

# A Novel Artificial Immune Algorithm for Solving the Job Shop Scheduling Problem

Mohsen Afshari\*

Department of Computer, Science and Research  
Branch, Islamic Azad University, Khuzestan, Iran

Hedieh Sajedi

Computer Science Faculty, Tehran University, Iran

## ABSTRACT

Scheduling problems are difficult types of production arrangement problems that enumerated among NP-Complete problems. Some of evolutionary algorithms such as Genetic Algorithm, Ant Colony Optimization etc. have been used to solve this problem. In new years, Artificial Immune Algorithm is used to solve optimization problems such as routing and scheduling. One of complex scheduling problems is Job-shop Scheduling problem. In this article we use immune system concepts of human body, to implement a new artificial immune algorithm for solving Job-shop scheduling problem. A new population generation method was proposed based on G&T algorithm. We use two mutation methods, namely Shift Change method and Inverse method in Job-shop scheduling for first time. Moreover, we describe a vaccination method named MCV, to make maximum advance in solutions, and then achieve to more than one optimal solution concurrently and release from local optimum. Finally, we test our method on the very famous benchmark of JSP, namely FT06, then show experimental results and get some conclusions.

## General Terms

Job-shop Scheduling Problem, Artificial Immune Algorithm, Vaccination

## Keywords

Artificial Immune System, Job-Shop Scheduling Problem, Mutation, Vaccination, Receptor Editing, Clonal Selection

## 1. INTRODUCTION

Scheduling problem is concerned with devotion of some resources to tasks to optimize some objective functions[1]. This problem has complex combination of goals, resources and constraints, so, this kind of problems are main NP-complete problems [2].

Recently, Artificial Immune System (AIS) is used to solve problems from different fields such as Robotic [3], Anomaly Detection [4], Combinatory Optimization [5], Learning [6], etc. One type of optimization problems is scheduling, and one of the very most common models in field of scheduling is that of the Job-shop scheduling. This problem belongs to NP-hard problems, whose optimal solution is difficult to achieve [7]. Some evolutionary methods such as Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Partial Swarm Optimization (PSO), Tabu Search (TS) etc. must be used to solve this problem. But these algorithms have a few lacks. For example GAs have two main drawbacks. One of them is lack of local search ability and the other is the premature convergence [8].

But, AIS is able of variety in the development of antibodies [9]. This algorithm is superior to all the other evolutionary

algorithms, and it is not encountered with drawbacks of such algorithms.

In this paper an Immune Algorithm for solving the JSP is proposed. We try to use best approach for all the steps of proposed algorithm. Several procedures such as, Population Generation, Clonal Selection, Mutation, Vaccination and Receptor Editing are designed in order to construct a good algorithm. So we can see an admissible proficiency in overall algorithm. The rest of this paper is organized as following:

In section 2 we present an introduction to artificial immune systems. In section 3 we addressed a history of applying AIS to JSP. The JSP is introduced in section 4. In section 5 we show our proposed immune algorithm. We demonstrate the experimental results of proposed algorithm in section 6, and finally, conclusion is presented in section 7.

## 2. AN INTRODUCTION TO ARTIFICIAL IMMUNE SYSTEMS

Artificial Immune Systems(AISs) are relatively young emerging techniques, which find out, gain and apply several biologically inspired immune mechanisms, in order to computational problem solving[10].

Application areas that have been addressed by AIS techniques can be broadly summarized as (1) Learning (2) Anomaly Detection and (3) Optimization. Thus, learning includes clustering, classification and pattern recognition, robotic and control applications; Anomaly Detection includes fault detection and computer and network protection applications, and Optimization includes real-world problems which basically makes combinatory and also numeric function optimization[11].

When the immune context is introduced into the computational intelligence field, the expression “antigen” is usually used to mean an optimization problem and “antibody” is used to indicate the solution to the optimization problem[12].

Clonal proliferation produces some replica of selected antibodies depend on affinity of them, i.e., the greater affinity with the antigen, makes more replica for antibody.

In AIS, the process of mutation is emulated by swapping the position of two or more genes of an antibody. There are many mutation types. For example Murata and Ishibuchi[13] presented four mutation methods, Adjacent two-job change, Arbitrary two-job change, Arbitrary three-job change, and Shift change. The use of mutation is to develop a solution, i.e., make a good solution still better[14].

The vaccination process when used in AIS, usually partially changes the related solution according to a-priory knowledge of the problem that is being solved. The method generates excellent solutions(higher affinity) with higher possibility than if the individuals is mutated randomly[15].

### 3. HISTORY

We address some of researches that are relevant with this issue namely applying AIS to JSP.

