HYBRID MULTI-AGENT AND IMMUNE ALGORITHM APPROACH TO HYBRID FLOW SHOPS SCHEDULING WITH SDST

Adel ABDELHADI[1], Leila Hayet MOUSS[2] and Ouahab KADRI[3]

[1] Department of Computer Science and Math, University of Khenchela, Algeria
Abdelhadi.adel@gmail.com
[2]Laboratory of Automation & Production Engineering, University of Batna, Algeria
Mouss.hayet@gmail.com
[3]Department of Computer Science and Math, University of BATNA 2, Algeria
ouehabk@yahoo.fr

***ABSTRACT**:The existing literature on process scheduling issues have either ignored installation times or assumed that installation times on all machines is free by association with the task sequence. This working arrangement addresses hybrid flow shop scheduling issues under which there are sequence-dependent configuration times referred to as HFS with SDST. This family of production systems are common in industries such as biological printed circuit boards, metallurgy and vehicles and automobiles making. Due to the increasing complexity of industrialized sectors, simple planning systems have failed to create a realistic industrial scheduling. Therefore, a hybrid multi-agent and immune algorithm can be used as an alternative approach to solve complex problems and produce an efficient industrial schedule in a timely manner. We propose in this paper a multi-agent and immune hybrid algorithms for scheduling HFS with SDST. The findings of this paper suggest that the proposed algorithm outperforms some of the existing ones including PSO (particle swarm optimization), GA (Genetic Algorithm), LSA (Local Search Algorithm) and NEHH (Nawaz Enscore and Ham).*

***KEYWORDS**: Hybrid flow shops; Multi Agent System; Immune algorithms; Makespan; Sequence dependent setup times.*

1 INTRODUCTION

The flow shop problem (FS) has been recognized as one of the most prominent challenges within the literature of scheduling systems (Gupta & Stafford, 2006). In FS, all actions lead to identical work instructions (processing method) that need to be processed at each step. This has been considered as a fundamental problem in the literature. Subsequently, the seminal work by (Brah, 1999) and (Gupta, 2000) has emphasized the importance of the HFS issue, prompting a growing research interest and contribution to the literature scheduling. In this context, it is important not to mix the HFS with multi-processor flow shop or flexible flow line when setting up machines simultaneously during the HFS production floors. According to (Brah, 1999) and (Gupta, 2000), available machines at each floor are identical. In all workshop-planning problems, the solution is to allow conducting a production sequence for the work performed on the machines to meet optimal performance standards and criteria.

One key difficulty in HFS is identifying a feasible schedule. This is because the scope of potential scheduling increases exponentially with the total number of diverse searches to process and the number of tasks executed in the production process. Thus, the two-floor HFS (through a machine in the first floor and a group of machines in the second floor) is an NP-Complex (Gupta & Stafford, 2006). In addition, (Andres, 2005) showed that if priority is allowed, the problem can also be NP-Complete.

Furthermore, most of the existing literature assumes that ‘setup time’ is a negligible aspect or part of the work processing time. This, however, may have undesirable effects on the quality of the solution and may not produce an optimal schedule, which in turns affect the performance of production units. Therefore, it is important to incorporate ‘setup time’ in scheduling choices and in instructions to address an additional representative alternative of HFS problems. This paper, hence, aims to address this aspect and proposes an extension to the existing literature by incorporating setup time in scheduling choices. While there are several configurations that explicitly incorporate time, we employ sequence dependent setup time, SDST, which is widely used in the literature. As a result, we propose a hybrid flow-shop with sequence dependent setup time configuration (HFS with SDST) and 'makespan' criterion HFS/SDST/Cmax (Vignier & Al, 1995). In this context, (Johnson &, 1954) develops the mathematical prototypical aimed at dealing with HFS/SDST/Cmax. We follow the mathematical model proposed by (Johnson &, 1954), which is complementary to several problems than HFS and appropriate to the NP-Complete problems.

In this paper, we study the performance of the immune algorithm system (IAS) under multi-agent system. Common existing methods have provided current applications for similar research, which are widely combined to create a new IAS with configurations that are more advanced than the earlier version. In addition, we also compare the performance of our methodology to the most recent applications and methods proposed in the literature of computational intelligence, intelligence-oriented procedures such as PSO (particle swarm optimization) (Li & Al, 2014), GA (Genetic Algorithm)(Vignier & Al, 1999)(Kurtz & Askin, 2004)(Norman & Bean, 2001), LSA (Local Search Algorithm)(Ren & Al, 2015), NEH_H (Nawaz Enscore and Ham)(Liu & Al, 2016), simulated annealing, Tabou search, amongst others.

The remainder of this paper is organized as follows. Section 2 provides a critical literature review on the HFS with SDST. Section 3 introduces the method we propose for scheduling, which suggest using MAS with IAS technique as well as proposing a novel AI technique. We illustrate the application of our approach in Section 4. This also include testing its performance in comparison to other existing approaches. Finally, Section 5 concludes.

