academicJournals

Vol. 5(6), pp. 112-122, September 2013
DOI 10.5897/JMER2013.0275
ISSN 2141-2383 © 2013 Academic Journals
http://www.academicjournals.org/JMER

Journal of Mechanical Engineering Research

Full Length Research Paper

An efficient hybrid approach based on multi agent system and emergence method for the integration of systematic preventive maintenance policies in hybrid flow-shop scheduling to minimize makespan

Adel Abdelhadi and L. Hayet Mouss

Laboratory Automation and Production, Department of Industrial Engineering, University of BATNA, Rue Chahid Boukhlouf 05000 BATNA, ALGERIA.

Accepted 13 September, 2013.

This paper proposes a novel hybrid algorithm for the integration of systematic preventive maintenance policies in hybrid flow shop scheduling to minimize makespan. We have implemented a problem-solving approach for optimizing the processing time and methods based on metaheuristics. The proposed approach is inspired by the behavior of the human body. This hybridization is between a multi agent system and inspirations of the human body, especially genetics. The effectiveness of our approach has been demonstrated repeatedly in this paper. The proposed approach is applied to three preventive maintenance policies. These policies are intended to maximize the availability or to maintain a minimum level of reliability during the production chain. The results show that our algorithm outperforms existing algorithms. We assumed that the machines might be unavailable periodically during the production scheduling.

Key words: Multi agent systems, emergence, genetic algorithm, makespan, systematic maintenance, scheduling, hybrid flow shop scheduling.

INTRODUCTION

One of the assumptions of the most studied scheduling is the consideration that machines may not be periodically available during the production scheduling. Although many researchers have attempted to integrate the production and preventive maintenance planning by different methods, some of these methods are so complex that one cannot independently code them to achieve the same effectiveness, or some strongly used, the specific functionalities of the original problem considered cannot be extended to other problems.

This paper proposes to apply a simple methods of integration, yet easily extendable to other scheduling problems of the machine. This paper examines the scheduling of a workshop under the hybrid flow shop systematic preventive maintenance.

The objective is to minimize the execution time. A multi agent based on emergence method, including genetic algorithm, and some constructive heuristics are developed to tackle the problem.

SYSTEMATIC PREVENTIVE MAINTENANCE

Companies need different types of machines to produce goods. Each machine is not reliable in the sense that it deteriorates with age and use, and ultimately fails (Kelly

*Corresponding author. E-mail: abdelhadi.adel@gmail.com. Tel: +213 33 81 52 49. Fax: +213 33 81 52 49.

and Harris, 1978). Maintenance operations can be classified into two major groups: corrective maintenance (CM) and preventive maintenance (PM). The CM corresponds to actions during failure that has already occurred. The PM is the measure of a system while it is still active. The PM is done to maintain the system at the desired operation.

Several policies can be defined in order to determine when it is necessary to conduct operations on PM machines according to various criteria (Ozekici, 1996; Gertsbakh, 1977; Nakajima, 1989; Barlow and Hunter, 1960). The following are three classic policies (Celeux et al., 2006).

Policy I: Preventive maintenance at fixed time intervals and predefined

Operations of the PM are provided in advance in predefined time intervals (T_{PMF}) regardless of probabilistic models for the time of failure and make the best use of outages after a week, a month or even periods of annual production cycles. In this policy, the fixed time intervals are determined and PM operations are performed exactly these time intervals. As we have assumed that the Jobs are non-preemptive (the process of a job cannot be interrupted), whenever there is an overlap between the processes of a job and the operations of the PM (Bunea and Bedford, 2001).

Policy II: Model of the optimal period for preventive maintenance, maximizing machine availability

Classically, the optimal period between two sequential preventive maintenance activities is determined by maximizing machine availability. In this policy, the PM is performed based on the optimal maintenance period. The time of failure is assumed to follow a Weibull probability distribution, T ≈ W [θ, β] with β> 1. Tr is the number of time units that repair and Tp is the number of time units of the PM. T_{PM} is the interval between two consecutive PM. The objective of this policy is to maximize system availability. According to Kutanoglu (2003), the optimal maintenance T_{PMop} can be calculated by the Equation 1:

$$Tpmop = \theta.\left[\frac{Tp}{Tr(\beta-1)}\right]^{1/\beta} \quad (1)$$

In general, we can say that the policy II is to conduct PM whenever a machine is in units of time T_{PMop} operation.

Policy III: Maintain a minimum threshold of reliability for a given production period t

In some systems, aging and wear affect the failure rate, that is to say, it can be increased over time. This policy is to conduct a systematic PM after a T_{PM} to ensure a minimum of system reliability (R_0 (t)) from time t = 0. It is assumed that the PM restores the machine to good as new condition. In this case, the PM will be conducted at regular intervals 0, T_{PM}, 2 T_{PM}, 3 T_{PM}, ….., N T_{PM} which are considered as points of renovation. When time to failure follows a Weibull model, T ≈ W [θ, β], with β> 1 (the failure rate increases with time), the time between the PM in this policy can be obtained by the Equation 2:

$$T\mathrm{pm} = \left[-\theta^{\beta}.\frac{\ln R0(t)}{t}\right]^{1/\beta-1} \quad (2)$$

Integration of PM and production scheduling in policy III is made the same way as we do in policy II.

