

Research Article

A New Method Based on Multi Agent System and Artificial Immune System for Systematic Maintenance

Adel Abdelhadi and L. Hayet Mouss

Department of Industrial Engineering, Laboratory Automation and Production,
University of Batna, 1, Rue Chahid Boukhrouf 05000, Batna, Algeria

Abstract: This study propose a novel method for the integration of systematic preventive maintenance policies in hybrid flow shop scheduling. The proposed approach is inspired by the behavior of the human body. We have implemented a problem-solving approach for optimizing the processing time, methods based on Métaheuristiques. This hybridization is between a Multi agent system and inspirations of the human body, especially artificial immune system. The effectiveness of our approach has been demonstrated repeatedly in this study. 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.

Keywords: Artificial immune system, hybrid flow shop scheduling, makespan, multi agent systems, scheduling, systematic maintenance

INTRODUCTION

One of the assumptions of the most studied scheduling considering 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 study proposes to apply a simple methods of integration, yet easily extendible to other scheduling problems of the machine. This study 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 artificial immune system and some constructive heuristics are developed to tackle the problem.

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 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 $P_{j,i}$) 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:

- Sequence job
- 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 and Bruno (2002) shows that the problem of Flow Shop with downtime on one machine is NP-hard. Kubiak *et al.* (2002) explores Scheduling a Flow

Corresponding Author: Adel Abdelhadi, Department of Industrial Engineering, Laboratory Automation and Production, University of Batna, 1, Rue Chahid Boukhrouf 05000, Batna, Algeria

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

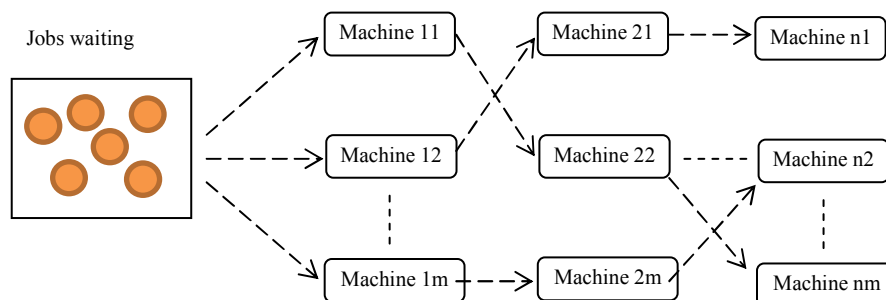


Fig. 1: Diagram of a hybrid flow shop workshop

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 (2007) 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 (2009) 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 (2007).

Blazewicz *et al.* (2008) 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 (2009) 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 (1999) studied the problem with stochastic machine breakdowns and time settings separated.

Yang *et al.* (2011) consider 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 (2005) survey existing methods to solve scheduling problems under constraints of availability and the complexity of the results.

METHODOLOGY

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 that of Nawaz Ensore and Ham (NEH). Each of these heuristics gives good results, but no guarantees the optimal solution.

Johnson algorithm: Johnson's algorithm is a paradox scheduling. Johnson Reference 54 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 C_{max} (makespan).

NEH algorithm: Nawaz *et al.* (1983) 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:

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

Other methods of resolution: There are other methods for solving the scheduling problem as:

- 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 and Bruno, 2002).
- 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 and Bruno, 2002).

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

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 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 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 operating.

Several policies can be defined in Ozekici (1996), Gertsbakh (1987), Nakajima (1999) and Barlow and Hunter (1960), in order to determine when it is necessary to conduct operations on PM machines according to various criteria. The following are three classic policies (Celeux *et al.*, 2006).

Policy I: Preventive maintenance at fixed time intervals and predefined: The 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, 2005).