2 REVIEW OF RELATED LITERATURE

The purpose of this section is to survey the main literature related to HFS and Cmax criterion. The literature highlights both studies that employ SDST and those that do not. In this context, (Riane & Al, 1998) represent an earlier contribution to the literature of HFS, which provides a general treatment of optimal two and three- stage production schedules with setup times. Subsequent literature extended the concept of scheduling to include hybrid HFS in various settings most notably by (Gao & Al, 2006) and (Salvador, 1973). (Yoshida & Hotomi, 1979) proposed a B&B technique to minimize the Cmax criterion. It has been shown that the optimal schedule is obtained when Cmax reaches its minimum.

This is, however, has been not always achievable as HFS problems may vary with context and nature of the problems leading to the rise of various heuristic solutions. For example, (Portmann & Al, 1998) and (Allahverdi & Al, 2008) develop two methods based on Johnson's algorithm, while (Gupta & Stafford, 2006) proposes a novel technique based on the longest heuristic processing time (LHPT). In addition, bio-inspired methods have become increasingly popular with methods ranging from metaheuristic and evolutionary methods to Genetic Algorithm (GA) methods as proposed by (Vignier & Al, 1999) and (Kurtz & Askin, 2004). These methods have been implemented independently or combined with the existing standard heuristic methods. For example, (Serifoglu & Ulusoy, 2004) extends the B&B suggested by (Gao & Al, 2006) by introducing a number of heuristics at the beginning of the search and GA to increase the performance during the search in the B&B process to find an improved value of the upper bound. In (Yoshida & Hotomi, 1979), discuss scheduling under a hybrid three-stage flow shop problem that follows a specific structure that consist of a machine in the first stage, two machines in the second stage, and one machine in the third stage. (Aghezzaf & Al, 1995) propose two heuristic procedures based on the Cmax criterion.

One particular case of interest HFS problems is when SDST constraint is included. This latter is a common constraint in various industries and reflect a realistic representation by considering cases as common as the change of color, formats, and others, in the production process. This may have been one of the least dealt with aspects in historically; yet it has significant implications and received a great attention in recent literature. It worth noting that various solutions to this type of problems proposed so far in the literature are generally approximate solutions and not exact.

Recent literature proposed various approaches to solve the HFS with SDST problem. For example, (Zandieh & Al, 2006) proposed to measure measured scheduling HFS using Cmax, through no buffers among floors and no SDST. B&B-bound methods are practical to control the best variation schedule compared to Cmax. The flow-shop problems considered by (Yaurima & Al, 2009) distinguish setup times from processing times. In addition, (Gupta & Stafford, 2006) reviews a number of HFS with SDST problems and possible methods to solve them, with emphasis on the two-machine flow-shop problem. (Gupta,2000) discuss

HFS scheduling problem that consider a feasible model with two-agents where the exclusion periods for all Works in all floors are separated from processing times. In (Naderi & Al, 2010) two techniques are proposed to minimize the Cmax in textile manufacturing.

The difficulty of the problem, however, has grown with many approaches leaning towards heuristic or hybrid metaheuristics. (Vignier & Al, 1995) offer a complete survey of FS problems including HFS problems concerning set up time, with and without SD. (Norman & Bean, 2001) consider issues concerning HFS and propose two algorithms to deal with the hybrid and flexible set up of the problem. (Hung & Ching, 2013) Suggest an AI approach and show that it outperforms the RKGA proposed by (Norman & Bean, 2001). Moreover, group scheduling, within the context of HFS problem, is also presented in (Kurtz & Askin, 2004).

(Ruiz & Al, 2005) Study a Genetic Algorithm for the HFS with SDST, and availability limits. Furthermore, (Ebrahimi & AL, 2017) suggest two innovative algorithms that treat machines working simultaneously and address factors of scheduling problem. (Srikar & Ghosh, 2014) measure a variation FS with SDST in their MILP model, which used less variables than earlier models. (Srikar & Ghosh, 2014) Employed consequence variables that capture whether a Work is scheduled for a certain period of time before another Work.

(Wang & Al, 2019) study the problem of two-floor no-wait HFS problem and examine in which the SDST at the first stage is considered. Various methods are developed including a BB method, TS and three heuristic. Findings from computational experiments show that the proposed methods are efficient.

(Li & Al, 2014) suggest a hybrid algorithm that combines PSO and ILS approaches to solve HFS with PM activities. The proposed approach examines various crossovers operators and mutation operators.

(Almeder & Hartl, 2013) attempt to assess the efficiency of GA algorithms, via the minimization of Cmax as the optimization criterion, in a more realistic setting. Furthermore, (Almeder & Hartl, 2013) propose two advanced GA, which showed superiority to those existing metaheuristics algorithms

The above discussion suggest that the there is a little work done in the literature of HFS with SDST, which justifies the need for further research and development in this area. Thus, our aim is to we aim to contribute to the literature by developing a Hybrid Multi Agent and Immune algorithm approach to solve this complex problem.

3 MULTI-AGENT IMMUNE SCHEDULING ALGORITHM

Under the method we propose, MAISA, let antigens be the objective function (the Cmax function) to optimized and antibodies denote candidate solution. . MAISA approach is developed based on clonal selection and particularly inspired through the suggested CLONALG technique. MAISA too is founded on the standard of the maturing affinity of the artificial immune system, AIS.

