A Fuzzy Genetic Algorithm for the Integrated Single-Machine Scheduling with Flexible Maintenance Problem under Human Resource Constraints

Abstract

This research focuses on the problem of scheduling jobs on a single machine that requires flexible maintenance under human resource constraint. A fuzzy genetic algorithm that integrates production, maintenance and human resources constraints is developed. This algorithm uses fuzzy logic to deal with uncertainties. Experiments show that the consideration of human resource constraints and uncertainties in the integrated and proactive scheduling allows proposing realistic and applicable solutions.

Keywords

Scheduling; Flexible maintenance, Fuzzy logic, Genetic algorithm, Human resource constraints, Single machine problem.

1 Introduction

Recently, realizing the inherent conflicts between production and preventive maintenance (PM), scheduling problems integrated with PM have gained much attention among researchers. This integration is justified by expected cost savings and better resource utilization. PM planning might delay the production schedule, but will reduce expected number of machine failures. In general, maintenance is integrated through two different ways. The first way is periodic maintenance performed periodically with a fixed time interval. The second way is flexible maintenance, where the maintenance intervals are not fixed or the starting times of such intervals are supposed to be flexible. These problems are proved to be NP-hard [1]. Among the recent works, we can mention all of Liao and Chen [2], Ji *et al.* [3], Low et al. [4], Chen [5], Lee and Kim [6], Sbihi and Varnier [7], Cui *et al.* [8].

The majority of these researches assume a perfect environment in terms of resource availability (e.g., machines, tools, people) and consider production jobs as maintenance ones [9, 10] although they do not have the same properties. Generally, maintenance activities involve human resources with different skills level and availabilities and impact directly the generated schedules. Availability in turn, assesses the feasibility or not of the resulting schedules. Scheduling under human resource constraints have been studied for production and maintenance activities separately, but rarely for integrated production and maintenance scheduling problem [11].

Besides, scheduling parameters are not always precise due to both human and machine resource factors. Classical approaches, within a deterministic scheduling theory, relying on precise data might are not suitable to deal with uncertain scenarios. Scheduling under uncertainties addresses this need and aims to take into consideration the data imperfection. To model data imperfection, fuzzy logic [12] has been increasingly used to capture and process imprecise and uncertain information [13, 14]. Imprecise scheduling parameters have been represented as fuzzy numbers.

Literature reports many optimization based heuristics for the single machine scheduling problem. Most of these methods seldom studied the single machine scheduling problem with flexible maintenance (see e.g. [15-18].). Moreover, no work considers uncertainties related to both production and maintenance simultaneously. Indeed, these two activities are too closed and it makes more sense to consider uncertainties for both of them in order to propose more realistic scheduling. Furthermore, competence and availability constraints impact maintenance activities' processing times. That is, we can achieve an optimal processing time by allocating a maintenance activity to an available and efficient human operator.

In order to response to the shortcomings identified in the literature, we propose in this work a genetic algorithm (GA) to solve the single-machine scheduling problem, taking into account both production and flexible maintenance activities modeled through fuzzy time intervals. In the tackled problem, the maintenance activities are subject to human resources constraints, in the sense that every maintenance requires a staff characterized by specific competences and some availability intervals. The objective is to optimize a common objective function which takes into account both the maintenance and the production criteria.

The rest of the paper is organized as follows: The problem description is provided in Section 2. The proposed algorithm 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 assumptions

We consider here, a single-machine scheduling problem with fuzzy processing times which does not operate continuously. The problem may be described as follows. Let $J = \{J_1, J_2, ...J_N\}$ be a set of N jobs to be processed by a single machine. Each job J_i is available at time zero, and has a fuzzy processing time p_i and a fuzzy completion time c_i Preemption is not allowed and the machine can handle at most one job at a time and cannot stand idle until the last job is finished. According to the classification given in [19] extended here to include resource constraints, this problem is denoted by $1/nprmp - pfm/C_{max}$. It has to be mentioned that the addressed problem is *NP-hard* in the strong sense [5].