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, $T \approx W [\theta, \beta]$ with $\beta > 1$. 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 Kutanoğlu (2007), the optimal maintenance T_{PMop} can be calculated by the Eq. (1):

$$T_{pmop} = \theta \cdot \left[\frac{T_p}{Tr(\beta - 1)} \right]^{1/\beta} \quad (2)$$

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 TPM 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 $0, T_{PM}, 2 T_{PM}, 3 T_{PM}, \dots, N T_{PM}$ which are considered as points of renovation. When time to failure follows a Weibull model, $T \approx W [\theta, \beta]$, with $\beta > 1$ (the failure rate increases with time), the time between the PM in this policy can be obtained by the Eq. (2):

$$T_{pm} = \left[-\theta^\beta \cdot \frac{\ln R_0(t)}{t} \right]^{1/\beta-1} \quad (3)$$

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} (Kutanoğlu, 2007).

Multi-Agent System and Artificial Immune System approach (MASAIS): In our proposed approach MASAIS, antigens refer to the objective function (makespan) to be optimized. Antibodies referring to candidate solutions.

The approach proposed in these AIS was built based on clonal selection and inspired particularly by the proposed CLONALG algorithm. AIS also based on the principle of affinity maturation of the immune system.

AIS start from a number of antibodies, called population. Population is improved by a set of operators until a stopping criterion is reached.

An iteration of the AIS to generate the next population could be described as follows:

- First, an acceleration mechanism is applied. The acceleration mechanism, candidates with better ability solutions are transferred to the next population.
- The other antibodies of the new population are multiplied by mutation and crossover operators.
- To select antibodies undergo operators, a selection function that uses a value well of each antibody and an affinity calculation is applied.
- Firstly, the candidate solutions with better objective functions are given higher probabilities of being selected to produce the next generation of candidate solutions.

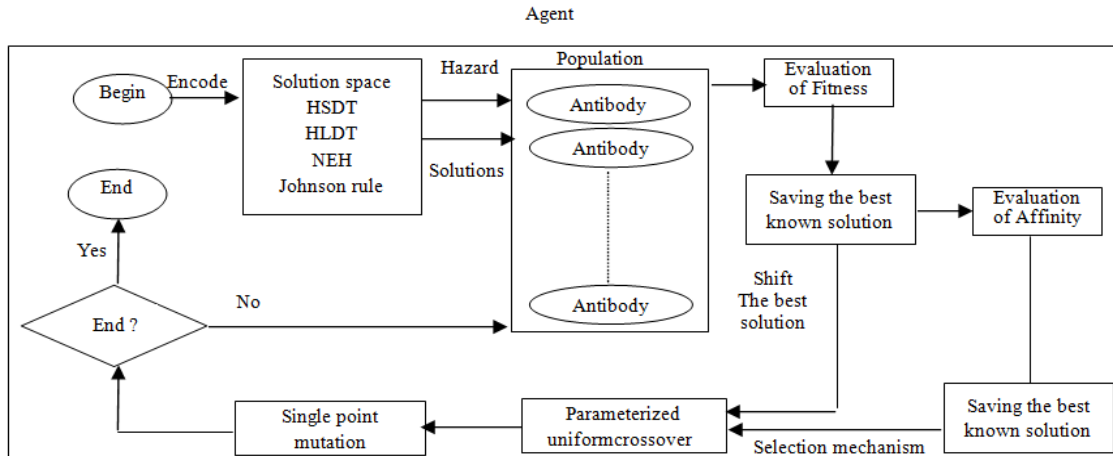


Fig. 2: Flowchart of our approach MASAIS for the integration of maintenance policies in hybrid flow shop scheduling

- This mechanism is to ensure that the next generation contains a large proportion of candidate's solutions with good properties.

On the other hand, the calculation of affinities between antibodies is to remove similar antibodies. It performs the following tasks.

If a candidate solution to a value greater than a prescribed Affinity Threshold value (AT), it is assigned a lower probability by multiplying the prior probability obtained from the value of many of the antibody with a lower factor which is Affinity Adjustment (AA). This will reduce the probability of being selected.

Flowchart of our approach MASAIS: Figure 2 shows a detailed flow chart of our MASAIS approach for the integration of maintenance policies in hybrid flow shop scheduling.