In Policy II and III, the PM depends on the extent of time that the machine is in operation, in contrast to policy I where the activities are performed PM function of time (the time of operation of the machines n is not important). Another problem is the duration of the activities of the PM, which is called D_{PM} (Kutanoglu, 2003).

HYBRID FLOW SHOP SCHEDULING

The hybrid flow shop scheduling problem (HFSP) can be stated as follows: consider a set of n jobs to be processed in m stages. Each stage i can have several identical machines in parallel, denoted by m_i. In HFSP, we need all the jobs go through the stages of the same order starting from stage 1 to stage m. Each job can be operated by any machine all in one stage, however, when it is assigned to a machine, the process cannot be interrupted. Each machine can run on only one job at a time. There is no precedence constraint between jobs, that is to say, they can be processed in any order. The processing time of each job j at stage i (denoted by Pj,i) i is fixed and known in advance. Since the machines are identical, the processing time of a job is a constant stage between machines on this stage. In HFSP, there are two dimensions of decision:

i) Sequence job.
ii) Assign the job to machines on each stage.

Figure 1 shows a diagram of a hybrid flow shop workshop.

LITERATURE REVIEW OF HFSP PROBLEM WITH MAKESPAN CRITERION

Many realistic assumptions have been incorporated into scheduling problems. For example, due to the interaction between the activities of production and maintenance, many researchers have studied together to plan two activities.

Adiri et al. (1998) shows that the problem of Flow Shop

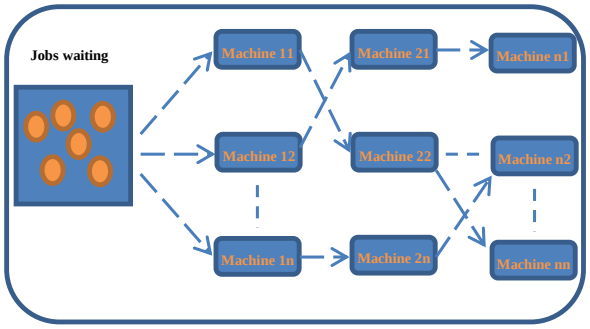

Figure 1. Diagram of a hybrid flow shop workshop.

with downtime on one machine is NP-hard.

Kubiak et al. (2002) explores scheduling a flow shop with availability constraints (SFSAC). He considers two variants of non-preemptive SFSAC. In the first variant, the starting times of maintenance activities are fixed, while in the second the starting times of maintenance activities are expected to be flexible. An algorithm based on genetic algorithm and taboo search is applied to solve the problem.

Cheng and Wang (2005) explores the non-preemptive scheduling of two stages Flow Shop with one machine on the first stage and m machines on the second stage in the minimization of the execution time. They assume that each machine had over a lockup period and these periods are known in advance. They also investigate the worst case performance of three other heuristics.

Reeves (2005) intend to integrate a single machine scheduling and planning of preventive maintenance. They develop a genetic algorithm to solve the integrated model developed by Kutanoglu (2003).

Blazewicz et al. (2001) investigate the two machines Flow Shop by any number of downtime on a single machine and prove that the problem of minimizing the makespan is strongly NP-hard.

Breit (2006) addresses the problem of scheduling n preemptive job in a Flow Shop two machines on which the first machine is not available for processing during a given time interval.

Allahverdi (1995) studied the problem with stochastic machine breakdowns and time settings separated. Yang et al. (2008) considered a two-machine flow shop where maintenance activities must be made after obtaining a fixed number of jobs. The durations of these maintenance activities are constants.

Schmidt (2000) survey existing methods to solve scheduling problems under constraints of availability and the complexity of the results.

METHODS OF SOLVING THE PROBLEM OF SCHEDULING FLOW SHOP

Minimizing the makespan in the case of general flow shop is NP-hard in the strong sense. Several heuristics have been proposed to solve it, and especially that of Johnson and Nawaz Enscore and Ham (NEH). Each of these heuristics gives good results, but do not guarantee the optimal solution.

Johnson's algorithm

Johnson's algorithm is a paradox scheduling. It is undoubtedly the most cited reference in the world of scheduling. Hardly anyone has read it, it has virtually no industrial interest, but the originality of its approach and simplicity make it a "cult object".

Johnson's algorithm (Mccall, 1956) is applied to a problem of two-machine Flow Shop and the criterion is to maximize the Cmax (makespan).