Besides, in order to maintain its high availability, machine maintenance must be undertake. In this paper, we consider a single flexible maintenance M with multiple occurrences M_k, $k=1,..Nb\_Occ$, following a given period T and a fuzzy duration referred to as $p_k^{'}$. A maintenance M_k must be completed within a time window, noted $TI_k=[Tmin_k, Tmax_k]$ (Fig.1), when the maintenance activity is most cost effective and before the equipment loses optimum performance. However, it can be planed before $Tmin_k$ and it is considered in advance (this earliness is noted $E_k^{'}$), or after $Tmax_k$ and it is considered late (this tardiness is noted $T_k^{'}$). We assume that the first time window is arranged in advance. The k^{th} time window depends on the completion time of the $(k\text{-}1)^{th}$ maintenance occurrence M_k.

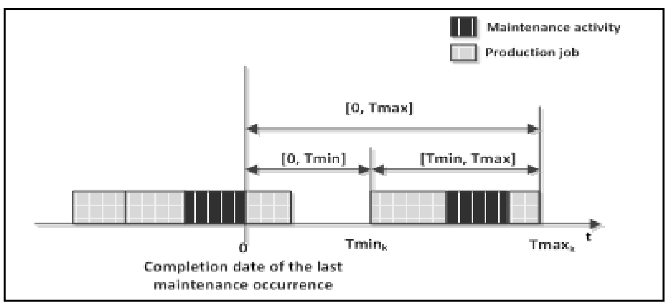

Figure 1. Maintenance activity tolerance interval

The human competence or qualification required for a maintenance activity is considered and expressed by *Com*. The maintenance service is composed of R human resources (*HR*). Each HR_l $_{(l=1,..R)}$ is characterized by a competence level $Comp_l$ allowing him to execute a maintenance activity with a duration ph_l . Moreover, each HR_l has a timetabling which determines its availability. Consequently, we specify for each HR_l a set AI_{lj} $_{(j=1,..m)}$ of m availability intervals $AI_l = \{[LB_{11}, UB_{11}], .. [LB_{lm}, UB_{lm}]\}$. The symbols LB and UB denote respectively, the lower and the upper bounds of an availability interval AI.

The imprecision of processing times of both production and maintenance activities are modeled using the Triangular fuzzy numbers (TFNs) (Figure 2.a). Due to their computational simplicity, TFNs are more suitable in several applications. They are useful in promoting presentation and information processing in a fuzzy environment and successfully applied in various domains. The execution time and the completion time of each production or maintenance operation are also fuzzy (see Figure 2.b). Moreover, advances $E_k^{'}$ and delays $T_k^{'}$ of a maintenance occurrence M_k will be at their turn fuzzy with triangular membership function.

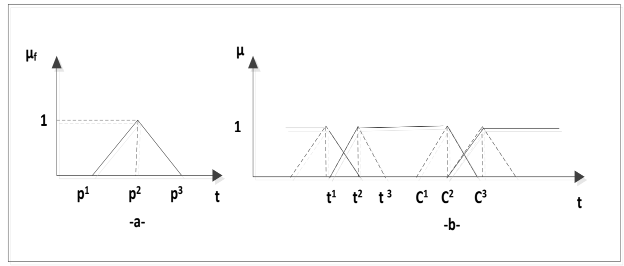

Figure 2. Fuzzy modeling of production and maintenance activities processing times and execution intervals

The global objective is to find a permutation of N production jobs that minimizes the C_{max}, while inserting maintenance activities. The latter are planned by taking into account the human resource constraints which are the availability and the competence level. The production objective function f_p is to find a sequence of operations of N jobs that minimizes the completion time of the last job on the last machine. f_p is computed as follows:

$$f_p = c_N \quad (1)$$

The maintenance objective is to minimize the total earliness/tardiness of all the maintenance activities with respect to the pre-specified maintenance intervals. The value of f_m is computed as follows (eq.2):

$$f_m = \sum_{k=1}^{Nb\_Occ} \left(E_k^{'} + T_k^{'}\right) \quad (2)$$

To optimize both the previous criteria, we consider the function f defined as follows (eq.3):

$$f = f_p + f_m \quad (3)$$

3 The proposed fuzzy integrated genetic algorithm

We develop here, a fuzzy genetic algorithm for the integrated production and maintenance schedules where the availability and the competence constraints of the human interveners that are in charge of maintenance are taken into account. The processing times of production jobs and maintenance activities are fuzzy modeled. The most remarkable characteristics of this fuzzy genetic algorithm are the following:

- The use of an integrated representation of production and maintenance data which embeds human resource constraints;
- The use of the dispatching rules to generate a portion of the initial population;
- The consideration of uncertainties related to both the production and the maintenance activities;
- The use of a readjustment process applied after a crossover and a mutation to ensure the feasibility of the crossed and muted chromosomes;
- The use of a fuzzy global objective function that takes into account both the production and the maintenance criteria.

The complete procedures of the proposed fuzzy integrated GA (hereafter FIGA) are depicted in Algorithm 1.

/*Initialization*/
Step 1. *Set the values of the population size (Psize); the crossover probability (P_c) and the mutation probability (P_m) and the termination conditions(Nb_{Gen}, $Nb_{Improve}$)*
Begin
Create a population of Psize chromosomes;
j=0;
/*Selection, Crossover and Mutation*/
Step 2.
j=j+1;
Evaluate each solution in the current population
Select pairs of chromosomes from a current population according to the selection probability to generate Parentset.
Step 3.
Select a pair of solutions P_1, P_2 from parentset
Perform crossover on P_1 and P_2 to obtain E_1, E_2 with probability P_c
Step 4. *Perform readjustment on E_1 and E_2*
Step 5. *Perform mutation on E_1 and E_2 to obtain E_3 and E_4 with probability P_m*
Step 6. *Perform readjustment on E_3 and E_4*
Step 7. Replacement
Evaluate each solution in the current population
Update the population to avoid entrapment in local optima and algorithm stall
Step 8. *If a prespecified stopping condition is satisfied, stop this algorithm. Otherwise, return to step 2.*
EndFor
End.

Algorithm 1. FIGA steps procedure

FIGA contains two major processes which are “Genetic search” and “Readjustment”. Genetic search is implemented through genetic operators and directed by selection pressure (Steps 1-4). The most remarkable characteristics of FIGA is the “Readjustment” process (Step 5) which overrides the modifications applied by the crossover and mutation operators in order to restore human resource assignment feasibility. The mechanisms and operators of FIGA are described below.

3.1 The Encoding scheme and the fitness function

In FIGA, Each chromosome is expressed by an integrated sequence of jobs and PM activities. Moreover, to take into account the human resource constraints, we add to the maintenance genes the execution time $t_k^{'}$ and the identifiers of the human resources *(HR_l)* assigned to execute the maintenance occurrence M_k. The length of a chromosome is then equal to the total number of activities including production and maintenance (eq.4). Production jobs are represented by the genes 1 to N and the maintenance occurrences are represented by the genes *N+1* to *LN+NB_Occ*.

Example 1: Let us consider an example with 2 maintenance occurrences and 4 production jobs. Production jobs are encoded from 1 to 4 (Fig.3) and the maintenance occurrences are encoded by the numbers 5 and 6. Their execution times $t_5^{'}$ and $t_6^{'}$ are respectively *(27 30 33)* and *(76 85 94)*. Human resources affected to M_1 and M_2 are respectively HR_2 and HR_1.

1	3	**5,(27 30 33),HR_2**	4	2	**6,(76 85 94),HR_1**

Figure 3. Example of a chromosome encoding

The objective function is to minimize f . Therefore, the definition of fitness function is just the reciprocal of f (section 2.4).