One of the first papers in field of scheduling was published by Emma Hart and Peter Ross in 1999[16]. In this paper, an algorithm with two steps was designed. In first step, existing scheduling pattern is found and then, in second step, according to founded pattern, a new schedule is generated. Aickelin et al.[17] showed that schedule generated using AIS, is stronger than that one generated using GA. Coello et al.[18] used clonal selection and hyper mutation principles in order to solve the JSP. The results of their method showed that using AIS makes considerable improvement in solution, and number of iterations to get to the admissible solution is less than GA. Xu et al.[19] represented an immune algorithm that used vaccination, mutation and cross-over operations. An important improvement of their algorithm was generation of effective vaccine to develop the solution. Zuo et al.[20] expressed a division of scheduling types, and stated that optimal solution is in active schedule group. They used clonal selection principle and immune network theory, and applied mutation and cross-over operations. Another properties of their algorithm are use of niche technology, chaos variables and receptor editing operation. They tested proposed algorithm on FT06 benchmark, and results showed effectiveness of the algorithm. After it, Ge et al.[21] designed an immune algorithm that used G&T method in order to generate initial population. They planned models of vaccination and receptor editing to improve the solutions. Results showed effectiveness of their algorithm in consumed time and some other performance measures. Then, Yaqin et al.[7] studied JSP with multiple objectives. They produced random solutions and added them into population. Also, they used mutation and cross-over operations. Experimental results indicated ability of that algorithm. Tsai et al.[9] used Taguchi method to come together the operations in order to achieve to the better solutions. Also, they used clonal selection and hyper mutation principles. Luh and Chueh[22] used gene division and recombination methods in order to make a balance between search and production of antibodies. They applied six operations on antibodies. They employed clonal selection principle too. Experimental results showed high capability of their algorithm on finding the optimal solution. Hong[23] applied chaotic mechanism into clonal selection principle in order to improve the performance of algorithm. Also, he used niche technique, immune network theory, and negative selection in order to keep the best solutions. His algorithm has high convergence speed and avoids blind search. Zhang and Wu[12] used AIS with Simulated Annealing algorithm. They used a fuzzy system in order to determine bottleneck jobs. Then they designed immune mechanisms. Also, they applied vaccination and immune selection into the algorithm. Wei et al.[24] used Genetic Algorithm with AIS. Used operation was two types: Genetic operations namely mutation and cross-over, and Immune operations namely vaccination and immune selection. Experimental results showed that the algorithm has high performance and higher convergence speed. Ma et al.[2] proposed an immune algorithm that used a vaccination method based on prior knowledge of problem. Also, they applied immune selection operation and used it into Genetic Algorithm in order to use local information to find the optimal solution and prevent generation of repetitive solutions.

### 4. PROBLEM DEFINITION

Job Shop Scheduling is a resource allocation problem subject to allocation and sequencing constraints. In this problem there are  $n$  jobs and  $m$  machines. Each Job <sub>$i$</sub>  consists of  $n_i$  operations. The goal is to find a potential schedule that minimizes some of performance measures depending on the completion time  $C_i$  of the last operation of each job [25]. One of these measures is total duration time required to complete all the jobs. This time is called makespan.

Then, the problem is to assign each operation to a proper machine (routing problem), and to organize the operations on the machines (sequencing problem) in order to minimize the makespan [26].

The size of search space with  $n$  jobs and  $m$  machines is show in equation (1) [19]:

$$\text{Space size} = \frac{(n * m)!}{m!^n} \quad (1)$$

In this problem, it is possible that the job entered the system, doesn't need to all of the machines, or needs one specific machine more than once. These situations may make JSP more complex. But usually, JSP is considered with following rules and assumptions:

1. There is no constraint on processing time of each job on machine.
2. There is no run out time. Each job should be processed to its finishing point [27].
3. All jobs enter the system at zero time and can start to execution immediately. It means that there is no constraint on assignment time.
4. All jobs entered the system, must use all of the machines. In other words, if there are  $n$  jobs and  $m$  machines, each job must be consisted of  $m$  operations that each of them will be executed on exactly on one of machines.
5. Order of operations of each job on machines can change from one job to the others. For example, in the environment with 3 machines, job <sub>$i$</sub>  may be processed on machine 1, 3, 2 and job <sub>$j$</sub>  may be processed on machine 2, 1, 3 respectively.
6. Main objective is to minimize the makespan. Namely, jobs must be organized on machines in order to minimize the overall execution time of jobs.
7. One machine can process at most one operation at a time.
8. Every job can be performed just on one machine at a time [28].
9. Operations can't be interrupted, which means operation execution is Atomic.
10. There is not any machine crash, which means all the machines are always available [12].
11. Machines may be idle during overall execution process.
12. The only limitation in this problem is that of the priority order of the operations of each job. A job has to follow the order of operations assigned to it and can't contravene that one constraint [14].

### 5. PROPOSED ARTIFICIAL IMMUNE ALGORITHM

Based on the immune system mechanism of biological organisms, such as diversity, learning, memory etc., an immune algorithm for solving JSP is constructed. Figure 1 shows the model of proposed algorithm, where the antigen corresponds to the problem to be solved while the antibody corresponds to a solution of the problem.

