

*Doğuş Üniversitesi Dergisi, 8 (1) 2007, 12-27*

## **A NEW APPROACH TO SOLVE FLOWSHOP SCHEDULING PROBLEMS BY ARTIFICIAL IMMUNE SYSTEMS**

### ***AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN YAPAY BAĞIŞIKLIK SİSTEMLERİ İLE ÇÖZÜMÜNDE YENİ BİR YAKLAŞIM***

**Orhan ENGİN**

*Selçuk University,  
Department of Industrial Engineering*

**Alper DÖYEN**

*Boğaziçi University,  
Department of Industrial Engineering*

**ABSTRACT:** The n-job, m-machine flow shop scheduling problem is one of the most general job scheduling problems. This study deals with the criteria of makespan minimization for the flow shop scheduling problem. Artificial Immune Systems (AIS) are new intelligent problem solving techniques that are being used in scheduling problems. AIS can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. In this research, a computational method based on clonal selection principle and affinity maturation mechanisms of the immune response is used. The operation parameters of meta-heuristics have an important role on the quality of the solution. Thus, a generic systematic procedure which bases on a multi-step experimental design approach for determining the efficient system parameters for AIS is presented. Experimental results show that, the artificial immune system algorithm is more efficient than both the classical heuristic flow shop scheduling algorithms and simulated annealing.

**Key words:** *Flow shop scheduling, artificial immune systems, clonal selection.*

**ÖZET:** n iş m makina akış tipi iş çizelgeleme problemi en genel iş çizelgeleme problemlerinden biridir. Bu çalışma akış tipi çizelgeleme problemi için toplam tamamlanma zamanı minimizasyonu ile ilgilenmektedir. Yapay Bağışıklık Sistemleri (YBS), çizelgeleme problemlerinde son dönemlerde kullanılan yeni bir problem çözme tekniğidir. YBS, doğal bağışıklık sisteminin prensiplerini ve mekanizmalarını kullanarak problemlere çözüm üreten bir hesaplama sistemidir. Bu çalışmada, bağışıklık tepkisinin iki ayrı mekanizması olan klonel seçim prensibi ve benzerlik mekanizması üzerine kurulmuş bir metod kullanılmıştır. Meta sezgisel yöntemlerde seçilen operatörler, çözüm kalitesi üzerinde önemli bir role sahiptir. Bu nedenle, yapay bağışıklık sisteminin etkin parametrelerinin belirlenmesinde çok aşamalı bir deney tasarımı prosedürü uygulanmıştır. Deney sonuçları, yapay bağışıklık sistemlerinin klasik çizelgeleme ve tavlama benzetimi algoritmalarından daha iyi sonuçlar verdiğini göstermiştir.

**Anahtar kelimeler:** *Akış tipi çizelgeleme, yapay bağışıklık sistemleri, klonel seçim.*

## 1. Introduction

Most of the research in the area of flowshop scheduling problem has concentrated on the development of a permutation flow shop schedule. It is the problem of scheduling  $n$ -jobs on  $m$ -sequential machines. The machines in a flowshop are capable of processing at most one job at a time, and each job can be processed on at most one machine at any time. The  $n$ -jobs are independent, simultaneously available at time zero, and the machine sequences of all jobs are the same. Each job has a known and finite processing time on each machine, and the processing times are independent of the order in which operations are carried out.

Preemption of individual jobs is not allowed. The objective is usually to find a sequence of  $n$ -jobs that minimizes makespan. The  $n$ -job,  $m$ -machine flowshop sequencing is a Non-Deterministically Polynomial (NP)-Hard problem. Optimal solutions can only be obtained by enumeration techniques. But these methods take a large amount of computational effort and time. Thus, heuristic methods are developed to solve these problems.