Each agent in our approach performs the same process, the calculation of the objective function (Fitness), which corresponds in our case the value of the makespan.

Each agent in our approach performs the following operations:

- Produce a set of antibodies (TP) as a starting population
- Calculate antibodies good values
- Calculate affinity values antibodies
- Perform an acceleration mechanism
- Select two parents using a selection mechanism
- Perform uniform crossover
- Make a point mutation

This study is repeated until the arrival to the agent who has the lowest fitness value.

Coding scheme and operators of MASAIS:

Initialization:

Setting parameters: Define the size of the Population (TP), the number of solutions directly copied to the next

population (Nr), the Probability of mutation (Pm), the weight of similarity factors (W_1, W_2, W_3, W_4), the Affinity Threshold (AT) and Adjusting the Affinity (AA).

Generation of initial population:

- Randomly generating an initial population of (TP-4) antibodies
- Generate antibodies with HSDT
- Generate antibodies with HLDT
- Generate antibodies with Johnson rule ($m/2, m/2$)
- Generate antibodies with NEHH

Random key: The coding scheme is Random Key (RK), the first representation proposed by Norman and Bean (2001) for problems of several identical machines and later used by Goldberg (1989) and Zandieh *et al.* (2010).

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:

- 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.
- 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.

2.96	1.58	2.22	1.13
------	------	------	------

Fig. 3: Encoded solution using CK representation

- For all successive stages $i, i = \{2, 3, \dots, m\}$ 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 $n = 4, m = 2, m_1 = 2, m_2 = 2$.

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

It is well known that the initial solutions can strongly influence the final results obtained by the AIS. We therefore generated initial solutions as follows: four solutions are produced by the heuristics HSDT, HLDT, Johnson rule and NEHH 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 the selection of parents undergo crossover, we use the classification selection that could be described as follows.

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; Zandieh *et al.*, 2010). It is necessary to specify that 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 $m_1 = 2$ shown in Fig. 4.

Single point mutation: A mutation operator is used to tweak the sequence, i.e., generate a new sequence, but similar. The main purpose of the application of

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

Fig. 4: Procedure for uniform crossover applied to an example with $n = 5$ and $m_1 = 2$

Before mutation	1.62	2.34	1.97	2.12	2.89
After mutation	1.62	1.68	1.97	2.12	2.89

Fig. 5: Procedure single point mutation applied to an example with $n = 5$ and $m_1 = 2$

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, 2003; Barros, 2007). SPM procedure can be stated as follows.

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

Calculation of the objective function: In general, an objective function (Fitness) includes one or more performance indicators that measure the effectiveness of an antibody.

The antibody candidates are first transformed into a scheduling valid. In the end, they are evaluated using an objective function to obtain their fitness values. For a maximization problem, a high value of fitness is desirable and attempts by the AIS maximize it.

For a minimization problem, the objective function is formulated in a way to transform it into a maximization problem.

In our case, the makespan must be minimized; a candidate solution with a high makespan is assigned a value of low fitness.

An antibody i the fitness function is calculated as follows:

$$f(i) = \frac{1}{\sum_{i=1}^m \frac{1}{C \max(i)}} \tag{4}$$

where,

- f (i) = The fitness value for the antibody I
- Cmax (i) = The makespan for the antibody i
- tp = The size of population

Affinity: According to the calculated probabilities with both the value and the value of good affinity, the selection is performed using a selection mechanism ranking.

To better evaluate the effectiveness of the function of the affinity of AIS, we again applied settings uniform crossover and single point mutation.

Evaluation affinity AIS increases antibody diversity in a population and, therefore, provides the opportunity to visit most of the search space at the expense of the computation time required.

The main question is whether in such a complex problem, the calculation of affinity is the cost, that is to say the proportion of time it consumes total calculation time of the algorithm. In the next section, we describe in detail the procedure for affinity calculation.

Calculation procedure affinity: To calculate the affinity antibodies are compared with the Best Known Antibody (BKA) obtained so far. The affinity simply expresses the similarity between an antibody and BKA.