MAISA begins with a number of antibodies, called a population. The population is improved by a set of operators until the stopping rule – criterion – is met.

The iterative process of MAISA to generate the population is described as follows:

- First, we apply an acceleration mechanism. Using acceleration mechanism, candidate solutions with better aptitude are moved to the next population.
- The antibodies of the new population are multiplied by mutation and crossover operators.
- The antibodies submitted to operators are selected using a selection function, which uses the optimal value of each antibody as well as applying affinity computation.
- Candidate solutions with better Cmax have higher likelihood of being selected to produce the next generation of candidate solutions.
- The aim of this mechanism is to ensure the next generation will contain a large number of candidate solutions with good properties.

On the other hand, the computation of affinities between antibodies consists of process of eliminating the similar antibodies. This achieves the following task:

A candidate solution is assigned a lower probability if its value is greater than the affinity threshold value (AT). The assigned probability is computed by multiplying the previous probability obtained from large number of of antibodies with a lower operators, which is the affinity setting (AS). This will lower the likelihood of being selected.

3.1 MAISA Flowchart

Fig. 1 illustrates a detailed flowchart of the proposed MAISA approach for HFS with SDST scheduling. Each agent under MAISA approach performs the same process, which involves the computation of the objective function (Fitness). This latter corresponds to the value of the makespan.

The agent performs the following operations:

- Construct a set of antibodies (SP) as the initial population.
- Compute the correct antibody values.
- Compute the affinity values of the antibodies.
- Perform an acceleration mechanism.
- Select two parents using a selection mechanism.
- Make a uniform crossover (CUS).
- Make a single-point mutation (SPM).

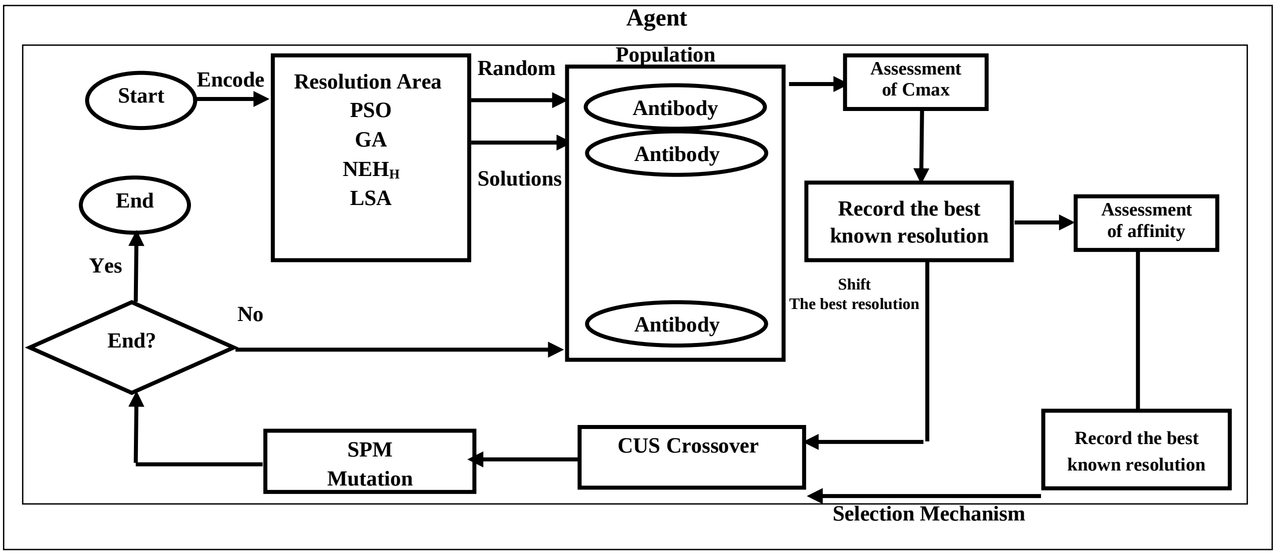

Fig. 1 MAISA Flowchart for HFS with SDST scheduling

3.2 MAISA Algorithm

The algorithm of the MAISA method is presented in Table 1 as follows:

Table 1.MAISA Algorithm

MAISA Algorithm
For all Agent **do**
Generate a set of antibodies (SP) as an initial population.
While ending condition is not attained **do**
Compute the values of antibodies
Determine antibody affinity values
Set up the fitness endpoints of the antibodies
Perform an acceleration mechanism
For I = Nr + 1 to SP **do** //Nr individuals
Select two parents by a selection mechanism
Make the CUS crossover and produce the *i*-th successor
If rand <Pm **then**
Create a SPM mutation on the *i*-th successor
End if
End for
End While
End for

3.3 Coding scheme and operators

3.3.1 Parameter Setting

The initial parameters need to include the following: the size of the population(SP), the number of solutions relative to the population (Qp), the probability of mutation (Cm), the mass of similarity structures (M_1, M_2, M_3, M_4), the affinity threshold (AT) and affinity setting (AS).

3.3.2 Generating the initial population

- Producing arbitrarily an initial populace of (SP-4) antibodies.
- Create antibodies by PSO.
- Create antibodies by GA.
- Create antibodies by LSA.
- Create antibodies by NEHH.