3.2 Initial population generation

Randomly generating an initial population is commonly applied in genetic algorithm. The reason is that it may leads to a population diversification which is in favor of the evolution. However, random individuals often have poor fitness and thus may slow down the convergence. To solve the problem, a feasible approach is to employ an effective constructive heuristic [20]. This approach not only provides a good deal of diversity in the population, but also performs better than the approach where the initial population consisted of only randomly generated solutions. Therefore, in this work, we propose a two-step initialization procedure as follows:

Step1. In this step, we generate two parts of the initial population. The first part (the largest one) consists of randomly generated production sequences. The second part is made up of two production sequences generated using two dispatching rules[1][19]: (1) *SPT*; and (2) *LPT.*

Step2. In this step, PM activities are inserted into the production sequences that are generated in Step1. The human resource constraints are also considered at this stage. The M_k occurrences are planned according to human resource availability. If there are many possibilities for planning M_k, the one that best minimizes f_m is selected. The maintenance activity insertion process is outlined in Algorithm 2.

[1] (1) SPT: Shortest processing time; (2) LPT: Longest processing time

Let M a maintenance activity with processing time p' and tolerance interval TI=[Tmin,Tmax]

Begin

For *each occurrence* M_k *of* M

If *There is an available resource* HR_l *with an availability interval* AI_{lj}

Then *Schedule* M_k *in its tolerance interval*

Else *Let* AI^1 *and* AI^2 *the nearest availability intervals, respectively just before and just after* TI_k

Compute the partial maintenance objective function f_m *when inserting* M_k *at* AI^1 ***and*** AI^2 *and choose the one optimizing it the best*

Update the maintenance gene (execution time and assigned resource)

Endfor

End.

Algorithm 2. Maintenance insertion procedure

3.3 The population improvement

In FIGA, the parents are chosen according to a tournament selection. Chromosome recombination is performed by using the two-point crossover (TP version I) of Murata *et al.* [21]. This is, to overcome the chromosome illegality which most likely occur with the classical two-point crossover operator. Finally, the swap mutation [22] operator is used in our method. In the mutation process, the genes of the two random positions of production are swapped.

3.4 Readjustment

After a crossover or a mutation process, the maintenance execution intervals of activities could be modified. This may lead to human resource unavailability in the new execution interval. Therefore, readjustment step is introduced to make sure that the maintenance activity modifications due to crossover and/or mutation steps are still compatible with human resource availability intervals. In case of human resource unavailability, a readjustment (i.e. rescheduling) of jobs is performed so that the previous compatibility is restored. Algorithm 3 describes the readjustment process.

Let M_k *a maintenance activity occurrence with processing time* p'_k *, execution interval* EI'_k *and affected resource* HR_l

Begin

For *each occurrence* M_k ***of M***

If *the affected resource* HR_l *to* M_k *is available in* EI'_k ***Then*** *Update the resource availability*

Else If *there's another available human resource in* EI'_k

Then *Affect it to* M_k

Else *Let* AI^1 *and* AI^2 *the nearest availability intervals, respectively just before and just after* EI'_k

Compute partial global objective function f *when inserting* M_k *at* AI^1 *and* AI^2 *and choose the one optimizing best* f *Update resources availabilities*

EndFor

Update the maintenance gene (execution time and assigned resource)

End

Algorithm 3. Readjustment procedure

3.5. Population renewal

The final step of FIGA is the update of the current population to avoid entrapment in local optima and algorithm stall. The offspring are evaluated (global objective function f as defined in section 2.4) and the best performing ones are included in the new population. The rest of solutions are selected from the parent population with an elitist strategy. This allows a certain percentage of chromosomes with poor fitness values to be included in the population so that there will be a better chance of breaking out of local minima.

3.6 Termination criteria

In traditional genetic algorithms, either the computation time or the number of generations is selected as a termination criterion. For FIGA, we consider both of them. That is, if a solution cannot be improved any more in consecutive $NB_{Improve}$ generations, the algorithm terminates. However, this criterion alone could be very time-consuming for large problems. Therefore, a maximum number of generations is chosen as the second termination criterion for FIGA. In other words, FIGA terminates when at least one of the two criteria is met.