## 5.1. Details of Proposed Algorithm

### 5.1.1. Recognition of Antigen

We use operation-based presentation to show antigen and antibody, because in this type of presentation each permutation of genes is a candidate solution. The value of a gene in operation-based presentation is in fact the number of job delivered to the machine. For example Figure 2 shows an antibody that according to it, first operation of job2, second operation of job2, first operation of job3, third operation of job2, second and third operations of job3 are delivered to corresponding machines respectively. An antigen consists of numbers of machine and time of operation corresponded to those machines.

### 5.1.2. Initialization of Antibody

In order to generate an admissible population library and, we develop a method based on the G&T algorithm. Describe of the method is as following:

1. Suppose that A is set of the initial executable operation of each job. Then suppose for all operations  $(j,m) \in A$ ,  $S_{jm}=0$  that  $S_{jm}$  is first time that operation  $(j,m)$  can start to execute, j and m is number of job and machine respectively.  
For all  $(j,m) \in A$  calculate  $t(A) = \min(S_{jm} + P_{jm})$ .  $P_{jm}$  is processing time of operation  $(j,m)$ . If  $S_{jm} < t(A)$ , then add m to set M. Then construct set G as following:
3. For all  $m \in M$  do following steps:
  - 3.1. Select a machine  $m^*$ . For all operations  $(j,m) \in A$ , put that operation into set G if it is processed on the machine  $m^*$ .
  - 3.2. For all operations  $(j,m^*) \in G$ , compute first finish time of those operations. Suppose such operation is  $(j^*,m^*)$ .
  - 3.3. Select  $(j^*,m^*)$  from G and add it to schedule.
  - 3.4. Remove  $(j^*,m^*)$  from A, add next operation of job  $j^*$  to A, update  $S_{jm}$ , then go to step 3. Do this loop until there is no unselected  $m \in G$ .
4. Go to step 2.
5. When A is empty, a schedule is generated.

### 5.1.3. Affinity Calculation

Affinity is the fitness rate of antibodies [29]. An antibody with higher affinity with antigen and lower affinity with other antibodies is better than others. It means increase in affinity of antibody-antigen makes positive effect and increase in affinity of antibody-antibody makes negative effect on final value of affinity. Considering above reasons, we use equation (2) to calculate the affinity:

$$\text{Affinity} = \frac{Ab\_Ag}{Ab\_Ab} \quad (2)$$

In above equation,  $Ab\_Ag$  is antibody-antigen affinity and  $Ab\_Ab$  is antibody-antibody affinity. Now we calculate these two values:

$$Ab\_Ag_i = 1 / (\mu_i / \min(\mu)) \quad (3)$$

In equation (3)  $\mu_i$  is makespan of antibody i.

To calculate  $Ab\_Ab$ , we use the following equation:

$$Ab\_Ab_i = 1 / (\sum_{j \neq i}^{1 \sim N} \sqrt{\sum_k (X_k^i - X_k^j)^2}) \quad (4)$$

In equation (4)  $X_k^i$  is k-th gene of i-th antibody and N indicates the number of genes.

### 5.1.4. Select the Best Antibody

Whereas the merge sort has a good logarithmic complexity, we use this sort algorithm to sort antibodies according to their affinity values, and then select the best antibodies to dispatch to the cloning step.

### 5.1.5. Clone the Antibodies

In order to perform cloning step, at first, we must define a high and a low border for number of clones. We call them 'maxC' and 'minC'. Now we design an equation considering maxC, minC and affinity value of each antibody. It means the greater affinity, the greater number of clones generate. Thereupon we use equation (5) to observe above subjects:

$$N_c(i) = \text{round} \left( \frac{1}{1 + e^{-\text{fit}(i) \cdot \alpha}} \cdot \text{max} \right) + \text{min} \quad (5)$$

$\text{fit}(i)$  is value of affinity of antibody i after normalization, and  $\alpha$  is a parameter to control the number of the clones. It is obvious that if  $\text{fit}(i)$  increases,  $N_c(i)$  will go to  $\text{max} + \text{min}$ , and if  $\text{fit}(i)$  decreases,  $N_c(i)$  will depart to min. So,  $\text{maxC} = \text{max} + \text{min}$  and  $\text{minC} = \text{min}$ . Then, each antibody i clones  $N_c(i)$  times.

### 5.1.6. Mutation

Mutation procedure has an important role in the cloning procedure conducted within the AIS[30].

Many types of mutation are designed. For example: Adjacent two-job change, Arbitrary two-job change, Arbitrary three-job change, Shift change[13], Inverse, etc.

In[30], Pongcharoen et al. tested some mutation methods in GA to solve TSP problem. The results showed that when Shift change and Inverse methods are employed in sequence, best solutions will be obtained. We use these mutation methods in AIS algorithm for first time, in order to optimize our mutation procedure.