The theory of entropy is used here to estimate the probability of recurrence of an information memorandum. Yang *et al.* (2011) set information entropy, $H(x)$ of a discrete random variable $X = \{x_1, x_2, \dots, x_n\}$ with probability mass function $P(X = x) = p_i$, $i = 1, 2, \dots, N$ such that:

$$H(x) = -\sum_{i=1}^n p_i \log(p_i) \quad (5)$$

The use of information entropy gives the similarity of the antibody i (or sequence i) relative to a reference antibody (or sequences) can be expressed as follows:

$$aff(i) = \frac{1}{1 + \frac{1}{k} \sum_{j=1}^k h_{ij}} \quad (6)$$

where,

- $aff(i)$ = The similarity measure
- k = The size of the population and
- $h_{ij} = -p_{ij} \log(p_{ij})$

The antibody affinity for each value in our problem is calculated as follows:

- For each Job j , a report of similarity is determined.
- Then, the average ratios of all jobs in an antibody are defined as the value of the overall affinity of the antibody.
- The report s similarity of job j is calculated by comparing the position of the job in the

corresponding antibodies and BKA. It is necessary to remember that in the representation of the RK random key, we specify the sequence of Jobs and the sale of the first floor.

- The sequence of Job and the allocation for subsequent stages are determined by the following rules: the time the earliest completion of Jobs to the previous stage and the first available machine, respectively. Therefore, the ratio of similarity is obtained from the position of Job j at stage 1.

Each criterion has a weight that shows its relative importance (denoted by W_i , $i = \{1, 2, 3, 4\}$). If a criterion is met, the job j receives its weight and the total value of the weights is the ratio of received similarity Job j . After calculating the similarity of each Job, average similarities Jobs are used as the affinity of the antibody candidate.

RESULTS AND DISCUSSION

In this section, we evaluate our multi-agent approach based on artificial immune system proposed.

Our goal is to compare our approach with heuristics, GA, 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 set a time limit of the CPU set to $n2 \times m \times 1.5$ msec. This stopping criterion cannot only longer than the number of Jobs or increases stages, but is also more sensitive to an increase in the number of Jobs that 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 Min_{sol}) is calculated by an algorithm. RPD is obtained by the following Eq. (4):

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} * 100 \quad (7)$$

Alg_{sol} where 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 MAS AIS. 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), the weight of similarity factors (W_1 ,

Table 1: MASAIS levels parameters

Parameters	N° of level	Levels
Size population (N, P _m)	3	50, 100, 150
(W ₁ , W ₂ , W ₃ , W ₄)	2	(1, 0.10), (2, 0.15), (3, 0.20) (0.20, 0.10, 0.35, 0.35), (0.30, 0.20, 0.25, 0.25)
(AT, AA)	3	(0.40, 0.50), (0.60, 0.50), (0.80, 0.50)

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)

W₂, W₃, W₄), the Affinity Threshold (AT) and Adjusting the Affinity (AA). Table 1 show the levels considered.

A set of 30 patients into 3 groups (n = 40, 70, 100) is generated and solved by the algorithms. After analyzing the results SMAAIS we choose Nr = 20, Pm = 0.15 and TP = 100, W₁ = 0.3, W₂ = 0.2, W₃ = 0.25 and W₄ = 0.25, AT = 0.6 and AA = 0.5. In other words, if the affinity value becomes greater than 0.6, the probability of being selected by a mechanism selected rating is multiplied by 0.5.

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 (mi) range of processing time (P_{j,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:

$$x_i \approx \frac{n}{m_i} \tag{8}$$

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.

The data for the policy I for PM: T_{PMF} are determined according to x_i. If x_i < 25, then T_{PMF} = 450, otherwise T_{PMF} = 650. The duration of the operation PM (D_{PM}) is set to 50, 100 and 150%, respectively of the processing time.

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 hindered the achievement of certain treatments Jobs on machines without interruptions due to the small amount of T_{PMOp}.