4 Experiment and Discussion

In this section, we present some experimental results to show the effectiveness of the new proposed fuzzy GA for the integrated single-machine and PM scheduling problem. We implemented the algorithms in *JAVA* and ran them on a personal computer with an *Intel Core i5 1.70 GOHz CPU* and *6 Go RAM* memory under *Windows 7* operating system. In what following, we will first describe how test data are generated. Then, we analyse the effectiveness of the proposed method.

4.1 Data generation and parameters setting

To our knowledge, there is no instance for the single machine scheduling problem with flexible maintenance under human resources constraints. Therefore, the instances used for evaluating FIGA are new and non-standard ones. We used two types of data. The first one is related to production and maintenance, while the second one is related to the human resource part. For the first type, the experiments have been conducted on 10 benchmarks with N arranging from 20 to 200. It is further divided into ten sets with 10 instances in each set. The processing times p_i of production jobs are generated from a uniform distribution u[1, 99]. The maintenance activity processing time p' is assumed to be of medium duration, it is given by the average of job processing times as $p' = \frac{1}{N}\sum_{i=1}^{N} p_i$. Both processing times of production and maintenance activities are fuzzy modeled and computed using a widening coefficient noted *wf* which represents the distance between the most probable value of the fuzzy measure and its maximum or minimum values. The optimal preventive maintenance period T is $0.25x\sum_{i=1}^{N} p_i$ and the tolerated advance/tardiness of maintenance activities is $\Delta T = 0.05 \times T$. For the second type, due to the lack of the benchmark instances on human resources, the availability intervals of human resources are generated according to the production horizon. All existing skills depend on maintenance activities type. Thereby, for each instance of integrated production and maintenance benchmark, a human resources instance is specified.

We consider only one type of PM activity with multiple occurrences and two human resources (HR_1, HR_2) with respectively two competence levels ($Comp_1$, $Comp_2$). $Comp_1$ and $Comp_2$ are generated from a uniform distribution u]0.2[. In order to execute an occurrence M_k by a HR, two durations are possible (ph_1, ph_2). Such as: $ph_1 = p' \tilde{\times} Comp_1$ and $ph_2 = p' \tilde{\times} Comp_2$. The set AI_l of availability intervals with $l \in \{1,2\}$ is generated according to algorithm 4.

Begin

Step 1. *Compute the production horizon* $PH = \sum_{i=1}^{N} p_i + \sum_{k=1}^{NB\_Occ} p'_k$

Step 2. *Divide PH to Nb_Occ sub-horizons with HT size.*

Step 3. *Generate in each sub-horizon* $[D_k,F_k]$ $k \in [1,Nb\_Occ]$ *an* $AI_{lk}=[LB_{lk}, UB_{lk}]$

- *The lower bound* LB_{lk} *is randomly generated in* $[D_k,D]$ *with* $D=D_k+((HT/2)-\mu*(p'/2))$
- *The upper bound* UB_{lk} *is randomly generated in* $[F,F_k]$ *with* $F=(F_k-(HT/2))+\mu*(p'/2)$

End

Algorithm 4. Availability intervals generation

In our problem, it is insufficient and inappropriate to handle fuzzy data with only the basic fuzzy operators introduced in [23]. Indeed, it is necessary to explain how the objective functions are compared. Therefore, we introduce a new fuzzy operator Obj defined as follows:

$$obj(A,B) = \begin{cases} A & if\ (D(A) < D(B)) \vee (D(A) = D(B)) \wedge (a^2 < b^2)) \\ B & if\ (D(B) < D(A)) \vee (D(A) = D(B)) \wedge (b^2 < a^2)) \end{cases}$$

Where $D(A)$ (respectively $D(B)$) is the distance of A (respectively of B) from the abscissa 0 that is computed by

$$D(A) = \frac{a_1 + a^3}{2}$$

Since the objective is to minimize (f), the operator returns the minimal value. The distance cannot be used to compare values. Indeed, two values could have a same distance D, but could be not equivalent. For this reason, we adopted the method proposed by [24] to provide the comparison fuzzy operator Obj according to first the distance D and then the medium values of fuzzy numbers as shown in figure 5.