NEH algorithm

In 1983, Nawaz, Enscore Jr. and Ham proposed an algorithm based on the assumption that a lot with a total execution time is high priority (the lot is located primarily in a partial ordering) compared a lower task in the case of minimizing makespan. We adapted this heuristic for the case of minimizing the sum of the delays, focusing on tasks that have the value:

$$neh_i = \left(\frac{1}{w_i}\right) * \left(d_i - \left(r_i + \sum_{j=1}^{m} p_{ij}\right)\right) \quad (3)$$

Other methods of resolution

There are other methods for solving the scheduling problem, such as:

i) The application of heuristic shorter duration of treatment (or HSDT) for HFSP: HSDT organizes Jobs in ascending order processing time Jobs in stage 1 (Adiri et al., 1998).
ii) The application of heuristic longer duration of treatment (or HLDT) for HFSP: HLDT organizes Jobs in decreasing order of processing time Jobs in stage 1 (Adiri et al., 1998).

LIMITATIONS OF EXISTING METHODS

Although a large number of methods, including mathematical programming, different criteria and heuristics have been presented to integrate production scheduling and maintenance, they have many drawbacks.

For example, the proposed methods are often quite complex and require arduous tasks of coding to implement. The Exact methods are not practical for small instances (up

to 10 to 15 Jobs) and even in this case, the computation time tends to be very high.

MULTI-AGENT SYSTEM AND GENETIC ALGORITHM APPROACH (MASGA)

Our proposed multi agent approach is based on emergence method; genetic algorithm (GA) looking for a problem space with a population of chromosomes, each of which represents a coded solution. Fitness value is assigned to each chromosome based on its performance. The most desirable is the chromosome which has the smallest value. The population evolves through a set of operators until a stopping criterion is visited. A typical iteration GA, a generation proceeds as such:

- The best chromosomes of the current population (Nr individuals) are copied directly to the next generation (strategy of the elite).
- A selection mechanism selects the chromosomes of the current population so that the chromosome with the highest fitness value decline has more chance of being selected.
- The selected chromosomes mate and produce new offspring (crossover).
- After the breeding process, each offspring could mutate into another mechanism called mutation with probability Pm.
- Then the new population is evaluated again and the whole process is repeated (Goldberg, 1989).

Flowchart of our approach MASGA

Figure 2 shows a flowchart of our approach MASGA for the integration of maintenance policies in hybrid flow shop scheduling. Each agent in our approach performs the same process, that is, the calculation of the objective function (fitness), which is in our case the value of the makespan.

General procedure of our approach MASGA

The general procedure of our proposal summarized as follows:

```
Procedure the proposed MASGA Algorithm
  Generating a set of chromosomes (TP) as a starting population
    while a stopping criterion is not reached do
      Calculate the values of Fitness
      Perform the elite strategy / / Nr persons are copied to the
                                    next generation
      For I = 1 to Nr + TP to
        Select two parents using a selection mechanism
        Perform crossover on two selected parents and
          Generate the ith descending
        If rand <Pm then / / rand is a random number
                                    uniformly distributed (0, 1)
          Make a mutation in the i-th descendant
        endif
      endfor
    Endwhile
end
```

Each agent in our approach performs the following operations:

- Produce a set of chromosomes (TP) as a starting population.
- Calculate the values of Fitness (makespan).
- Select two parents using a selection mechanism.
- Perform uniform crossover on two selected parents.
- Perform a single point mutation type.

This work is repeated until the arrival to the agent that the value of the lowest fitness.

Coding scheme and operators of MASGA

In GA, the representation of chromosomes is binary strings consisting of 0 and 1, which is obviously not appropriate to describe HFSP because it is quite annoying to represent and use to schedule in this form.

Random key

The coding scheme is random key (RK), the first representation proposed by Norman and Bean (1999) for problems of several identical machines and later used by (Goldberg, 1989; Norman and Bean, 1999; Zandieh et al., 2006).

The most important advantages of this type of coding scheme are to be simple to implement and easily adaptable to all operators. It could be described as follows:

- Each Job is assigned a real number whose integer part is the number of the machine to which the job is assigned and whose fractional part is used to control the Jobs assigned to each machine.
- Random numbers are used only for the first stage. They determine the sequence and assignment Job only for stage 1.
- For all successive stages i, i = {2, 3, ..., m} Job sequence is determined by the earliest time for completion of Jobs in the preceding stage and the assignment rule the machine is the first machine available. For example, consider a problem with n = 4, m = 2, m1 = 2, m2 = 2.

For our problem, we generate four random numbers from a uniform distribution between (1, 1 + m1) for the first stage (Figure 3).

It is well known that the initial solutions can strongly influence the final results obtained by the GA. We therefore generated initial solutions as follows: four solutions are produced by the heuristics HSDT, HLDT, Johnson rule and NEH_H, and the rest is randomly generated.

Chromosomes low makespans are the most desirable and, therefore, a number of chromosomes (Nr) with the lowest values of makespan are automatically copied to the next generation. This mechanism is called reproduction. The rest of the chromosomes (TP - Nr)% or offspring are produced by crossing two other sequences or relatives by an operator called crossover operator. The crossover operators should avoid generating infeasible solutions.

