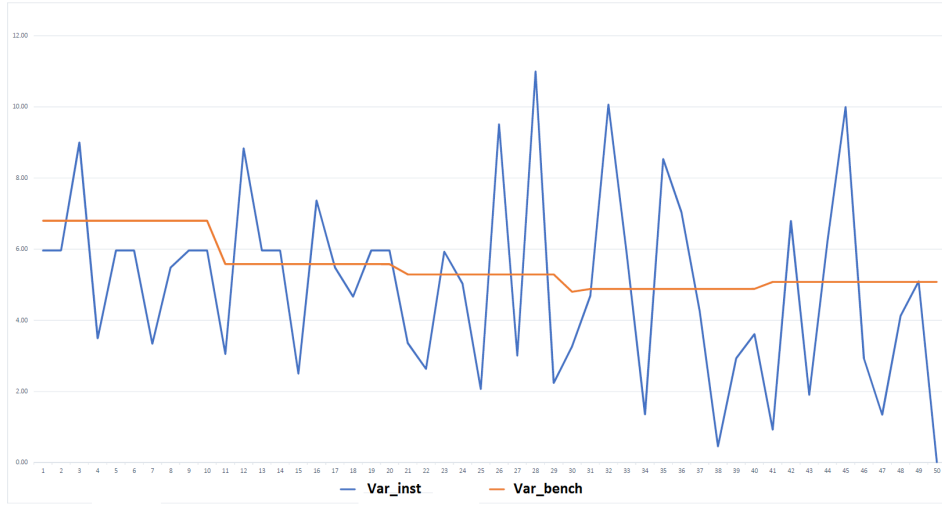


Figure 8: Deviation of instances from their averages.

Initial solution	N	SAI		LAI	
		LC (%)	HC (%)	LC (%)	HC (%)
<i>EDD</i>	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
<i>Random</i>	9	5.61	7.41	5.86	6.66
	10	7.66	5.77	6.10	7.54
	11	1.98	6.39	5.36	8.16
	12	6.12	7.99	7.12	7.77
	13	6.62	3.51	5.93	8.23

Table 5 Standard deviation results.

We remark from Table 5 that the standard deviations do not exceed 9%. Moreover, all instances' deviations are in $[0.01, 21.44]$ for all of the cases. Based on these results, one can see the stability of our method.

We chose the case of SAI/HC and the initial solution generated according to the EDD rule. Then we present in Figure 8 the graph which shows in detail the standard deviation of the 50 instances compared to the average of the standard deviations for each benchmark instances. Thus, this represents the dispersion of the data relative to the center.

We remark that the gaps are very acceptable since they do not exceed 5.71 (see the instance 8 of benchmark 11).

4.3 Performance analysis of IGLS on large size instances

In this set of experiments, we consider benchmark instances up to 700 jobs and fixed the execution time limit to 60 seconds. Table 6 shows the obtained results by both IGLS and the

heuristic of Touat et al. (2018) on these relatively large size instances. For each instance, we give respectively, the values of the IGLS production objective function (f_p), the ones of IGLS maintenance objective function (f_m), the ones of IGLS global objective function (f), the ones of the heuristic objective function (f_H), the IGLS CPU times ($T_{GLS}(s)$) and the heuristic CPU times ($T_H(s)$). We also compared the IGLS results to ones of the heuristic proposed in Touat et al. (2018) by computing the RPD values according to Eq. 33 where f_{IP} is replaced by f_H . We can see that IGLS is able to find good solutions to these benchmark instances by the time limit.