**Shift change:** In this method we select one gene or a part of the antibody, cut it, and finally, inserts it at other point of the antibody. In order to do this sequence, we shift a part of the antibody, and then insert selected part at new point. Figure 3 illustrates an example of Shift Change mutation method.

**Inverse:** in this method, we select a part of the antibody, and then reverse it in-place. Figure 4 illustrates an example of Inverse mutation method.

Because of using operation-based presentation, each change in antibody makes a new antibody that indicates a potential solution.

### 5.1.7. MCV: Maximum Conflicts-based Vaccination Method

Given an antibody, a vaccination means modifying the genes on a number of bits in harmony with priory knowledge so as to gain better(higher) affinity value[31].

In order to extract a vaccine, the algorithm reads the sequence of job execution on each of m machines, and then creates m strings from those sequences. In order to do this, when a job number has read, it will be concatenated at the end of corresponding string. Finally, each string can be used as a vaccine.

MCV process is as following:

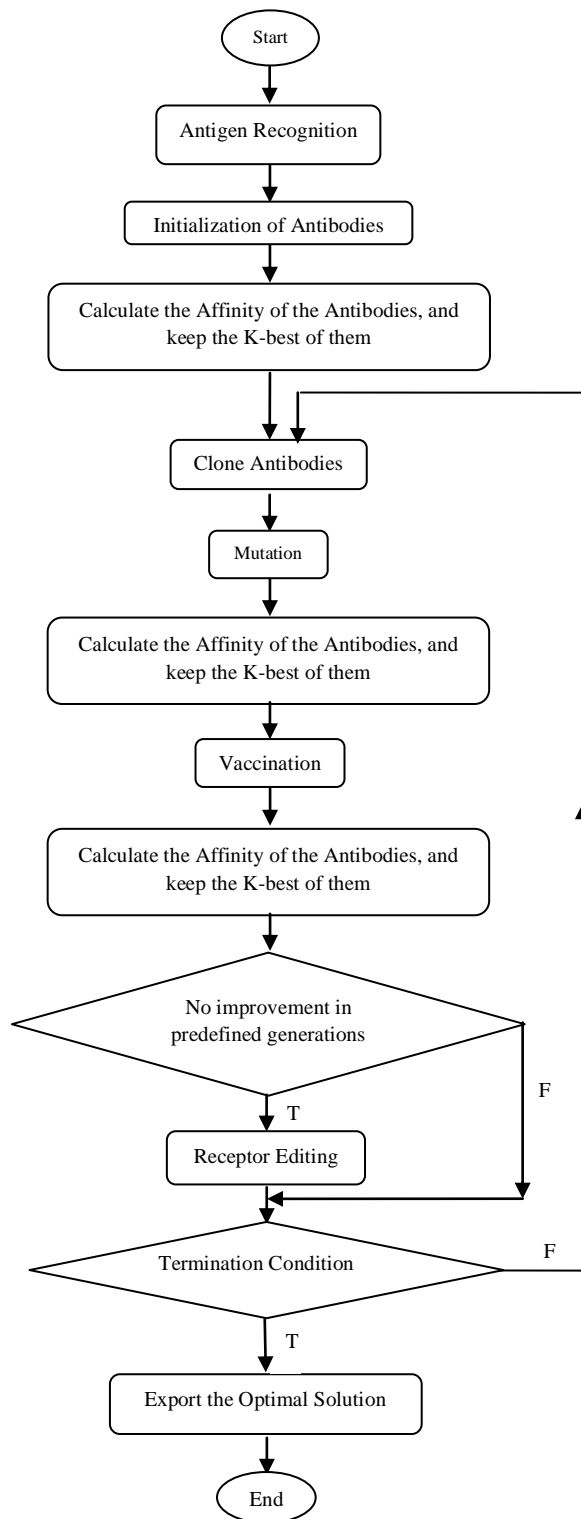
- a. Select an antibody.
- b. Select a vaccine.
- c. Start from beginning of antibody and find all conflicts with vaccine, then remove those conflicts.
- d. Add new antibody to antibody library.