Selection mechanism

For selection of parents to undergo crossover, we use the classification selection that could be described as follows:

- Individuals of the current population are first sorted according to their objective functions.
- Each individual is assigned a probability normalized so that the best solutions are more likely to be selected.
- Then, individuals are randomly selected as parents to submit to operators based on their probabilities.

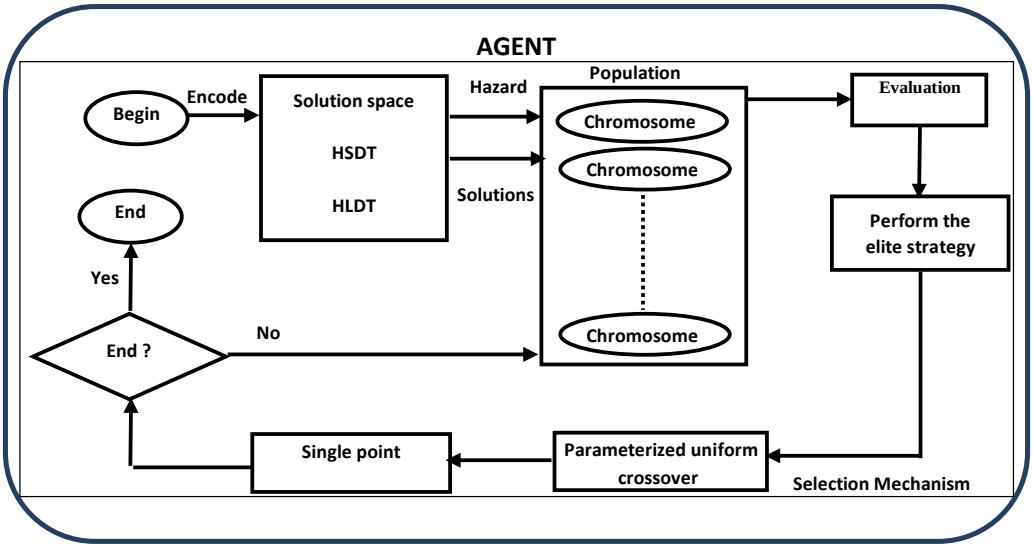

Figure 2. Flowchart of our approach MASGA for the integration of maintenance policies in hybrid flow shop scheduling.

2,96	1,58	2,22	1,13

Figure 3. Encoded solution using CK representation.

Parent 1	2,66	1,58	2,92	1,13	2,64
Parent 2	1,42	2,81	2,12	1,92	2,48
Random N°	0,62	0,34	0,97	0,12	0,89
Child	2,66	1,58	2,12	1,13	2,48

Figure 4. Procedure for uniform crossover applied to an example with n = 5 and m1 = 2.

Crossover uniform set

The goal is to generate a better offspring, that is to say, to create better sequences by combining the parents. Our crossover is uniformly set (CUS), because it has shown its effectiveness in HFSP in previous studies in the literature (Goldberg, 1989; Norman and Bean, 1999; Zandieh et al., 2006). It is necessary to specify the work of CUS by random keys and defined as follows:

- For each job a random number between (0, 1) is generated.
- If the value is less than 0.8, corresponding to the Job Board parent 1 is copied to the child if the parent RK 2 is selected.
- The Jobs are sorted in ascending order of RK.

The procedure is illustrated numerically by applying it to an example with n = 5 and m1 = 2 as shown in Figure 4.

Single point mutation

A mutation operator is used to tweak the sequence, that is, generate a new sequence, but similar. The main purpose of the application of mutation is to avoid convergence to a local optimum and diversifying population. The mutation operator can also be seen as a simple form of local search.

Many researchers have concluded that only the single point mutation, called SPM can provide better results than other mutations such as SWAP or inversion.

Therefore, we use the SPM mutation (Ansell and Phillips, 1997; Barros, 2003). SPM procedure can be stated as follows:

Before Mutation	**1,62**	**2,34**	**1,97**	**2,12**	**2,89**
After Mutation	**1,62**	**1,68**	**1,97**	**2,12**	**2,89**

Figure 5. Procedure single point mutation applied to an example with n = 5 and m1 = 2.

Table 1. MASGA levels parameters.

Parameters	N° of level	Levels
Size population	3	50, 100, 150
(N_r, P_m)	3	(1, 0.10), (2, 0.15), (3, 0.20)

- RK of Job randomly chosen at random is regenerated. Figure 5 shows an illustrative solution that mutates.

RESULTS AND DISCUSSION

Here, we evaluate our multi-agent approach based on emergence method proposed. Our goal is to compare our approach with heuristics, NEH, Johnson rule (m/2, m/2), HSDT and HLDT.

We implemented its heuristics in MATLAB 7.0 that runs on a PC Intel Core 2 Duo 2.0 GHz and 2 GB RAM. The stopping criterion used when testing with all instances of heuristics is a set time limit of the CPU set to $n^2 \times m \times 1.5$ Ms. This stopping criterion cannot only be longer than the number of Jobs or increase in stages, it is also more sensitive to an increase in the number of Jobs than the number of stages.