SAI/LC							
N	f_p	f_m	f	f_H	RPD	$T_{GLS}(s)$	$T_H(s)$
20	3593.90	1032.00	2312.95	2692.00	-14.08%	0.028	0.002
40	9052.00	3920.70	6486.35	10369.90	-37.44%	0.092	0.007
60	26418.60	1520.80	16250.90	17996.00	-9.70%	3.19	0.017
80	32220.50	581.00	16400.75	40526.00	-59.53%	1.731	0.037
100	41188.40	926.80	21057.60	57717.40	-63.52%	3.311	0.06
140	64783.70	1776.20	33279.95	61211.70	-45.63%	8.954	0.14
200	162285.20	3924.60	83104.90	65613.30	26.66%	22.879	0.27
300	293188.00	9706.60	151447.30	396547.00	-41.81%	44.427	1.40
500	555250.10	30077.40	292663.75	964704.00	-49.66%	55.842	7.267
700	985488.10	61968.60	523728.35	1218075.60	-57.00%	61.397	24.95
SAI/HC							
N	f_p	f_m	f	f_H	RPD	$T_{GLS}(s)$	$T_H(s)$
20	2069.30	73.70	1071.50	783.60	36.00%	0.028	0.004
40	8158.10	207.80	4182.95	3011.50	38.80%	2.891	0.0083
60	10499.40	437.60	5468.50	4493.80	21.68%	2.547	0.014
80	16647.90	527.90	8587.90	9734.20	-11.78%	1.186	0.029
100	20861.90	935.70	10898.80	11914.20	-8.52%	2.347	0.028
140	33085.50	1853.20	17469.40	20244.30	-13.71%	6.577	0.089
200	83266.50	3979.10	43622.80	46821.60	-6.83%	17.44	0.18
300	238167.00	9448.30	123808.00	148353.00	-16.55%	37.01	0.59
500	561495.00	28192.30	294843.00	652866.00	-44.84%	48.72	4.864
700	985705.00	62054.30	523879.00	1134890.40	-53.84%	55.28	14.68
LAI/LC							
N	f_p	f_m	f	f_H	RPD	$T_{GLS}(s)$	$T_H(s)$
20	4515.70	414.40	2465.05	2026.70	21.62%	1.906	0.0021
40	16919.40	1519.60	9219.50	7697.60	19.77%	1.309	0.0082
60	29079.70	752.20	14916.00	12608.60	18.29%	1.17	0.0187
80	33517.80	3724.20	18621.00	21473.50	-13.28%	1.283	0.041
100	33409.70	4772.90	19091.60	20556.00	-7.12%	3.315	0.039
140	58467.56	5315.23	31891.40	33932.10	-6.01%	8.25	0.092
200	131645.06	11967.73	71806.40	67258.50	6.76%	24.06	0.152
300	473505.62	15274.37	244390.00	284965.00	-14.23%	38.321	1.17
500	1455782.96	73284.00	764533.48	1442516.00	-47.00%	49.518	5.87
700	992342.80	59212.30	525777.55	2323392.60	-77.37%	62.52	16.59
LAI/HC							
N	f_p	f_m	f	f_H	RPD	$T_{GLS}(s)$	$T_H(s)$
20	1417.00	134.40	775.70	575.30	34.83%	0.024	0.002
40	6360.60	609.80	3485.20	2695.20	29.31%	0.122	0.0074
60	10416.20	501.90	5459.05	4279.90	27.55%	0.814	0.0093
80	16900.30	341.70	8621.00	9755.10	-11.62%	1.283	0.039
100	21698.30	464.80	11081.60	12044.00	-7.99%	3.315	0.028
140	46355.70	1426.90	23891.30	20787.40	14.93%	8.25	0.093
200	112827.00	2783.00	57805.20	47789.30	20.95%	24.06	0.207
300	197664.00	9115.80	103390.00	121007.00	-14.55%	38.321	0.63
500	553593.00	29021.80	291307.00	632641.00	-53.95%	49.518	4.51
700	983273.00	62516.70	522895.00	1458265.20	-64.14%	62.52	10.82

Table 6 The IGLS results for the large size problems.

We can see that for both classes of benchmark instances *SAI* and *LAI*, the quality of the three objective functions (f_p, f_m , and f) are better when considering human resources with high competences (*HC*). We can also remark that the CPU times spent to solve the medium instances having sizes ranging from 20 to 140 are comparable for the four classes of

benchmark instances (*SAI/LC*, *SAI/HC*, *LAI/LC* and *LAI/HC*). They are all solved efficiently in less than 9 seconds.