3.3.3 Hazard key

The Coding technique we adopt in this work is Hazard key (HK), which was illustrated by (Hung & Ching, 2013) and extended and further generalized by (Ruiz & Al, 2005).The key advantage of HK is in its simplicity and flexibility. The HK algorithm is described as shadows in Table 2 as follows:

Table 2. HK Algorithm

HK Algorithm

For every Work **do**
Allocate a real number whose integer part is the number of the machines to which the Work is allocated
The fractional part is used to order the Works assigned to each machine.
Only Hazard numbers are utilized for floor1.
They define the Work arrangement and task just for floor 1.
For all the following floors j, {j = 2, …, n} **do**
The Work sequence is determined by the earliest times of completion of Works in the previous floor.
The machine allocation rule is the first available machine.
End for
End for

For this study, we need to generate found random numbers from the uniform distribution within the range $(1, 1 + n_1)$ towards the initial floor as shown in Fig.2:

2.86	**1.68**	**2.42**	**1.43**

Fig.2. Encoded resolution by HK representation.

It is well recognized that the first solutions can greatly affect the final results obtained by MAISA. We consequently produced first solutions as follows: four solutions are created by the technique PSO, GA, NEHH and LSA, and the remainder is produced arbitrarily. Antibodies with low Cmax are desired and, consequently, a sum of antibodies (Nr) with the lowest value of Cmax are automatically copied to the next generation. This mechanism is known as reproduction. The remaining antibodies (SP - Nr) % or offspring are produced by crossing two other sequences or parents by an operator called a crossover operator. Crossover operators must avoid generating non-feasible solutions.

3.3.4 Assortment device

For the choice of parents experience crossover, we utilize the arrangement assortment that might be defined as shadows:

Table 3. Assortment Device Algorithm.

Assortment Device Algorithm

The entities of the present population are principal organized conferring to their Cmax.
For every antibody **do**
Allocate a uniform possibility so that the greatest resolutions are additional probable to be designated.
End for
- Entities are arbitrarily designated as parents to acquiesce to factors founded on their possibilities.

3.3.5 CUS Crossover

The objective is to produce a superior descendant that is to approximately, to generate enhanced arrangements through merging the parents. Our crossover is uniformly set (CUS), since it has presented its success in the works of (Ruiz & Al, 2005). It is required to require that the study of CUS by hazard keys and presented as shadows:

Table 4. CUS Algorithm.

CUS Algorithm

For every Work **do**
Produced a hazard numeral among (0, 1).
End for
If the value is fewer than 0.7 **then**
the HK of the Work conforming to parents 1
is derivative to the child,
else, the CA of parents 2 is designated.
End if
Works are categorized conferring to the climbing order of the HK.

The process is demonstrated arithmetically by relating it to an instance through n = 6 and m = 2 displayed in Fig. 3.

Parents 1	2,86	1,78	2,82	1,43	2,74	1,50
Parents 2	1,24	2,95	2,35	1,85	2,33	1,77
Hazard N°	0,36	0,43	0,79	0,21	0,98	0,33
Child	2,86	1,78	2,35	1,43	2,33	1,50

Fig. 3. Technique for CUS crossover practical to an instance with n = 6 and m = 2.

3.3.6 SPM Mutation

A mutation mechanism is applied to nip the arrangement, i.e. produce a novel order, however analogous.

The key resolution of the use of mutation is to escape merging to a local optimum and differentiating population. The mutation mechanism can too be seen as an easy procedure of limited exploration.

Various scientists have determined that only the SPM mutation can offer improved consequences than additional mutations such as SWAP or overturn.

Consequently, we practice single point mutation (Ebrahimi & AL, 2017). SPM technique can be specified as shadows:

- HK of Work an arbitrarily selected at hazard is restarted. Fig. 4 displays a descriptive explanation that transforms.

Before Mutation	1,43	2,43	1,88	2,53	2,67	1,75
After Mutation	1,43	1,86	1,88	2,53	2,67	1,75

Fig. 4. SPM Mutation to an instance through n = 6 and m = 2

3.3.7 Fitness

In general, the objective function, Fitness, includes one or more performance indicators that capture the effectiveness of an antibody. The candidate antibodies are first transformed into a valid schedule. Then, they are evaluated using an objective function in order to obtain their fitness values. Higher fitness values are desired while SIA is dealing with a maximization problem. On the other hand, if the problem is a minimization problem, the objective function is modified in such a way as 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 low fitness value.

For an antibody i, the fitness function can be defined as: follows:

$$f(i) = \frac{\frac{1}{C_{max(i)}}}{\sum_{i=1}^{SP} \frac{1}{C_{max(i)}}} \quad (1)$$

where *f(i)* and Cmax(*i*) are the fitness value and the makespan of the antibody *i*, and *SP* is the size of the population.

3.3.8 Affinity

The selection mechanism depends on the probabilities calculated by both the good value and affinity value. In order to better assess the effectiveness of the AIS affinity function, we apply uniform crossover, CUS, and the single-point mutation, SPM. In this context, the affinity assessment of SIA increases the diversity of antibodies in a population and therefore allows for extensive search space in exchange of computing time. The main challenging question is whether for such a complex problem the affinity calculation is worth its cost. This is to say the proportion of the time that it takes for the algorithm to complete the computations. In the following, we discuss the computation procedure of the affinity.