We use the relative percentage deviation (RPD) as a performance measure to compare common methods. The right solution obtained for each instance (named Minsol) is calculated by an algorithm. RPD is obtained by the following formula:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \qquad (4)$$

where Alg_{sol} is the value of the objective function obtained for a given algorithm and instance. Clearly, lower values of RPD are preferable.

Setting the parameters

It is known that the different levels of parameters affect clearly the quality of the solutions obtained by our approach MASGA. We applied a set of parameters on the size of the population (SP), the number of solutions directly copied to the next population (Nr), the probability of mutation (Pm). Table 1 show the levels considered.

A set of 30 cases in group 3 (n = 40, 70, 100) is generated and solved by the algorithms. After analyzing the results MASGA we choose Nr = 10, Pm = 0.15 and SP = 100.

Data generation

The data needed to solve this problem consists of two parts, the data on the production scheduling and data for preventive maintenance. It is necessary to process that data must be produced in order to ensure that a large number of operations performed on each PM would machine. If the time between two consecutive operations of the PM is less than the maximum processing time, cannot be certain Jobs never treated. On the other hand, if time becomes very large, it is very likely that no operation of the PM is required.

The first part of the data includes the number of Job (n), number of stages (m), the number of identical machines at each stage (m_i) range of processing time (Pj,i) and time loans; Is n = {40, 70, 100}, and m = {2, 4, 8}. To set the number of machines at each stage, we have two sets. In the first, we have a number of uniform random distributions of machines ranging from one to three machines per stage, and in the second, we have a fixed number of two machines on each stage. Times ready for stage 1 are set to 0 for all Jobs. Times ready to stage (i + 1) is the execution time in stage i, so these data should not be generated. Table 2 shows the factors and their levels.

The second part of the data is divided into three parts, each of which considers a policy. As mentioned above, the generation T_{PMF}, T_{PMop} and T_{PM} must be made with the utmost care. To do this, we need to define an artificial variable "x_i" to estimate the workload on the machines in each stage i as follows:

$$x_i \approx \frac{n}{m_i}$$

Table 2. Factors and levels.

Factors	Levels
Number of Jobs	40, 70, 100
Number des stages	2, 4, 8
Distribution of a machine	a. Constant : 2 b. Variable : U (1, 3)
Processing time	U (1, 99)

Table 3. Average RPD for the algorithms grouped by n and m for policy I.

n	*m*	*Algorithms*				
		MASGA	***NEH$_H$***	***John***	***HSDT***	***HLDT***
40	2	3.36	8.69	20.16	24.49	30.48
	4	3.64	9.96	19.98	30.63	32.01
	8	4.37	7.75	22.13	28.59	30.75
70	2	4.40	8.29	25.17	31.11	27.26
	4	2.66	9.80	18.03	25.12	30.91
	8	3.87	7.11	19.02	21.30	28.06
100	2	2.47	5.68	23.75	31.17	31.32
	4	3.11	4.57	24.48	24.08	26.79
	8	3.52	4.91	20.69	27.18	25.52
Average		3.49	7.42	21.49	27.08	29.23

So that "x_i" is the expected number of Jobs on each machine in stage i. Therefore, the range of this variable is as follows:

$x_i = \{10,13.3,17.5,20,23.3,25,33.3,35,40,50,70,100\}$

For example, in the case of n = 70, g = 2, m1 = m2 = 2 and 3, we obtain x1 = x2 = 35 and 23.3. Other data required for each policy are generated as follows:

1. The data for the policy I for PM: T_{PMF} are determined according to xi. If xi <25, then T_{PMF} = 450, otherwise T_{PMF} = 650. The duration of the operation PM (D_{PM}) is set to 50, 100 and 150% of the processing time.
2. The data for the policy II for PM: As mentioned earlier, there are nine combinations of n and g. For each combination, β = {2, 3, 4} is defined. D_{PM} is the same as the policy I. In this policy, tp is set to 1 and 8 rpm for all experiments. The values of θ are set according to the variable x. Levels of θ are chosen to ensure that a significant number of transactions would be carried out in each PM machine. For example, it should be noted that a small value for θ would result in a very large value of T_{PMop} then a very large value would probably hinder the achievement of certain treatments Jobs on machines without interruptions due to the small amount of T_{PMop}.
3. The data for the policy III for PM: Levels of θ, β and D_{PM} are the same as the policy II. The goal is 95% after the period t of production, so R_0 (t) = 0.95. To calculate T_{PM}, it is necessary to determine the time t, which can be easily obtained from the processing time of a given instance. This is as a result of the processing times being uniformly distributed over (1, 99), t = xi * 50.

All results of the different levels of factors cited in 54, 162 and 162 scenarios for policy I, II and III, respectively. For each scenario, there are 10 different problems that result from a total of 540, 1620, 1620 instances.