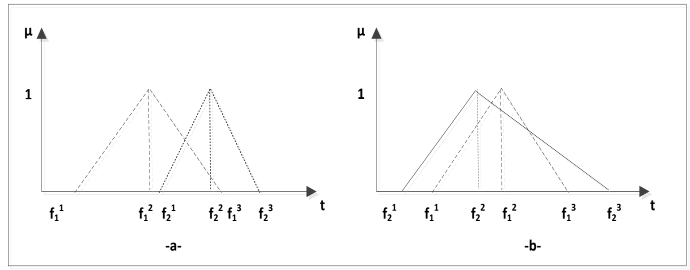

Figure 5. The fuzzy objective function comparison

We proceed with the analysis of the results of pilot experiments generated randomly from a factorial design, and then we fix every factor to the most interesting level. The complete details are not reported for the sake of concise presentation. We set: P_{Size} = 150; Nb_{Gen}=500; $Nb_{improve}$=20; P_{Cross} = 0.7; P_{Mut} = 0.05; wf = 0.1; α = 20%.

4.2 Experiment 2: Performance analysis of FIGA

Since no optimal solutions are known for the studied problem, we compare our fuzzy integrated GA results against the lower bound c_{low} proposed by Chen [5] for the same problem without human resources consideration. For our problem, we adapted this lower bound to the fuzzy modeled of both processing times of production and maintenance activities, using the basic operators introduced in [23], with the assumption that human resources are available when needed. c_{low} can be estimated as $\sum_{i=1}^{N} p_i + \left[(\sum_{i=1}^{N} p_i \div T) - 1 \right] \times p_i$. The first term of c_{low} is the total processing time of the production jobs, and the second is the minimum total maintenance time. It is clearly that there is no solution less than the lower bound.

The relative percentage deviation (RPD) is used as an index to evaluate the performance of FIGA. This index is described as $\frac{D(f) - D(c_{low})}{D(c_{low})} \times 100$. $D(f)$ is the distance of the global objectif function defined in section 2 and $D(c_{low})$ the distance of the newly proposed fuzzy lower bound.

In order to demonstrate the performance of the proposed FIGA algorithm, an experiment that evaluates the performance of FIGA in comparison with FIGA-R is performed. FIGA-R is similar to FIGA without the readjustment process. In FIGA-R, after a crossover or mutation operation if the new maintenance planning intervals don't satisfy the human resource availabilities, the resulting chromosome is not retained within the population. We aim here to measure the impact of the readjustment process on the global objective function f. The experimental results are summarized in Table 1.

The results shown in Table 1 reveal that FIGA yields a small deviation from the lower bound. In worst cases, the global objective function f increase by less than 10%.

Moreover, in several cases, this deviation is less than 7%. That confirms the efficiency of our GA to generate best solutions for all problem instances. This is argued by the correct parameters setting and the choice of appropriate available human resources. Moreover, the performance of FIGA is better than FIGA-R. Thus, one can conclude that for the addressed problems the readjustment process performs well.

Table 1. Performance comparison of FIGA

N	c_{low}	***FIGA***	***RPD***	***FIGA-R***	***RPD***
20	1039 1144 1250	1191 1202 1214	7.69%	1244 1258 1272	9.92%
40	1947 2146 2345	*2211 2226 2241*	6.76%	2283 2299 2315	7.13%
60	2834 3122 3411	3219 3230 3242	8.18%	3410 3429 3448	9.82%
80	3824 4214 4605	4384 4399 4413	8.08%	4549 4557 4586	8.38%
100	4592 5058 5525	5251 5270 5289	8.97%	5520 5544 5569	9.61%
120	5627 6204 6780	6340 6365 6391	6.78%	6630 6661 6692	7.37%
140	6454 7113 7771	7314 7333 7351	6.17%	7553 7576 7599	6.52%
160	7465 8228 8991	8608 8630 8653	9.69%	9032 9063 9094	10.15%
180	8211 9047 9884	9289 9312 9335	6.12%	9602 9631 9660	6.45%
200	9146 10081 11017	10527 10545 10563	7.61%	10846 10867 10888	7.79%

4.2 Experiment 2: Robustness analysis of FIGA

A workshop is subject to many disruptions. That is why the predicted established plan has to take into their occurrences during exploitation. The plan has to be as robust as possible: it has to guarantee the respect of customers' objectives as long as possible even in the presence of disturbances.