3.3.9 Affinity computation algorithm

To compute the affinity, the antibodies are compared to the best known antibody (BKA) obtained so far. Affinity simply expresses the similarity between an antibody and BKA. The theory of affinity is used to estimate the probability of data reoccurrence. (Norman & Bean, 2001) defines the entropy of the data, H(x), of a discrete random variable X = {x1, x2,. . . , Xn} with probability mass function P (X = x) = pi, i = 1, 2,. . . , N such that:

$$H(x) = -\sum_{i=1}^{n} p_i \log(p_i) \quad (2)$$

The procedure of data entropy provides the resemblance of the antibody i (or arrangement i) to a reference antibody (or arrangement). This can be computed using the formula:

$$aff(i) = \frac{1}{1+\frac{1}{k}\sum_{j=1}^{k} h_{ij}} \quad (3)$$

where aff (i) is the resemblance size, k is the size of the population, and $h_{ij} = -p_{ij}\log(p_{ij})$.

The antibody Affinity for every value in our problem is considered as follows:

Table 5. Computation of Affinity Algorithm

Calculation of Affinity Algorithm

For every Work j **do**
Compute a resemblance report
End for
- The average ratio of all Works in an antibody is distinct as the general Affinity value of that antibody
- The similarity ratio of Work *j* is computed by comparing the position of this Work by matching the antibody and BKA.
- The order of the Work and the assignment for the subsequent floors are found by the following instructions:
• The earliest completion time for Works on the previous floor and the first available machines, respectively.
• The similarity ratio is found from the location of Work *j* on floor 1.
End

Every condition has a mass that displays its rank (represented via Mi, i = {1, 2, 3, 4}). If a condition is met, the Work j obtains its mass, and the total rate of the masses is the percentage of completed resemblance Work j. After computing the similarity ratio of each Work, the average similarities of the works are used as the affinity of the candidate antibody.

4 RESULTS AND DISCUSSION

The aim of this section is to assess our multi-agent approaches based on the AIS. This include comparing our approach to other existing heuristics such as PSO, GA, LSA and NEH_H. All the experiments are implemented in MATLAB 7, using an Intel Core 2 Duo with 3.0 GHz and 4 GB RAM. The stopping rule used during the test with all instances of heuristics is set at a CPU with limit of $n^2 \times m \times 2.5$ Ms **(***n* is the number of Works and *m* is the number of floors**)**. This stopping rule allows more time than the number of works (or increases number of floors) and is sensitive to increases in the number of works than the number of floors

We use the comparative fraction deviation (CFD) as a measure to compare between the performance of all the methods implemented. The correct solution obtained for each instance (called Minsol) is computed by the one of the algorithms. The CFD is obtained using the following formula:

$$\mathbf{CFD} = \frac{\mathbf{Alg_{sol}} - \mathbf{Min_{sol}}}{\mathbf{Min_{sol}}} \times \mathbf{100} \quad (4)$$

where Alg_{sol} is value of the objective function (Fitness) for a given algorithm for an instance. Lower values of CFD refer to better performance.

4.1 Parameter Setting

It is recognized that the varied heights of factors affects the quality of the solutions obtained through MAISA. We define a set of operators on the number of Agent (NA), the size of the population (SP), the number of solutions copied directly to the subsequent population (Nr), the possibility of mutation (Pm), the mass of features similarity (M_1, M_2, M_3, M_4), the Affinity Threshold (AT), and Affinity Setting (AS). Tab. 6 display the heights measured.

Table 6. MAISA Heights factors.

Factors	N° of Height	Heights
Number of Agent	3	10,20,30
Size Population	3	50, 100, 150
(Nr, Pm)	3	(1, 0.10), (2, 0.15), (3, 0.20)
(M_1, M_2, M_3, M_4)	2	(0.20, 0.10, 0.35, 0.35), (0.30, 0.20, 0.25, 0.25)
(AT, AS)	3	0.40, 0.60), (0.60, 0.50), (0.80, 0.50)

A set of 30 instances in 3 groups 3 (n = 50, 80, 100) is generated and solved by the algorithms. After analyzing the results obtained from MAISA, we select: Later examining the consequences MAISA we select NA=30, Nr = 20, Pm = 0.20 and SP = 100, M1= 0.20, M2 = 0.10, M3 = 0.35 and M4 = 0.35, AT = 0.80 and AS = 0.60. In other words, if the affinity value exceeds 0.6, the probability of being selected the selection mechanism is multiplied by 0.5.

Table 7. Parameters and heights

Parameters	Heights
Number of Works	50, 80, 100
Number des floors	3, 6, 9
Dispersal of a machine	Constant : 2 Variable : U (1, 3)
Treating time	U (1, 90)

4.2 Data generation

The data required to resolve this problematic contains the data of the production scheduling with sequence-dependent setup times.