Finally, one can notice that the IGLS performs better than the heuristic proposed in Touat et al. (2018) in the majority of cases. The RPD value could reach very low values when the sizes of the benchmark instances are large (500 and 700).

4.4 The impact of human resource assignment strategy

In this class of experiment, we will check the different human resource assignment strategies corresponding to the maintenance workers' experience (efficiency), equity and training presented in section 2.3 to measure their impact on the obtained results. The assignment is done according to a simple selection heuristic. We used the same production and maintenance data as the previous experiments with the instance size ranging from 20 to 700 production jobs. However, we generated new instances for the human resources. We considered 4, 6 and 8 human resources in the workshop with competences generated randomly in the interval $]0,2[$. The availabilities of medium wideness are generated as in Touat et al. (2018).

Table 7 shows the obtained results. For each benchmark instances, we give three global objective functions representing the average values of ten generated instances. These objective functions correspond to the three human resource strategies that we used. We denoted such objective functions by $f_{efficiency}$, $f_{training}$, and f_{equity} .

Human resources	N	$f_{efficiency}$	$f_{training}$	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 7 IGLS results according to the human resource assignment strategy.

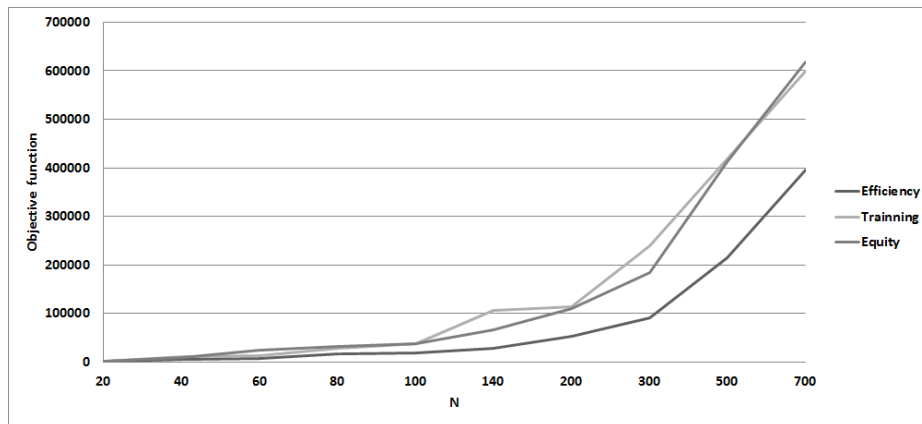


Figure 9: The objective function variation in a 4-resource workshop.

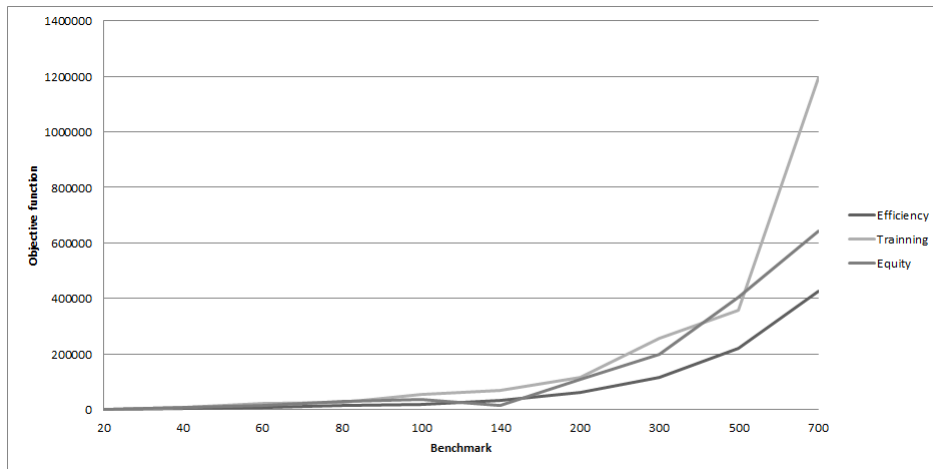


Figure 10: The objective function variation in a 6-resource workshop.