For example suppose a vaccine extracted as Figure 5 for machine 3. This figure shows sequences of jobs on machine 3. Figure 6 illustrates a solution (a), machine number needed for

each operation (b), and antibody after applying vaccination(c). Also, in (c) the conflicts are highlighted. We use number of conflicts between the selected antibody and all the vaccines as

a parameter to select a vaccine. So, we use the following formula:

$$\text{score}V_i = \frac{\sum_{i=1}^N \alpha_i}{N} \quad (6)$$



**Figure 1. Outlines of proposed algorithm**

2	2	3	2	3	3
---	---	---	---	---	---

Figure 2. A simple antibody

1	1	6	4	2	1	5	6	3	2	3	6	3	6	4	1	5	4	4	5	1	2	1	3	5	4	2	5	5	3	6	3	4	6	2	2
1	1	6	4	2	1	5	6	3	2	5	4	4	5	1	2	1	3	5	3	6	3	6	4	1	4	2	5	5	3	6	3	4	6	2	2

Figure 3. Shift change mutation

Highlighted section is selected, then cut and insert it at the point specified with a thick line

1	1	6	4	2	1	5	6	3	3	6	5	4	4	5	1	4	6	3	6	3	2	4	3	1	2	1	3	5	4	2	5	5	6	2	2
1	1	6	4	2	1	5	6	3	3	6	3	4	2	3	6	3	6	4	1	5	4	4	5	1	2	1	3	5	4	2	5	5	6	2	2

Figure 4. Inverse mutation

vaccine (job sequence on machine3)	3	6	1	4	2	5
------------------------------------	---	---	---	---	---	---

Figure 5. Extracted vaccine for machine 3

1	1	6	4	2	1	5	6	3	2	3	6	3	6	4	1	5	4	4	5	1	2	1	3	5	4	2	5	5	3	6	3	4	6	2	2
(a)																																			
3	1	2	2	2	2	3	4	3	3	4	6	6	1	1	4	2	3	4	5	6	5	5	1	6	5	6	1	4	2	5	5	6	3	1	4
(b)																																			
3	1	6	4	2	1	6	6	1	4	3	6	3	6	4	1	5	2	4	5	1	2	1	3	5	4	2	5	5	3	6	3	4	5	2	2
(c)																																			

Figure 6. (a) An antibody before vaccination (b) Machine sequence for that antibody (c) Antibody after vaccination

In equation (6),  $\alpha_i$  is equal to 0 if  $V_i - Ab_{m_i} = 0$  and is equal to 1 if  $V_i - Ab_{m_i} \neq 0$ .  $V_i$  is value of  $i$ -th element of vaccine selected to inject, and  $Ab_{m_i}$  is value of the antibody element selected for vaccination, that its machine number equals  $i$ . There are two standpoints. First, we can select the vaccine which has minimum number of conflicts with the antibody, and second, we can select the vaccine which has maximum number of conflicts with the antibody. In former, the algorithm may stop in local optima and in this approach a few antibodies will be improved, because in selection of antibodies for next step, there are many repetitive antibodies and it is not desirable. Also, escape from this situation is a very difficult. In later, a big number of different antibodies will be improved concurrently, and algorithm will have many good antibodies. So, when the algorithm detects the infeasible solution, it does not need to do further work, because there are some solutions that are as well as that earlier solution. Thus, the algorithm eliminates that antibody, and attends to another

good solution. Therefore, we select a vaccine with greatest number of conflicts with the selected antibody for injection.

### 5.1.8. Receptor Editing

In the condition that no improvement occurs for the best affinity value in a certain number of generations, the receptor editing will be applied into the solution library. In this procedure, the algorithm deletes predefined number of worst antibodies from library, and then adds the same number of the best solutions into the library. Then the algorithm starts a new loop.

### 5.1.9. Termination Condition

When the predefined numbers of optimal solutions are obtained, termination condition is occurred.

## 6. EXPERIMENTAL RESULTS

In order to test our proposed algorithm, we ran the proposed algorithm on the famous benchmark FT06. Experimental results are shown in Table 1. According to this table, when Receptor Editing parameter decreases, the average time elapsed to find the optimal solution will decrease because

with smaller value of Receptor Editing parameter, best antibodies have injected into the library quickly. It means the Receptor Editing is an important parameter for achieve to the best average time passed to find the best solutions. Other parameters that have a considerable role are Min and Max. By choosing proper values for these two parameters, the average time elapsed to find the optimal solution will decrease.