The data comprises the total of Work (n), total of floors (m), the total of identical machines at respectively floor (mi), variety of treating period (Pj,i) and period advances. Is n = {50, 80, 100}, and m = {3, 6, 9}. To group the total of machines at every floor, we have two groups. In the primary, we have a total of unchanging hazard dispersals of machines reaching from one to three machines per floor, and in the next, we have an enduring number of two machines on every floor. Times ready for floor 1 are group to 0 for completely Works. Periods prepared to floor (i + 1) is the finishing time in floor i, so these data should not be produced. Tab. 7 displays the parameters and their heights.

4.3 Experimental results

We report in this section the results of performance of MAISA compared to PSO, GA, LSA and NEH_H for the scheduling of a HFS with SDST using the comparative fraction deviation (CFD).

The experiments are based on all combinations of number of Works (n) and floors (m). Tab. 8 reports the average CFD for each setting of m and n and the overall average.

Table8. Average CFD for the techniques collected by n and m

n	m	Algorithms				
		MAISA	**PSO**	**GA**	**LSA**	**NEH_H**
50	3	2.85	3.36	4.59	7.66	7.79
	6	2.34	3.64	4.86	7.18	7.86
	9	2.27	3.37	4.25	7.13	7.95
80	3	1.83	2.80	5.39	6.17	8.68
	6	1.69	2.76	5.80	6.03	8.71
	9	1.28	3.27	4.11	6.22	8.53
100	3	2.10	3.57	4.58	6.75	7.49
	6	2.57	3.61	4.47	6.48	7.68
	9	2.66	3.52	4.81	6.69	7.72
Average		2.17	3.32	4.76	6.73	7.94

The results of the experiments, on average for each combination of n and m

Furthermore, we also conduct a variance analysis, ANOVA, which allows analyzing the differences among group means and variations in a sample. For the purpose of this paper, we compare the average CFD values of each of the methods we implemented. This include the CFD values across features, number of works and number of floors of all the five algorithms reported in Table 8.

Fig.5. depicts the graphical representation of means and minimum important variance (MIV) for each algorithm. Fig 5, in general, illustrates substantial differences across the mean CFD values among the algorithms. It also suggests the MAISA has the lowest mean CFD, which is within an acceptable range of deviations. Thus, according to Fig. 5, MAISA outperforms the other competing algorithms in terms of their CFD.

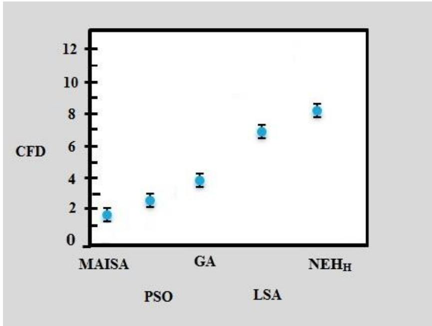

Fig.5. CFD chart mean and MIV intervals (at 95% sureness) for the category of technique

We further estimate the differences across features such as the number of work and number of floors on HFS with SDST scheduling algorithm presentation. We aim to investigate the behaviour of CFD for each the algorithms accounting for the differences across the heights of the features. Fig.6 and Fig. 7 display the graphical representation of the mean CFD for each algorithm under different features for interaction amid techniques deferent number of works and floors s features respectively.

As shown in Fig 6, MAISA outperforms the other competing approaches, where it achieved the lowest mean CFD for all different levels of number of works, in particular when the number of the works is 80. As the number of works increases to 100, LSA's mean CFD improves the most, which makes LSA the most efficient. PSO mean CFD remains relatively stable.

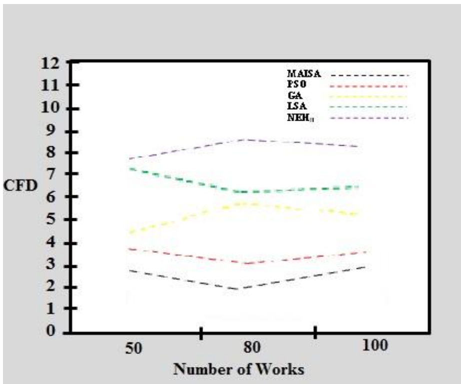

Fig. 6 CFD chart mean for interaction between Aproaches and number of Works feature.

Similarly, Fig7 shows that MAISA remains the preferred algorithm with the lowest mean CFD in all three floors. While its the CFD mean value increases to that of PSO when the number of floors is 6, it falls substantially when the number of the floors is 9. In addition, our findings suggest that unlike PSO, GA and LSA, the mean value CFD of NEHH improves substantially when the number of floors increases to 9,

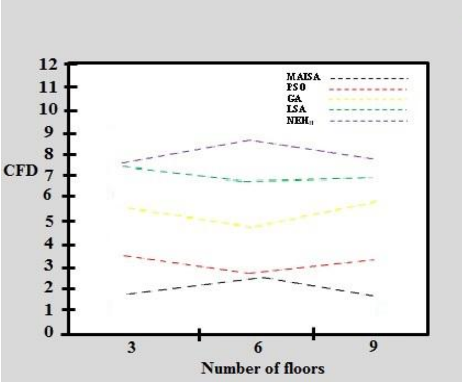

Fig. 7 CFD chart mean for interaction between Aproaches and number of floors feature.