Experimental results

The results of experiments on average for each combination of n and m (180 data by the mean) in three subsets (Policies I, II and III) are summarized in Tables 3, 4 and 5. As expected, metaheuristics algorithms perform better than the heuristics in the three policies. The

Table 4. Average RPD for the algorithms grouped by n and m for policy II.

n	*m*	*Algorithms*				
		MASGA	***NEH$_H$***	***John***	***HSDT***	***HLDT***
40	2	2.79	9.43	16.83	28.37	28.94
	4	3.51	9.81	18.82	27.89	31.87
	8	3.20	7.57	18.66	24.28	25.75
70	2	4.19	8.98	19.61	28.43	28.96
	4	3.11	9.12	18.71	25.17	29.60
	8	3.21	8.62	16.98	28.26	31.52
100	2	2.23	6.25	19.98	26.05	31.97
	4	3.02	5.31	16.64	27.99	27.53
	8	3.46	5.40	16.54	26.45	26.41
Average		3.19	7.83	18.09	26.99	29.17

Table 5. Average RPD for the algorithms grouped by n and m for policy III.

n	*m*	*Algorithms*				
		MASGA	***NEH$_H$***	***John***	***HSDT***	***HLDT***
40	2	**2.32**	8.74	16.06	26.99	25.94
	4	**3.99**	8.36	17.93	24.01	29.57
	8	**3.29**	7.37	19.70	28.69	28.14
70	2	**3.23**	9.59	17.44	27.18	32.06
	4	**4.67**	9.88	18.56	29.80	29.59
	8	**2.49**	8.40	21.90	27.32	32.74
100	2	**4.40**	5.34	18.89	30.13	30.49
	4	**4.94**	5.36	20.48	27.44	28.42
	8	**3.34**	4.13	18.72	31.70	30.51
Average		**3.63**	7.46	18.85	28.14	29.72

proposed MASGA provides better results than other algorithms in three policies with a RPD of 3.49, 3.19 and 3.63% respectively, while NEH_H get a RPD of 7.42, 7.83 and 7.46% in the policies I, II and III respectively. The results of the RPD of our approach MASGA down all 9 groups (combinations of n and m) as well as maintaining its robustness in the three policies PM.

As can be seen, NEH_H gets remarkably better results than other heuristics with RPD 7.42, 7.83 and 7.46% in the policy I, II and III. After NEH_H, Johnson algorithm gets means RPD equal to 21.49, 18.09 and 18.85% in the policies I, II and III respectively. The worst of the worst performing algorithms are HSDT and HLDT with almost 30% of RPD.

Figure 6 shows the graph means and least significant difference (LSD) intervals for the algorithms. There are statistically significant differences between the performances of algorithms.

As shown in Figure 6, the proposed MASGA gives good results compared to other algorithms. We analyze the interactions between factors such as e.g. the number of job, number of stages and type of policy on PM algorithm performance. In the end, we outline how RPD is obtained by the algorithms with respect to different levels of the factors. Due to the significantly worse performance of HSDT and HLDT, we subtract from the experience.

Figures 7, 8 and 9 shows the graph means for interaction between different factors of metaheuristics algorithms. MASGA provides the lowest RPD in three levels of the number of Jobs. In increasing the number of jobs, NEH_H is

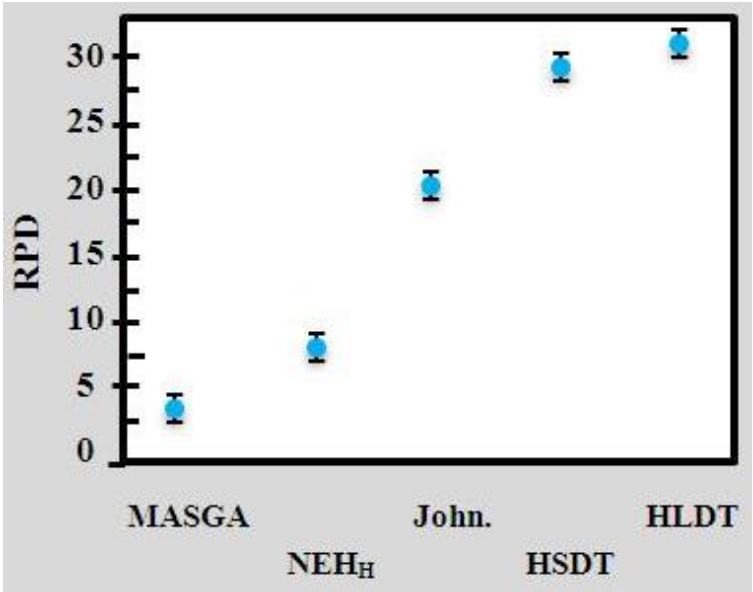

Figure 6. RPD graph mean and LSD intervals (at 95% confidence) for the type of factor algorithm.