To the best of our knowledge, there is no robustness procedure especially designed for the single-machine problem with makespan measure. In this paper, we simulate disruptions (machine breakdown, human resource defection, …) and we observe the behavior of our system.

We assume that the machine experiences a single disruption in a given scheduling period. Its arrival date can be at the beginning or the middle of the scheduling horizon in order to observe the impact of this event on the schedule's performance. Because of this restrictive assumption, this approach can be used within a *continuous rescheduling scheme*, where the system is rescheduled from scratch after every disruption. Thus, when a disruption occurs the remaining activities (production or maintenance) in the schedule are right shifted a sufficient amount to just accommodate the repair duration. We assume that the repair time distribution is uniform [1,50]. We recorded the *RDP* for both beginning and middle disruption arrival. Table 2 presents a summary of the results. In term of robustness, whatever disruption arrival date at the beginning or the middle of the schedule, FIGA yields a very small deviation from the lower bound. Except for the case of 20 jobs, f increase by less than 0.9%. In several cases, this deviation is less than 0.1%. Moreover, in 80% cases, we observed slightly better results in the 'middle' case. This is due to the fact that a disruption at the beginning would introduce a delay in the scheduling of all the activities. We noticed that more closer are 'beginning' and 'middle', greater is the disturbance.

Table 2. Performance comparison of FIGA with disruptions

N	f	***RPD Beginning***	***RDP Middle***
20	1428 1441 1454	2,50%	4,44%
40	2062 2084 2106	0,72%	0,91%
60	3396 3415 3434	0,50%	2,25%
80	4687 4697 4707	0,60%	0,62%
100	6482 6492 6502	0,08%	0,08%
120	7288 7303 7318	0,88%	0,89%
140	7241 7249 7257	0,12%	0,15%
160	8119 8131 8143	0,12%	0,15%
180	9209 9281 9353	0,08%	1,05%
200	10481 10493 10505	0,04%	0,09%

5 Conclusion

In this paper, we treat the scheduling problem of production and maintenance activities under competence and availability human resource constraints in order to optimize a common objective function in a single machine shop floor. Processing time of both production and maintenance activities are imprecise and are modeled using fuzzy logic. A fuzzy integrated genetic algorithm is developed to generate a population of production and maintenance activities schedules considering the human resource constraints where fuzzy data manipulation has been showed. For numerical experiments a human resource benchmarks generation method have been presented. Then, two studies which analyse the performance of the proposed FIGA in terms of objective function optimization and robustness face to disruptions have been presented. The first one showed that a small *wf* offers the best results. The second shows that the use fuzzy logic to deal with uncertainties allows absorbing disruptions that may occur in the workshop.

In future research we will use human resources' characteristic (cost,..) to define a new lower bound for this problem. Moreover, we can generalize the considered problem to a more complex one.

References

[1] C.-Y. Lee and S. D. Liman, "Single machine flow-time scheduling with scheduled maintenance," *Acta Informatica*, vol. 29, pp. 375-382, 1992.

[2] C. Liao and W. Chen, "Single-machine scheduling with periodic maintenance and nonresumable jobs," *Computers & Operations Research*, vol. 30, pp. 1335-1347, 2003.

[3] M. Ji, Y. He, and T. E. Cheng, "Single-machine scheduling with periodic maintenance to minimize makespan," *Computers & operations research*, vol. 34, pp. 1764-1770, 2007.

[4] C. Low, C.-J. Hsu, and C.-T. Su, "Minimizing the makespan with an availability constraint on a single machine under simple linear deterioration," *Computers & Mathematics with Applications*, vol. 56, pp. 257-265, 2008.

[5] J.-S. Chen, "Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan," *European Journal of Operational Research*, vol. 190, pp. 90-102, 2008.