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 a 95% after the period t of production, so R0 (t) = 0.95. To calculate TPM, it is necessary to determine the time t, which can be easily obtained from the processing time of a given instance. Because the processing times are uniformly distributed over (1, 99), t = x_i * 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 experimental results, on average for each combination of n and m (180 data by the mean) in three subsets (Policy I, II and III) are summarized in Table 3 to 5. As expected, métaheuristiques algorithms perform better than the heuristics in the three policies. The proposed MASIA provides better results than in the three political GA with a RPD of 1.81, 1.74 and 1.67%, respectively, while MASGA gets a RPD of 3.49, 3.19 and 3.63% in the policies I, II and III, 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 PRE MASAIS down all 9 groups (combinations of n and m) as well as maintaining its strength in the three policies MP.

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

		Algorithms					
n	m	MAS AIS	MAS GA	NEH _H	John	HSDT	HLDT
40	2	1.95	3.36	8.69	20.16	24.49	30.48
	4	1.94	3.64	9.96	19.98	30.63	32.01
	8	2.37	4.37	7.75	22.13	28.59	30.75
70	2	1.93	4.40	8.29	25.17	31.11	27.26
	4	1.79	2.66	9.80	18.03	25.12	30.91
	8	1.38	3.87	7.11	19.02	21.30	28.06
100	2	1.20	2.47	5.68	23.75	31.17	31.32
	4	1.67	3.11	4.57	24.48	24.08	26.79
	8	2.06	3.52	4.91	20.69	27.18	25.52
Average		1.81	3.49	7.42	21.49	27.08	29.23

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

		Algorithms					
n	m	MAS AIS	MAS GA	NEH _H	John	HSDT	HLDT
40	2	1.04	2.79	9.43	16.83	28.37	28.94
	4	2.08	3.51	9.81	18.82	27.89	31.87
	8	2.33	3.20	7.57	18.66	24.28	25.75
70	2	1.73	4.19	8.98	19.61	28.43	28.96
	4	1.56	3.11	9.12	18.71	25.17	29.60
	8	1.51	3.21	8.62	16.98	28.26	31.52
100	2	1.27	2.23	6.25	19.98	26.05	31.97
	4	2.33	3.02	5.31	16.64	27.99	27.53
	8	1.82	3.46	5.40	16.54	26.45	26.41
Average		1.74	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

		Algorithms					
n	m	MAS AIS	MAS GA	NEH _H	John	HSDT	HLDT
40	2	1.38	2.32	8.74	16.06	26.99	25.94
	4	2.18	3.99	8.36	17.93	24.01	29.57
	8	1.21	3.29	7.37	19.70	28.69	28.14
70	2	1.40	3.23	9.59	17.44	27.18	32.06
	4	1.11	4.67	9.88	18.56	29.80	29.59
	8	2.01	2.49	8.40	21.90	27.32	32.74
100	2	1.38	4.40	5.34	18.89	30.13	30.49
	4	2.18	4.94	5.36	20.48	27.44	28.42
	8	2.16	3.34	4.13	18.72	31.70	30.51
Average		1.67	3.63	7.46	18.85	28.14	29.72

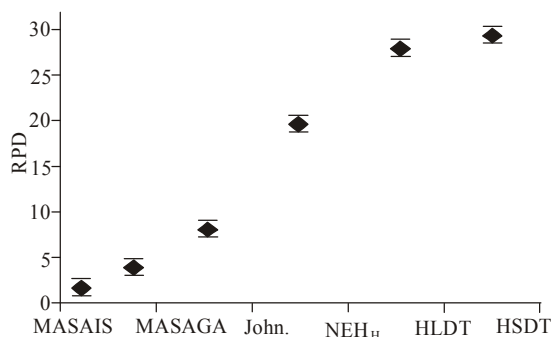


Fig. 6: RPD graph mean and LSD intervals (at 95% confidence) for the type of factor algorithm

As can be seen, NEHH gets remarkably better results than other heuristics with RPD 7.42, 7.83 and 7.46% in the policy I, II and III, respectively. After NEHH, 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 Fig. 6 the proposed MAS AIS 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 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.