Each heuristic considered in this paper was implemented on the same 3500 dataset. The algorithms MAISA, PSO, GA, LSA and NEH_H were executed 50 times obtaining the minimum average loss over the 50 series of each of the 3500 datasets. The CFDs of all the datasets are presented in Table 9.

Table 9.CFD statistics for Methods.

Methods	CFD			Total of periods Min^X
	Setting	Normal deviation	Max	
MAISA	0.02	0.03	0.16	2460
PSO	0.05	0.06	0.10	2580
GA	0.10	0.07	0.20	2490
LSA	0.41	0.13	0.53	2620
NEH_H	0.56	0.23	0.64	2780

X Using the best of the 20 MAISA runs

MAISA has the lowest values for loss statistics and finds the minimum loss schedules more frequently than other heuristics. The variation observed within the 30 series of MAISA, PSO, GA, LSA and NEH_H will be discussed below. A hypothesis test was performed. The results indicate that there is a significant difference between the average responses.

The difference realized inside the 30 MAISA and PSO runs will be debated advanced. A theory trial was achieved. The consequence designates that there is important variance among the purpose answers. Table 10 displays the essential computational time for MAISA against PSO.

Table 10. PSO solution time versus MAISA.

Problem size	Resolution time				
	Reduction		Comparable	Upsurge	
	Problem Ratio (%)	Average Reduction (%)	Problem Ratio (%)	Problem Ratio (%)	Average Upsurge (%)
Small	35	13.5	55	10	9.8
Medium	15	12.6	45	40	15.5
Large	5	5.2	15	80	18.8

Based on this data, we can understand the solution time for MAISA will meaningfully increase as problem size increases. This is practical since makespan will meaningfully increase as the problem size increases. Table 11 displays the makespan value for MAISA versus PSO

Table 11. PSO makespan versus MAISA.

Problem size	Makespan value				
	Reduction		Comparable	Upsurge	
	Problem Ratio (%)	Average Reduction (%)	Problem Ratio (%)	Problem Ratio (%)	Average Upsurge (%)
Small	55	20	45	0	-
Medium	70	37	30	0	-
Large	80	35	20	0	-

5 CONCLUSIONS

The aim of this paper is to propose a new an Immune Algorithm to solve the HFS with SDST problem on a multi-agents system. For this purpose, we applied advanced methods (operator), the Hazard key for coding, ranking selection, CUS crossover and SPM mutation.

In addition, we presented a new Empathy computation procedure. This procedure is based on computing similarity ratio of the antibodies, for which we proposed a criterion-based algorithm to compute the report's similarity.

We also conducted a simulation based exercise to compare the performance of the new method proposed, MAISA, to that of some existing methods including PSO, GA, LSA and NEHH. The performance benchmark suggested in this paper is based on CFD, which was estimated for all algorithms under different settings including number of works (ranging from 30 to 100) and floors (ranging from 3 to 9). Our findings suggest that MAISA outperforms all other algorithms consistently in all different settings.

6 ACKNOWLEDGEMENTS

The authors would like to thank the editor and two anonymous referees for their constructive and insightful comments on an earlier version of the paper. This work was carried out as part of the research project in the research laboratory of automation and production, Department of Industrial Engineering at the University of Batna, Algeria.

7 REFERENCES

J.N.D. Gupta and J. E. F. Stafford. (2006). *Flow shop scheduling research after five decades*, European Journal of Operational Research, 169, 699–711.

S. A. Brah. (1999). *Heuristics for scheduling in a flow shop with multiple processors*, European Journal of Operational Research, 113(1):113-122.

J.N.D. Gupta.(2000). *Two phase hybrid flow shop scheduling problem, Journal of Operational Research Society*, 39, 359–364.

T. S. Arthanari and K. G. Ramamurthy. (1971). *An extension of two machines sequencing problem*, Operations Research, 8, 10–22.

I. Ribas, R. Leisten and J.M. Framiñan. (2010). *Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective*, Computers and Operations research, 37 (8), 1439-1454.

L.N. de Castro and J. Timmis. (2002). *An artificial immune network for multimodal function optimization*, Evolutionary Computation, CEC'02,in: Proceedings of the 2002 Congress, pp. 699–704.

T. Kis, and E. Pesch. (2005). *A review of exact solution methods for the non-preemptive multiprocessor flow shop problem*, European Journal of Operational Research, 164(3), 592-608.

V. Y. Shen and Y. E. Chen. (1972). *A scheduling strategy for the flow shop problem in a system with two classes of processors*. Conference on Information and Systems Science. Proceedings, 645-649.

C. Sriskandarajah, and S. P. Sethi. (1989). *Scheduling Algorithms for Flexible Flow shops - Worst and Average Case Performance*, European Journal of Operational Research, 43(2), pp.143-160.

M. C. Portmann, A.Vignier, D. Dardilhac and D. Dezalay. (1998). *Branch and bound crossed with GA to solve hybrid flow shops*, European Journal of Operational Research, 107(2), 389-400.

J.A. Hoogeveen, J.K. Lenstra and B.Veltman. (1996). *Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard*, European Journal of Operational Research, 89, 172–175.