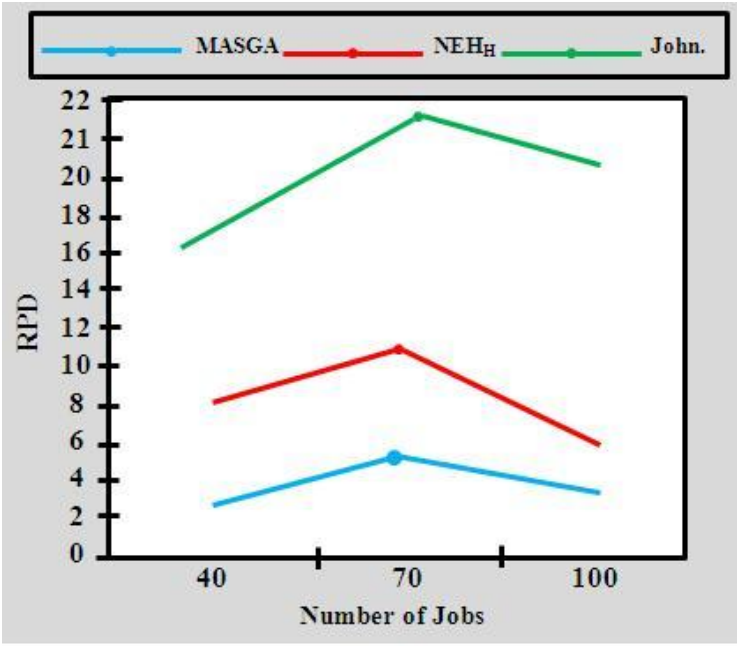

Figure 7. RPD graph mean for interaction between metaheuristics algorithms and number of jobs factor.

more efficient. Similarly, an increasing number of stages results in better performance for NEH_H. There is no interaction between the performance of algorithms and policies PM. In all cases, the MASGA gives the best results compared to other algorithms.

Conclusions

In this paper we presented our work - the integration of systematic preventive maintenance policies in hybrid flow shop shops scheduling to minimize makespan. It was

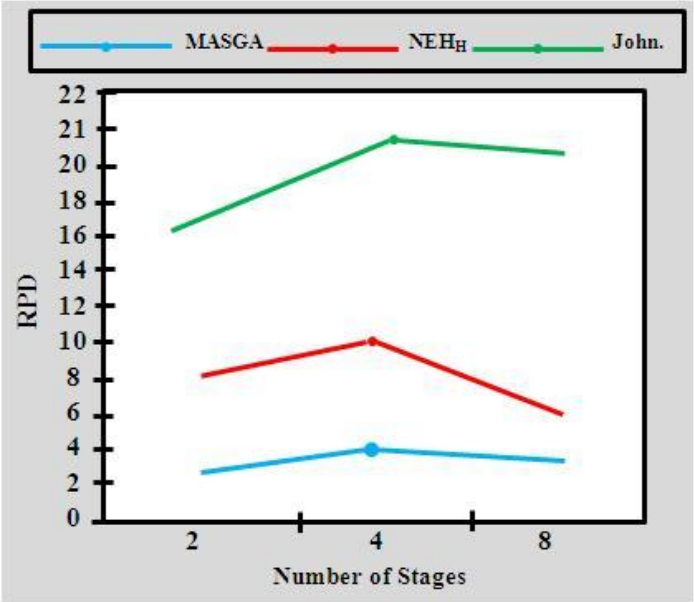

Figure 8. RPD graph mean for interaction between metaheuristics algorithms and number of stages factor.

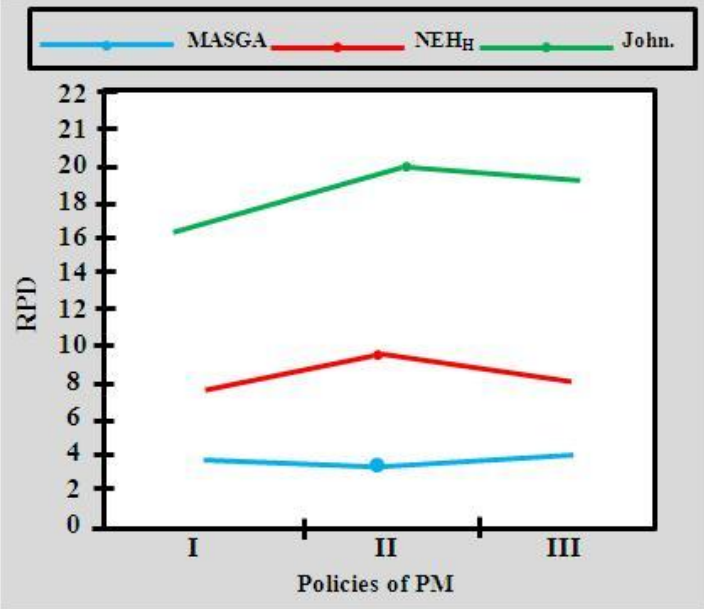

Figure 9. RPD graph mean for interaction between metaheuristics algorithms and policies of PM factor.

assumed that the machines could be periodically unavailable during the production scheduling. Including the criteria here were simple but effective. More importantly, they are adaptable to all scheduling problems. With this, we have overcome one of the key gaps in the integration of existing techniques in the literature.