We remark from Table 7 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 of the benchmark instances. This is due to the fact that the strategy based on efficiency involves the human resources having a high competence level unlike the training strategy which involves novice human resources. The quality of the results obtained by the equity strategy is less than the one issued from the strategy based on efficiency. However, the results of the equity strategy are better than those of the training strategy, except for benchmark 200 instances. Consequently, the operating time of maintenance activities is reduced and their tardiness too. This reduces the tardiness of all production jobs scheduled after such maintenance activities. The behavior of the different human resource strategies and their impact on the corresponding objective function is shown in Figures 9, 10, and 11.

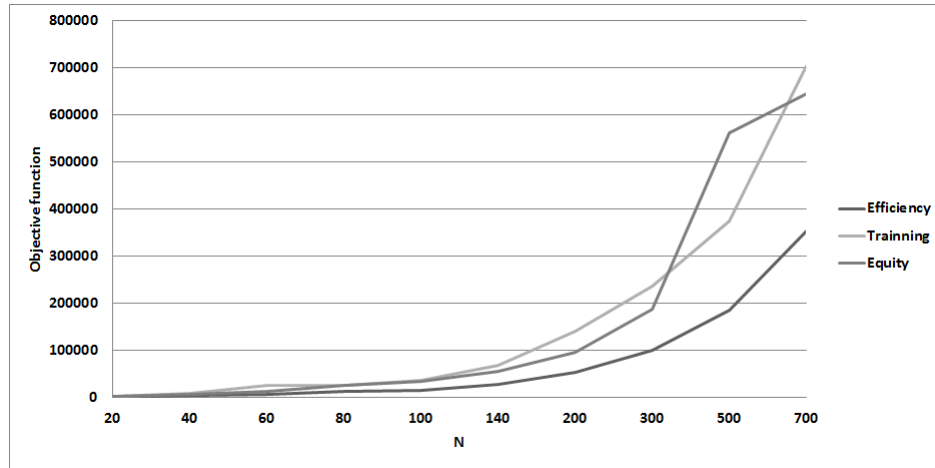


Figure 11: The objective function variation in a 8-resource workshop.

To explain in more detail the impact of strategy choice, we present in Table 8, for some randomly selected instances, the hourly workload of the human resources for each strategy. The competence line refers to the competence of the human resource and the corresponding operating time to perform it.

<i>N/inst</i>	<i>Strategy</i>	<i>HR₁</i>	<i>HR₂</i>	<i>HR₃</i>	<i>HR₄</i>	<i>HR₅</i>	<i>HR₆</i>
40/1	Competence	57/0,8	57/0,8	30/1,5	51/0,9	57/0,8	57/0,8
	Efficiency	0	0	360	0	0	0
	Training	570	57	30	0	0	0
	Equity	114	114	60	102	114	114
80/1	Competence	71/0,6	61/0,7	61/0,7	35/1,2	33/1,3	43/1
	Efficiency	0	0	0	0	759	0
	Training	1562	61	0	0	0	0
	Equity	213	183	183	175	165	172
700/5	Competence	39/1,3	63/0,8	36/1,4	36/1,4	85/0,6	63/0,8
	Efficiency	0	0	8640	0	0	0
	Training	0	0	36	0	20315	0
	Equity	1911	1953	1908	1908	1955	1953

Table 8 The human resource workload w.r.t to the human resource strategy in a 6-resource workshop.

We remark from Table 8 that when we adopt a strategy based on efficiency, only the resource with high competence is involved. However, when we adopt a strategy based on training, only resources with the lowest competence are involved.

5 Conclusion and future work

In this work, we studied the single machine problem with flexible maintenance activities. Each maintenance activity must be treated by a human resource characterized by a competence level and a timetabling determining his availabilities in the workshop.