**Table 1. Experimental results for different parameters (for 20 runs)**

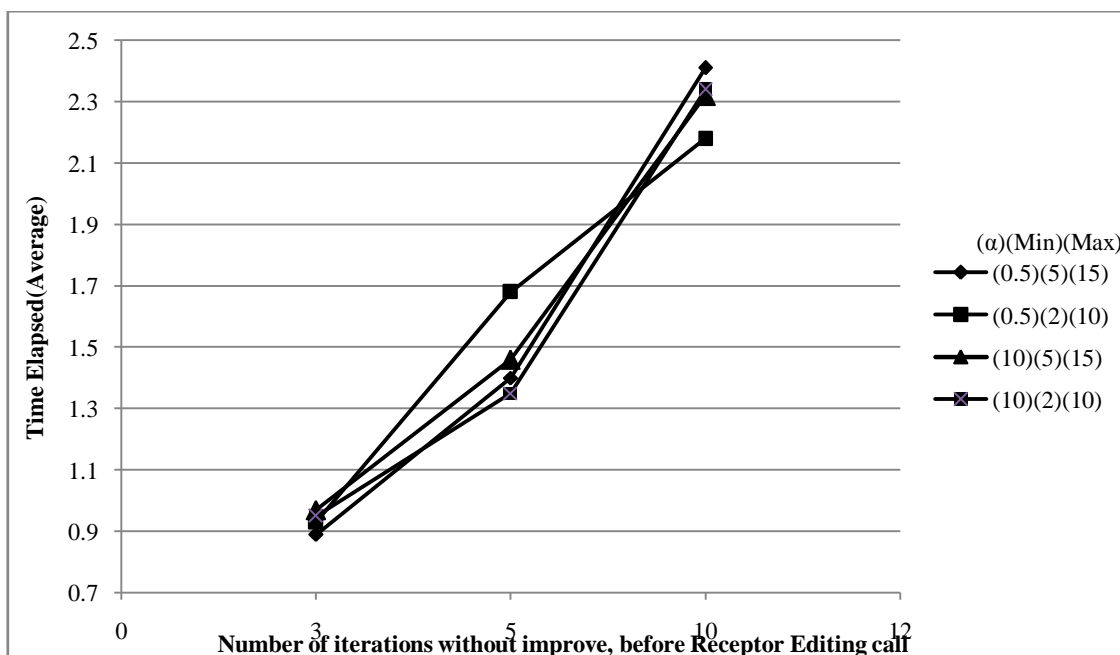
Avg-Time-One	Avg-Time-20th	Avg-Time-First	Receptor Editing	Max	Min	$\alpha$
2.41	48.14	4.63	10	15	5	0.5
1.40	27.94	4.22	5	15	5	0.5
0.89	17.87	3.94	3	15	5	0.5
2.18	43.54	4.39	10	10	2	0.5
1.68	33.54	4.47	5	10	2	0.5
0.93	18.67	4.40	3	10	2	0.5
2.32	46.44	4.97	10	15	5	10
1.46	29.24	4.66	5	15	5	10
0.97	19.39	4.64	3	15	5	10
2.34	46.77	4.78	10	10	2	10
1.35	27.07	3.58	5	10	2	10
0.95	19.09	4.26	3	10	2	10

Parameters description:

$\alpha$ : the parameter to control the number of clones. Min: the minimum number of clones. Max: the maximum number of clones. Receptor Editing: the number of repetitions no improvement occurs. In this case, our program calls Receptor Editing procedure. Avg-Time-First: the average time passed to find first optimal solution. Avg-Time-20th: the average

time passed to find 20 distinct optimal solutions. Avg-Time-One: the average time elapsed to find each of 20 optimal solutions.

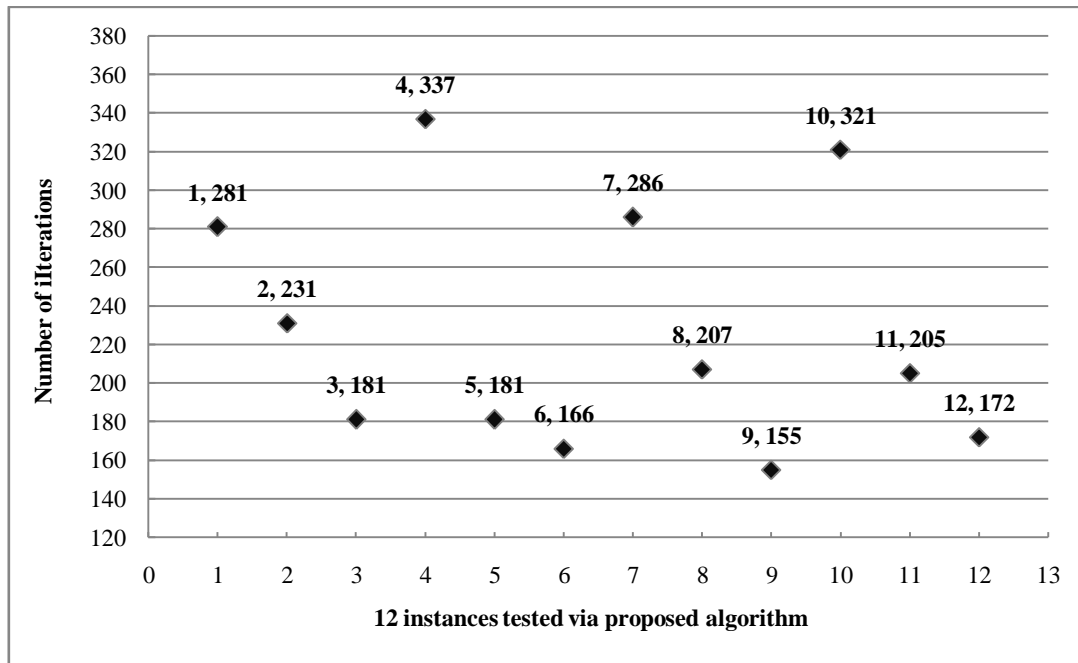
We represent a diagram as Figure 7 for ease of make a good comparison between the effects of different parameters on time passed to find the optimal solutions.