Figure 7 to 9 shows the graph means for interaction between Métaheuristiques algorithms deferens factors.

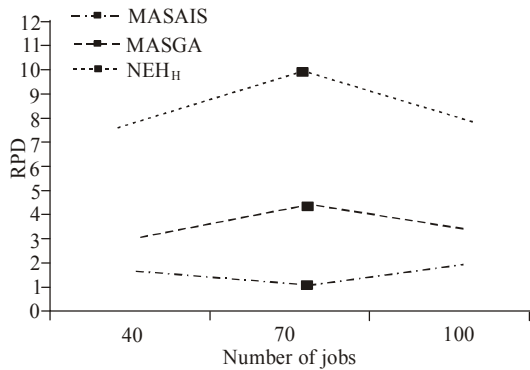


Fig. 7: RPD graph mean for interaction between metaheuristics algorithms and number of jobs factor

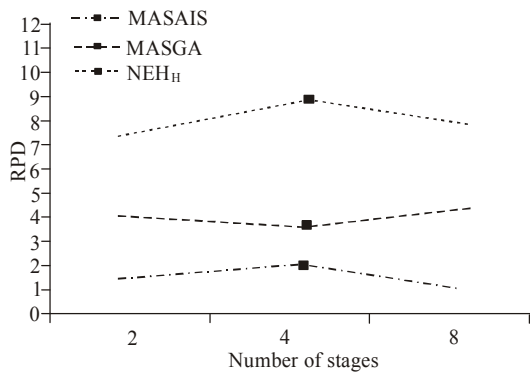


Fig. 8: RPD graph mean for interaction between metaheuristics algorithms and number of stages factor

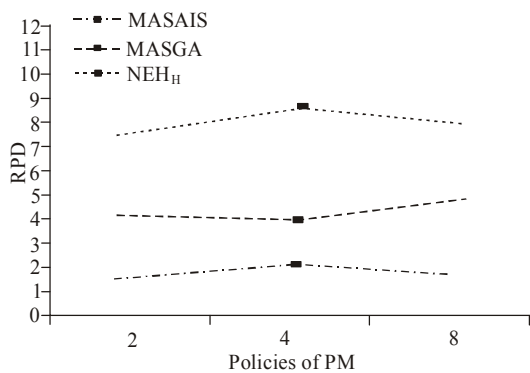


Fig. 9: RPD graph mean for interaction between metaheuristics algorithms and policies of PM factor

MAS AIS provides the lowest RPD in three levels of the number of Jobs. Such as increasing the number of jobs, NEHH is more efficient. Similarly, an increasing number of stages results in better performance for NEHH. There is no interaction between the performance of algorithms and policies PM. In all cases, the MAS AIS gives the best results compared to other algorithms.

CONCLUSION

In this study we presented our work, is the integration of systematic preventive maintenance policies in hybrid flow shop shops scheduling to minimize makespan. It was 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 artificial immune system algorithm in which we use advanced operators such as uniform set crossover and single point mutation.

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

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 MAS AIS gives satisfactory results compared to other algorithms, in addition to its robustness in the three policies of PM.

NOMENCLATURE

- AIS* : Immune System Artificial
- AA* : Affinity Adjustment
- AT* : Affinity Threshold
- 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
- LSD* : Least significant difference
- MAS* : Multi Agents System
- NEH* : Nawaz Enscore Ham
- PM* : Preventive Maintenance
- P_{j,i}* : Processing time of each job *j* at each stage *i*
- RPD* : Relative Percentage Deviation
- 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
- X_i* : Artificial Variable
- W_i* : Weight of similarity factors

ACKNOWLEDGMENT

This study 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.