To solve such a complex problem, we proposed a multi agent approach based on emergence method, genetic algorithm in which we use advanced operators such as uniform set crossover and single point mutation.

We also evaluated the adaptations of some well-known heuristics, including HSDT, HLDT, Johnson rule (m/2, m/2) and NEH_H.

A benchmark was established with great care to evaluate the algorithms. The benchmark content up to 100 jobs and 8 stages. All results showed that MASGA gives satisfactory results compared to other algorithms, in addition to its robustness in the three policies of PM.

ACKNOWLEDGEMENT

This work was conducted as part of research work of our doctoral thesis in the Laboratory of Automation and Production, in the Department of Industrial Engineering at the University of Batna, Algeria.

Nomenclature

CM : Corrective maintenance.
CUS : Crossover uniform set.
D_{MP} : Duration of preventive maintenance.
GA : Genetic algorithm.
HFSP : Hybrid flow shop scheduling problem.
HSDT : Heuristic shorter duration of treatment.
HLDT : Heuristic longest duration of treatment.
MAS : Multi agents system.
NEH : Nawaz Enscore Ham.
PM : Preventive maintenance.
Pj,i : Processing time of each job j at each stage i.
RPD : Relative percentage deviation, e.
RK : Random key.
SP : Systematic preventive maintenance.
SPM : Single point mutation.
SP : Size of population.
T_{PMF} : Time preventive maintenance fixed.
T_R : Time to repair.
T_{PM} : Time between two consecutive PM.
T_{PMop} : Time optimal preventive maintenance.
Xi : Artificial variable.

REFERENCES

Adiri I, Bruno J, Frosting E (1998). Single machine flow-time scheduling with a single breakdown. Acta Inform. 26:679-696.
Allahverdi A (1995). Two-stage production scheduling with separated setup times and stochastic breakdowns. J. Oper. Res. Soc. 46:896-904.
Ansell JI, Phillips MJ (1997). Practical aspects of modeling of repairable systems data using proportional hazards models. Reliab. Eng. Syst Saf. 58:165-171.
Barlow RE, Hunter LC (1960). Optimum preventive maintenance policies. Oper. Res. 8:90-100.
Barros A (2003). Maintenance of multi component systems under imperfect monitoring: modeling and stochastic optimization. PhD thesis. University of Technology of Troyes.
Blazewicz J, Breit J, Formanowicz P, Kubiak W, Schmidt G (2001). Heuristic algorithms for the two-machine flow shop problem with limited machine availability Omega. Comput. Oper. Res. 29:599-608.
Breit J (2006). A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint. Comput. Oper. Res. 33:2143–2153.
Bunea C, Bedford T (2001).The robustness of maintenance optimisation to modeling assumptions. In Proceedings of the European Safety and Reliability Conference ESREL'2001.
Celeux G, Corset F, Lannoy A, Ricard B (2006). Designing a Bayesian network for preventive maintenance from expert opinions in a rapid and reliable way. Reliab. Eng. Syst Saf. 91(7) 849-856.
Cheng T, Wang G (2005). An improved heuristic for two-machine flow shop scheduling with an availability constraint. Oper. Res. Lett. 26:223-235.
Gertsbakh I (1977). Models of Preventive Maintenance. North-Holland, Amsterdam.
Goldberg DE (1989). Genetic Algorithms in Search Optimization and Machine Learning. Reading Addison-Wesley. USA.
Kelly A, Harris MJ (1978). Management of Industrial Maintenance. Butter worth's Management Library. London.
Kubiak W, Blazewicz J, Formanowicz P, Breit J, Schmidt G (2002). Two-machine flow shops with limited machine availability. Eur. J. Oper. Res. 7:528-540.
Kutanoglu E (2003). Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. IIE Trans. 35(6):503-513.
Mccall JJ (1956). Maintenance policies for stochastically failing equipment: A survey. Manag. Sci. 11:493-524.
Nakajima S (1989). TPM Development Program: Implementing Total Productive Maintenance. Cambridge Productivity Press.
Nawaz M, Enscore EE, Ham I (1983). A heuristic algorithm for the m-machine, n-job flow shop sequencing problem OMEGA Int. J. Manag. Sci. 11(1):91–95.
Norman BA, Bean JC (1999). A genetic algorithm methodology for complex scheduling problems, Naval. Res. Logist. 46:199-211.
Ozekici S (1996). Reliability and maintenance of complex systems. NATO ASI Series –series F / Comput. Syst. Saf. 15:4.
Reeves CR (2005). A genetic algorithm for flow shop sequencing, Computers. Oper. Res. 22:5-13.
Schmidt G (2000). Scheduling with limited machine availability. Eur. J. Oper. Res. 121:1–15.
Yang DL, Hsu CJ, Kuo WH (2008). A two-machine flow shop scheduling problem with a separated maintenance constraint. Comput. Oper. Res. 35(3):876-883.
Zandieh M, Ghomi SMTF, Husseini SMM (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence dependent setup times. J. Appl. Math. Comput. 180:111-127.