**Figure 7. The time elapsed to find each of 20 optimal solutions**

Also, in Figure 8 we demonstrate number of iterations to find 20th optimal solution. In other words, it represents the number of iterations that algorithm run in order to find 20 distinct optimal solutions. For example, for parameters according to row6 from Table 1, algorithm iterated 166 times to find 20 distinct optimal solutions.

Table 2 shows a simple comparison between our proposed algorithm and some other related algorithms on FT06. As this table shows, our proposed algorithm is better than all other algorithms. Also, our proposed algorithm has an important excellence. It can find more than one optimal solution concurrently, which this property has not been considered in other algorithms yet.



**Figure 8. Number of iterations to find 20 distinct optimal solutions.**  
 Each number in horizontal axes is corresponding to same test condition number in Table 1

**Table 2: Average time elapsed to find one optimal solution via several algorithms**

Algorithm	Optimal Solution	best Solution Found	Time Elapsed(second)
GT-Active[32]	55	55	4
GT-ND[32]		55	4
CSANN-II[33]		55	31
CSA[34]		55	12
PSMCSA[34]		55	1.28
Proposed Algorithm		55	<b>0.89</b>

## 7. CONCLUSION

We introduced a novel Artificial Immune Algorithm to solve Job-shop scheduling problem. In order to generate a good population of initial solutions, a small change is applied into G&T method. Then, to sort solutions according to their affinities, the Merge Sort algorithm is used. Also, the approach uses two mutation methods namely Shift Change and Inverse consecutively for first time in JSP to make a good improvement in potential solution. A new vaccination method based on concepts of natural vaccination mechanism is used to make further improvement in solutions. In order to escape from local optima, in special condition that no improvement occurs in solutions, Receptor Editing procedure is applied to solution library. We executed the proposed approach to solve the 6×6 FT benchmark of the JSP. The computational experiment results indicate that the proposed approach is highly practical for solving Job-shop scheduling problem and

it can obtain more than one optimal solution in admissible time complexity.

## 8. REFERENCES

- [1] Weijun, X.; Zhiming, W.; Wei, Zh. and Genke, Y., ‘A New Hybrid Optimization Algorithm for the Job-shop Scheduling Problem’, American Control Conference, Boston, 2004
- [2] Ma, J.; Zhu, Y. and Shi, g., ‘Immune Genetic Algorithm for Flexible Job-shop Scheduling Problem’, International Conference on Automation and Logistics, Hong Kong and Macau, IEEE, 2010
- [3] Hart, E.; Webb, A.; Ross, P. and Lawson, A. ‘A role for immunology in next generation robot controllers’. In Artificial Immune Systems: Proceedings of ICARIS 2003, pages 46–57, Springer, 2003