REFERENCES

- Adiri, I. and J. Bruno, 2002. Frosting, single machine flow-time scheduling with a single breakdown. *Acta Inform.*, 26(6): 679-696.
- Allahverdi, A., 1999. Two-stage production scheduling with separated setup times and stochastic breakdowns. *J. Oper. Res. Soc.*, 46(2): 896-904.
- Ansell, J.I. and M.J. Phillips, 2003. Practical aspects of modeling of repairable systems data using proportional hazards models. *Reliab. Eng. Syst. Safe.*, 58(2): 165-171.
- Barlow, R.E. and L.C. Hunter, 1960. Optimum preventive maintenance policies. *Oper. Res.*, 8(3): 90-100.
- Barros, A., 2007. Maintenance of multi component systems under imperfect monitoring: Modeling and stochastic optimization. Ph.D. Thesis, University of Technology of Troyes.
- Blazewicz, J., J. Breit, P. Formanowicz, W. Kubiak and G. Schmidt, 2008. Heuristic algorithms for the two-machine flow shop problem with limited machine availability omega. *Comput. Oper. Res.*, 29(3): 599-608.
- Breit, J., 2009. A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint. *Comput. Oper. Res.*, 33(5): 2143-2153.
- Bunea, C. and T. Bedford, 2005. The robustness of maintenance optimisation to modeling assumptions. Proceedings of the European Safety and Reliability Conference ESREL'2005. September 16-20, Torino, Italy.
- Celeux, G., F. Corset, A. Lannoy and B. Ricard, 2006. Designing a Bayesian network for preventive maintenance from expert opinions in a rapid and reliable way. *Reliab. Eng. Syst. Safe.*, 91(7): 849-856.
- Cheng, T. and G. Wang, 2007. An improved heuristic for two-machine flow shop scheduling with an availability constraint. *Oper. Res. Lett.*, 26(2): 223-235.
- Gertsbakh, I., 1987. Models of Preventive Maintenance. North-Holland, Amsterdam.
- Goldberg, D.E., 1989. Genetic Algorithms in Search Optimization and Machine Learning. Reading Addison-Wesley, USA.
- Kelly, A. and M.J. Harris, 1978. Management of Industrial Maintenance. Butter Worth's Management Library, London.
- Kubiak, W., J. Blazewicz, P. Formanowicz, J. Breit and G. Schmidt, 2002. Two-machine flow shops with limited machine availability. *Eur. J. Oper. Res.*, 136(3): 528-540.
- Kutanoglu, E., 2007. Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. *IIE Trans.*, 35(6): 503-513.
- Mccall, J.J., 1956. Maintenance policies for stochastically failing equipment: A survey. *Manage. Sci.*, 11(1): 493-524.
- Nakajima, S., 1999. TPM Development Program: Implementing Total Productive Maintenance. Productivity Press, Cambridge.
- Nawaz, M., E.E. Ensore and I. Ham, 1983. A heuristic algorithm for the m-machine, n-job flow shop sequencing problem OMEGA. *Int. J. Manage. Sci.*, 11(1): 91-95.
- Norman, B.A. and J.C. Bean, 2001. A genetic algorithm methodology for complex scheduling problems. *Nav. Res. Log.*, 46(3): 199-211.
- Ozekici, S., 1996. Reliability and Maintenance of Complex Systems. Springer, Berlin, New York.
- Reeves, C.R., 2009. A genetic algorithm for flow shop sequencing. *Comput. Oper. Res.*, 22(5): 5-13.
- Schmidt, G., 2005. Scheduling with limited machine availability. *Eur. J. Oper. Res.*, 121(2): 1-15.
- Yang, D.L., C.J. Hsu and W.H. Kuo, 2011. A two-machine flow shop scheduling problem with a separated maintenance constraint. *Comput. Oper. Res.*, 35(3): 876-883.
- Zandieh, M., S.M.T.F. Ghomi and S.M.M. Husseini, 2010. An immune algorithm approach to hybrid flow shops scheduling with sequence dependent setup times. *J. Appl. Math. Comput.*, 180(3): 111-127.