A.N. Allahverdi, T.C.E. Cheng and M. Y. Kovalyov. (2008). *A survey of scheduling problems with setup times or cost*. European Journal of Operational Research, 187, 985–1032.

F. S. Serifoglu and G. Ulusoy. (2004). *Multiprocessor task scheduling in multistage hybrid flow shops: a genetic algorithm approach*, Journal of the Operational Research Society, 55(5), 504-512.

C. Andres. (2001). *Production Programming in Hybrid Flow Workshops with Times of Change of Departure dependent on the Sequence*. Phd dissertation, presented at University Polytechnic of Valencia (Spain).

S. R. Hejazi and S. Saghafia, Flow shop scheduling problem with makespan criterion: a review, *International Journal of Production Research,* 43(14), 2895-2929.

A.Vignier, J.C. Billaut and C. Proust. (1996). *Flow-related formal problems hybrid shop. State of the art*, Study Days: affectation and Ordonnancement. CNRS / GdR Automatic, GT3, 7-47, Tours.

A.Vignier, J. C. Billaut and C. Proust. (1999). *Hybrid flow shop scheduling problems: State of the art*, Rairo-Recherche Operationnelle-Operations Research, 33(2), 117-183.

S.M. Johnson. (1954). *Optimal two and three-stage production schedules with setup times included,* Naval Research Logistics Quarterly, 1, 61–67.

F. Riane, A. Artiba and S. E. Elmaghraby. (1998). *A hybrid three-stage flow shop problem: efficient heuristics to minimize makespan*, International Journal of Production Research, 40(17), 4353-4380.

J. Gao, M. Gen, and L. Y. Sun. (2006). *Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm*. Journal of Intelligent Manufacturing, 17(4), 493-507.

M.S. Salvador. (1973). *A solution to a special case of flow shop scheduling problems*, in: S.E. Elmaghraby (Ed.), Symposium on the Theory of Scheduling and its Applications, pp. 83–91,(Springer, Berlin.

T.Yoshida and K. Hotomi. (1979). *Optimal 2-stage production scheduling with setup times separated*. AIIE transactions, 11 (3), 261 -263.

E. A. Aghezzaf, A. Artiba and S. E. Elmaghraby. (1995). *Hybrid Flow Shop: an LP based heuristic for planning level problems*, ETFA Proceedings, 551-559.

M. E. Kurtz and R. G. Askin. (2004). *Scheduling flexible flow lines with sequence-dependent setup times*, European Journal of Operational Research, 159(1), 66-82.

M. Zandieh, S. M. T. F. Ghomi and S. M. M. Husseini. (2006). *An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times*, Applied Mathematics and Computation, 180(1), 111-127.

V. Yaurima, L. Burtseva and A. Tchernykh. (2009). *Hybrid flow shop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers*. Computers & Industrial Engineering, 56(4), 1452-1463.

B. Naderi, R. Ruiz and M. Zandieh. (2010). *Algorithms for a realistic variant of flow shop scheduling*, Computers & Operations Research, 37(2), 236-246.

B. A. Norman, and J.C. Bean. (2001). *A genetic algorithm methodology for complex scheduling problems*. Naval Research Logistics, vol. 46, pp. 199-211.

T.S.L. Hung and J.L. Ching. (2013). *A case study in a two-stage hybrid flow shop with setup time and dedicated machines*, International Journal of Production Economics (86), 133–143.

E.F. Stafford. (2012). *On the development of a mixed-integer linear programming model for the flow shop sequencing problem*, Journal of the Operational Research Society 39 (12), 1163–1174.

B.N. Srikar and S. Ghosh. (2014). *A MILP model for the N-job, M-stage flow shop with sequence dependent set-up times*, International Journal of Production Research 24 (6), 1459–1474.

V. Yaurima, L. Burtseva and A. Tchernykh. (2009). *Hybrid flow shop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers*, Computers & Industrial Engineering, 56(4):1452-1463.

M. Ebrahimi, S.M.T. Fatemi and G. Karim.(2017). *Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates*, Applied Mathematical Modelling ,38(10), 2490-2504.

C. Almeder and R.F. Hartl. (2013). *A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer*, Int. J. Prod. Econ, 145: 88-95.

S. Wang, X. Wang and L. Yu. (2019). *Two-stage no-wait hybrid flow-shop scheduling with sequence-dependent setup times*, International Journal of Systems Science: Operations & Logistics, 57(5).

T. Ren, M. Guo, L. Lin and Y. Miao. (2015). *A Local Search Algorithm for the Flow Shop Scheduling Problem with Release Dates*, Discrete Dynamics in Nature and Society, 17(3).

J.q. Li ,Q. Pan and K. Mao. (2014). *Hybrid Particle Swarm Optimization for Hybrid Flowshop Scheduling Problem with Maintenance Activities*, The Scientific World Journal, 20(4).

W. Liu, Y. Jin and M. Price. (2016). *A new Nawaz–Enscore–Ham-based heuristic for permutation flow-shop problems with bicriteria of makespan and machine idle time*, Engineering Optimization, 48(10):1-15.

R. Ruiz, C. Maroto and J. Alcaraz. (2005). *Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics*, European Journal of Operational Research, 165(1):34-54.