The assignment of human resources is subject to some rules related the workshop assignment strategy which could favor either the efficiency, the equity or training. Efficiency

is favored when human resources with high competences are assigned in priority. The equity strategy is adopted when all resources are assigned with the same workload. Finally, the training strategy means that the human resources having low competences are favored in order to improve their competencies.

To solve the problem, we first proposed a mathematical model for the studied problem implemented in Cplex to solve optimally some of its small size instances. Secondly, an adapted guided local search (IGLS) characterized by several local searches is provided. These local search methods are applied according to the highest utility. Moreover, to introduce more intensification and diversification into the search process performed by IGLS we incorporated a post-optimization procedure and a restart scheme. The IGLS is introduced to deal in practice with relatively large size instances.

The IGLS is implemented and compared to the exact method Cplex on small instances of the problem. Cplex is used as a reference method to check the solution quality of IGLS while IGLS is intended to solve large instances of the problem.

The Experimental results show the hardness of solving optimally the studied problem. Indeed, Cplex is only able to find solution to instances with less than 13 jobs in one hour time limit.

We can see that IGLS find good solutions in a reasonable CPU time for relatively large instances of the problem. We also studied three different human resource assignment strategies and showed their impact on the objective functions.

In future, we aim to introduce the learning effect to the human resources allowing them to improve their competence level. Furthermore, we will study this problem as a multi objective one.

References

- Adiri, I., Frostig, E., and Kan, A.-H.-G. R. (1989). Single machine flow-time scheduling with a single breakdown. *Acta Inf*, 26:679–696.
- Alsheddy, A. and Tsang, E.-P. (2011). Empowerment scheduling for a field workforce. *Journal of Scheduling*, 14:639–654.
- Bennour, M. and Crestani, D. (2007). Using competencies in performance estimation: From the activity to the process. *Computers in Industry*, 58(2):151–163.
- Bouzidi-Hassini, S., Benbouzid-Sitayeb, F., Marmier, F., and Rabahi, M. (2015). Considering human resource constraints for real joint production and maintenance schedules. *Computers & Industrial Engineering*, 90:197–211.
- Chen, J.-S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of the Operational Research*, 190:90–102.
- Ciro, G.-C., Dugardin, F., Yalaoui, F., and Kelly, R. (2015). 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.

- Cui, W.-W. and Lu, Z. (2014). Integrated production scheduling and periodic maintenances on a single machine with release dates. *IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei, Taiwan, 197:53–58.
- Du, J. and Leung, J.-Y.-T. (1990). Minimizing total tardiness on one machine is np-hard. *Mathematics of Operations Research*, 15(3):483–495.
- Graham, R.-L., Lawler, E.-L., Lenstra, J.-K., and Kan, A.-R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326.
- Guyon, O., Lemaire, P., Pinson, E., and Rivreau, D. (2014). Solving an integrated job-shop problem with human resource constraints. *Annals of Operations Research*, 1(213):147–171.
- Gérard, M., Clautiaux, F., and Sadykovde, R. (2016). Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce. *European Journal of Operational Research*, 3(252):1019–1030.
- Ladj, A., Tayeb, F. B.-S., and Varnier, C. (2016). An integrated prognostic based hybrid genetic-immune algorithm for scheduling jobs and predictive maintenance. *IEEE Congress on Evolutionary Computation (CEC)*, pages 2083–2089.
- Lee, C.-Y. and Liman, S.-D. (1992). Single machine flow-time scheduling with scheduled maintenance. *Acta Informatica*, 29:375–382.
- Lee, J.-Y. and Kim, Y.-D. (2012). Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance. *Computers I& Operations Research*, 39(9):2196–2205.
- Liao, C.-L. and Chen, W.-J. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, 30(9):1335–1347.
- Lilly, M.-T., Emovon, I., Ogaji, S.-O.-T., and Probert, S.-D. (2007). Four-day service-staff work-week in order to complete maintenance operations more effectively in a nigerian power-generating station. *Applied Energy*, 84(10):1044–1055.
- Liu, M., Wang, S., Chu, C., and Chu, F. (2015). 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.
- Low, C., Ji, M., Hsu, C.-J., and Su, C.-T. (2010). Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance. *Applied Mathematical Modelling*, 34(2):334–342.
- Luo, A., Cheng, T.-C.-E., and Ji, M. (2015). Single-machine scheduling with a variable maintenance activity. *Computers and Industrial Engineering*, 79:168–174.
- Ma, Y., Chu, C., and Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58(2):199–211.
- Marmier, F., Varnier, C., and Zerhouni, N. (2009). Proactive, dynamic and multi-criteria scheduling of maintenance activities. *International Journal of Production Research*, 47(8):2185–2201.