[6] J.-Y. Lee and Y.-D. Kim, "Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance," *Computers & Operations Research*, vol. 39, pp. 2196-2205, 2012.

[7] M. Sbihi and C. Varnier, "Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness," *Computers & Industrial Engineering*, vol. 55, pp. 830-840, 2008.

[8] W.-W. Cui, Z. Lu, and E. Pan, "Integrated production scheduling and maintenance policy for robustness in a single machine," *Computers & Operations Research*, vol. 47, pp. 81-91, 2014.

[9] B. Naderi, M. Zandieh, and M. Aminnayeri, "Incorporating periodic preventive maintenance into flexible flowshop scheduling problems," *Applied Soft Computing*, vol. 11, pp. 2094-2101, 2011.

[10] F. Zammori, M. Braglia, and D. Castellano, "Harmony search algorithm for single-machine scheduling problem with planned maintenance," *Computers & Industrial Engineering*, vol. 76, pp. 333-346, 2014.

[11] S. Bouzidi-Hassini, F. B.-S. Tayeb, F. Marmier, and M. Rabahi, "Considering human resource constraints for real joint production and maintenance schedules," *Computers & Industrial Engineering*, vol. 90, pp. 197-211, 2015.

[12] V. Dumitru and F. Luban, "Membership functions, some mathematical programming models and production scheduling," *Fuzzy Sets and Systems*, vol. 8, pp. 19-33, 1982.

[13] S. Petrovic, D. Petrovic, and E. Burke, "Fuzzy logic-based production scheduling and rescheduling in the presence of uncertainty," in *Planning Production and Inventories in the Extended Enterprise*, ed: Springer, 2011, pp. 531-562.

[14] S. Chanas and A. Kasperski, "On two single machine scheduling problems with fuzzy processing times and fuzzy due dates," *European Journal of Operational Research*, vol. 147, pp. 281-296, 2003.

[15] N. Sortrakul and C. R. Cassady, "Genetic algorithms for total weighted expected tardiness integrated preventive maintenance planning and production scheduling for a single machine," *Journal of Quality in Maintenance Engineering*, vol. 13, pp. 49-61, 2007.

[16] M. Rebai, I. Kacem, and K. H. Adjallah, "Earliness–tardiness minimization on a single machine to schedule preventive maintenance tasks: metaheuristic and exact methods," *Journal of Intelligent Manufacturing*, vol. 23, pp. 1207-1224, 2012.

[17] B. S. Kim and Y. Ozturkoglu, "Scheduling a single machine with multiple preventive maintenance activities and position-based deteriorations using genetic algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 67, pp. 1127-1137, 2013.

[18] J. Yulan, J. Zuhua, and H. Wenrui, "Multi-objective integrated optimization research on preventive maintenance planning and production scheduling for a single machine," *The International Journal of Advanced Manufacturing Technology*, vol. 39, pp. 954-964, 2008.

[19] M. L. Pinedo, "Scheduling: theory, algorithms, and systems," ed: Springer Science & Business Media, 2012.

[20] V. Kodaganallur, A. K. Sen, and S. Mitra, "Application of graph search and genetic algorithms for the single machine scheduling problem with sequence-dependent setup times and quadratic penalty function of completion times," *Computers & Industrial Engineering*, vol. 67, pp. 10-19, 2014.

[21] T. Murata, H. Ishibuchi, and H. Tanaka, "Genetic algorithms for flowshop scheduling problems," *Computers & Industrial Engineering*, vol. 30, pp. 1061-1071, 1996.

[22] R. Ruiz, C. Maroto, and J. Alcaraz, "Two new robust genetic algorithms for the flowshop scheduling problem," *Omega*, vol. 34, pp. 461-476, 2006.

[23] S.-M. Chen, "Fuzzy system reliability analysis using fuzzy number arithmetic operations," *Fuzzy Sets and Systems*, vol. 69, p. 231, 1995.

[24] A. Kaufmann, "Introduction a la theorie des sous-ensembles flous a l'usage des ingenieurs (fuz," 1978.