- [4] Taylor, D. and Corne, D. 'An investigation into negative selection algorithm for fault detection in refrigeration systems'. In *Artificial Immune Systems: Proceedings of ICARIS 2003*, pages 34–45. Springer, 2003
- [5] Wu, J.Y.: Artificial immune system for solving constrained global optimization problems. In: *Artificial Life 2007, ALIFE 2007*, Honolulu, HI, pp. 92–99, 2007
- [6] Neal, M. 'Meta-stable memory in an artificial immune network'. In *Artificial Immune Systems: Proceedings of ICARIS 2003*, pages 168–181, Springer, 2003
- [7] Yaqin, Zh.; Beizhi, L.; Jianguo, Y. and Qingxia, W., 'An Immune Algorithm For Batch Job-Shop Scheduling With Multi-Objectives', *International Technology and Innovation Conference*, 2006
- [8] Musilek, P.; Lau, A.; Reformat, M. and Wyard-Scott, L., 'Immune programming', *Journal on Information Sciences*, Elsevier, 2005
- [9] Tsai, J.T.; Ho, W.H.; Liu, T.K. and Chou, J.H., 'Improved immune algorithm for global numerical optimization and job-shop scheduling problems', *Apples Mathematics and Computation Conference*, pp. 406-424, Elsevier, 2007
- [10] Darmoul, S.; Pierreval, H. and Gabouj, S.H., 'Scheduling Using Artificial Immune System Metaphors: A Review', *IEEE*, 2006
- [11] Hart, E.; Timmis, J., 'Application Areas of AIS: The Past, The Present and The Future', *ICARIS2005*, pp. 483-497, Springer, 2005
- [12] Zhang, R. and Wu, Ch., 'A Hybrid Immune Simulated Annealing Algorithm for the Job Shop Scheduling Problem', *Journal of Applied Soft Computing*, pp. 79-89, Elsevier, 2010
- [13] Murata, T and Ishibuchi, H., 'Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems', *IEEE*, 1994
- [14] Bondal, Akshata A., 'Artificial Immune Systems Applied to Job Shop Scheduling', A thesis presented to the faculty of the Russ College of Engineering and Technology of Ohio University, 2008
- [15] Bernardino, H.S., Barbosa, H.J.C., 'Artificial Immune Systems for Optimization', *Nature-Inspired Algorithms for Optimisation*, pp. 389–411, Springer, 2009
- [16] Hart, E., Ross, P., 'An Immune System Approach to Scheduling in Changing Environments', *Genetic and Evolutionary Computation Conference - GECCO 1999*, Orlando, Florida, USA, pp.1559-1565, 1999
- [17] Aickelin, U.; Burke, E. and Mohamed Din, A., 'Investigating Artificial Immune Systems For Job Shop Rescheduling In Changing Environments', *Poster Proceeding of ACDM*, 2004
- [18] Coello Coello, Carlos A.; Daniel Cortés Rivas and Nareli Cruz-Cortés, 'Job Shop Scheduling using the Clonal Selection Principle', *ACDM 2004*, Springer, Bristol, U.K., 2004
- [19] Xu, X.; Wang, W. and Guan, Q., 'Adaptive Immune Algorithm for Solving Job-Shop Scheduling Problem', *ICNC 2005, LNCS 3611*, pp. 795 – 799, Springer-Verlag, 2005
- [20] Zuo, X.Q., Fan Y.Sh., 'Solving The Job Shop Scheduling Problem By An Immune Algorithm', *International Conference on Machine Learning and Cybernetics*, IEEE, 2005
- [21] Ge, H.W.; Sun, L. and Liang, Y.Ch., 'Solving Job-Shop Scheduling Problems by a Novel Artificial Immune System', *Conference on Artificial Intelligence, LNAI 3809*, pp. 839 – 842, Springer-Verlag, 2005
- [22] Luh, G.Ch. and Chueh, Ch.H., 'A Multi-Modal Immune Algorithm for the Job-Shop Scheduling Problem', *Journal of Information Science*, pp. 1516-1532, Elsevier, 2009
- [23] Hong, Lu, 'A Novel Artificial Immune Algorithm for Job Shop Scheduling', *International Conference on Computational Intelligence and Natural Computing*, IEEE, 2009
- [24] Wei, Q; Qiaoyun, L; Chuanbao, J. and Jie, S., 'Immune Genetic Algorithm and Its Application on Job Shop Scheduling', *International Workshop on Education Technology and Computer Science*, IEEE, 2010
- [25] Yagmahan, B. and Yenisey, M.M., 'Scheduling Practice and Recent Developments in Flow Shop and Job Shop Scheduling', *Computational Intelligence in Flow Shop and Job Shop Scheduling*, pp. 261-300, Springer, 2009
- [26] Bagheri, A.; Zandieh, M.; Mahdavi, I; Yazdani, M., 'An artificial immune algorithm for the flexible job-shop scheduling problem', *Journal of Future Generation Computer Systems*, pp. 533-541, Elsevier, 2010
- [27] Watson, J.P., 'Empirical Modeling and Analysis of Local Search Algorithm for the Job-Shop Scheduling Problem', Chapter2, *Ph.D. Dissertation*, 2003
- [28] Yaqin, Zh.; Beizhi, L.; Jianguo, Y. and Qingxia, W., 'Study on Modeling of Job Shop Scheduling with Multi-resource Constraints', *International Conference on Artificial Intelligence and Computational Intelligence*, pp. 313-317, IEEE, 2010
- [29] Yang, J.; Gong, M.; Jiao, L. and Zhang, L., 'Improved Clonal Selection Algorithm Based on Lamarckian Local Search Technique', *Congress on Evolutionary Computation*, IEEE, 2008
- [30] Pongchareon, P.; Chainate, W. and Pongchareon, S., 'Improving Artificial Immune System Performance: Inductive Bias and Alternative Mutations', *ICARIS 2008*, pp. 220-231, Springer, 2008
- [31] Jiao L.; Wang L., 'A Novel Genetic Algorithm Based on Immunity', *transactions on systems, man, and cybernetics-part A: systems and humans*, Vol. 30, No. 5, IEEE, 2000
- [32] Giffler, B. and Thompson, G.; 'Algorithms for solving production scheduling problems'; *operation research*, Vol. 8, pp. 487-503, 1960
- [33] Yang, Sh., 'An improved adaptive neural network for job-shop scheduling', in *proceeding of International conference of IEEE*, pp. 1200-1205, 2005
- [34] Murugesan, R. and Sivasakthi, K., 'Positive Selection Based Modified Clonal Selection Algorithm for Solving Job Shop Scheduling Problem', *Applied Mathematical Sciences*, Vol. 6, No. 46, PP. 2255 - 2271, 2012