- Mashkani, O. and Moslehi, G. (2016). Minimising the total completion time in a single machine scheduling problem under bimodal flexible periodic availability constraints. *International Journal of Computer Integrated Manufacturing*, 29(3):323–341.
- Nagata, Y. and I.Ono (2018). Guided local search with iterative ejections of bottleneck operations for the job shop scheduling problem. *Computers & Operations Research*, 90:60–71.
- Sabar, M., Montreuil, B., and Frayret, J.-M. (2009a). Competency and preference based personnel scheduling in large assembly lines. *International Journal of Computer Integrated Manufacturing*, 4(21):468–479.
- Sabar, M., Montreuil, B., and Frayret, J.-M. (2009b). A multi-agent-based approach for personnel scheduling in assembly centers. *Engineering Applications of Artificial Intelligence*, 22(7):1080–1088.
- Safaei, N., Banjevic, D., and Jardine, A.-K.-S. (2011). Workforce-constrained maintenance scheduling for military aircraft fleet: a case study. *Ann Oper Res*, 186:295–316.
- Sbihi, M. and Varnier, C. (2008). Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness. *Computers & Industrial Engineering*, 55(4):830–840.
- Schmidt, G. (2000). Scheduling with limited machine availability. *European Journal of Operational Research*, 121(1):1–15.
- Tang, H., Miller-Hooks, E., and Tomastik, R. (2007). Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research*, 43(5):591–609.
- Tischer, I. and Carrión, A. (2008). An application of guided local search : the scheduling of sugar cane harvest in colombia. *Engeneering & automation problems*, 6:60–65.
- Touat, M., Bouzidi-Hassini, S., Benbouzid-Sitayeb, F., and Benhamou, B. (2017a). 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.
- Touat, M., Tayeb, F. B.-S., and Benhamou, B. (2018). An effective heuristic for the single-machine scheduling problem with flexible maintenance under human resource constraints. *Procedia Computer Science*, 126:1395–1404.
- Touat, M., Tayeb, F. B.-S., Benhamou, B., Sadeg, L., Aklil, S., and Karaoui, M. (2019). An integrated guided local search considering human resource constraints for the single-machine scheduling problem with preventive maintenance. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3799–3804. IEEE.
- Touat, M., Tayeb, F. B.-S., Bouzidi-Hassini, S., and Benhamou, B. (2017b). 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.

- Voudouris, C. and E.Tsang (1995a). 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.
- Voudouris, C. and E.Tsang (1995b). Guided local search. *Department of Computer Science, University of Essex*, Technical Report:CSM-247.
- Yang, D.-L., Hung, C.-L., Hsu, C.-J., and Chern, M.-S. (2002). Minimizing the makespan in a single machine scheduling problem with a flexible maintenance. *Journal of the Chinese Institute of Industrial Engineers*, 19:63-66.
- Yang, S.-L., Ma, Y., Xu, D.-L., and Yang, J.-B. (2011). Minimizing total completion time on a single machine with a flexible maintenance activity. *Computers and Operations Research*, 38(4):755-770.
- Yazdani, M., Khalili, S.-M., Babagolzadeh, M., and Jolai, F. (2017). 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(1):46-59.
- Yurtkuran, A., Yagmahan, B., and Emel, E. (2018). 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.