Johnson's Rule (Johnson, 1954) has been the basis of many flow shop scheduling heuristics. Palmer (1965) first proposed a heuristic for the flow shop scheduling problem to minimize makespan. The heuristic generates a slope index for jobs and sequences them in a descending order of the index. Campbell et al. (1970) proposed Campbell, Dudek, Smith (CDS) heuristic which is a generalization of Johnson's two machine algorithm; it generates a set of  $m-1$  artificial two-machine problems from an original  $m$ -machine problem, then each of the generated problems are solved using Johnson's algorithm. Gupta (1971) used the concept of Palmer's "slope index" for the heuristic that he improved. Dannenbring (1970) proposed a variation of the CDS heuristic. Nawaz et al. (1983) proposed that, a job with longer total processing time should have higher priority in the sequence. They used this approach as the main idea for their heuristic. They showed that, their heuristic (NEH) outperformed the CDS algorithm. Hundal and Rajgopal (1988) made an improvement in the Palmer's method and CDS. Ho and Chang (1991) developed a new improvement heuristic for the permutation flow shop problem.

In this paper, a new AIS approach for solving the permutation flow shop scheduling problem is proposed. The algorithm based on the mechanisms of the vertebrate immune system. The ideas proposed by De Castro and Von Zuben (2000) is taken as a basis to construct the algorithm. A multi stage experimental design for parameter optimization is presented. The proposed algorithm was tested with best found parameters on benchmark problems that were used by Carlier (1978) and the results were compared with the results of the classical heuristic algorithms and the Simulated Annealing (SA) algorithm.

In the following section, the two mechanisms of immune system: the clonal selection and affinity maturation are presented briefly. The working principles of these mechanisms have been an inspiration source for the study. In the third section, a literature survey of the applications of AIS especially in scheduling problems are presented. In the fourth section, the proposed AIS algorithm is explained in detail, moreover the parameter selection method and the experimental results of the

proposed algorithm are presented in this section. In the fifth section, a comparison of the performance of AIS with some other methods is done by using the benchmark problems of Carlier (1978). And in the last section, the paper is concluded with some comments on the findings of the study.

## **2. The Vertebrate Immune System**

All living beings have an immune system whose complexity varies according to species. Vertebrated animals have a complex and effective immune system. The immune system performs several functions, however its most remarkable roles are the protection of the organism against the attack of pathogens and elimination of mal functioning cells. The pathogens are recognized and eliminated by immune cells. There are several types of immune cells but the well-knowns are lymphocytes. These are white blood cells. There are two types of lymphocytes: B-cells and T-cells. Both cells have receptor molecules on their surfaces (the B-cell receptor molecule also called as antibody). These receptor molecules are able to recognize disease causing pathogens. When antigens and receptor molecules have complementary shapes they can bind together. The binding ensures the recognition of the antigen and the immune response starts.

### **2.1 Clonal Selection Principle**

After an antigen is recognized by immune cell receptors, the antigen stimulates the B-cell to proliferate (divide) and mature into terminal (non-dividing) antibody secreting cells (plasma cells) (De Castro and Von Zuben, 2000). The proliferation in the immune system is succeeded by cell divisions (mitosis). After the proliferation, the system has a clone of cells that are copies of each other. The proliferation rate of a cell is directly proportional to its recognizing degree of the antigen. Also there is a selective mechanism: The offspring cells which better recognize the antigen are selected and differentiated into long-lived memory cells. Memory cells circulate through the blood, lymph and tissues, and when exposed to a second antigenic stimulus they differentiate into large lymphocytes capable of producing high affinity antibodies, pre-selected for the specific antigen that had stimulated the primary response (De Castro and Von Zuben, 1999).

The immune cell learns by raising the population size and affinity (the degree of the cell recognition with the antigen) of the cells. There is another way of learning: the effectiveness of the immune response to secondary encounters could be considerably enhanced by storing some high affinity antibody producing cells from the first infection (memory cells), so as to form a large initial clone for subsequent encounters (Ada and Nossal, 1987). By this method, the response becomes more faster and efficient. De Castro and Von Zuben (2000) showed the whole clonal selection principle as in Figure 1.

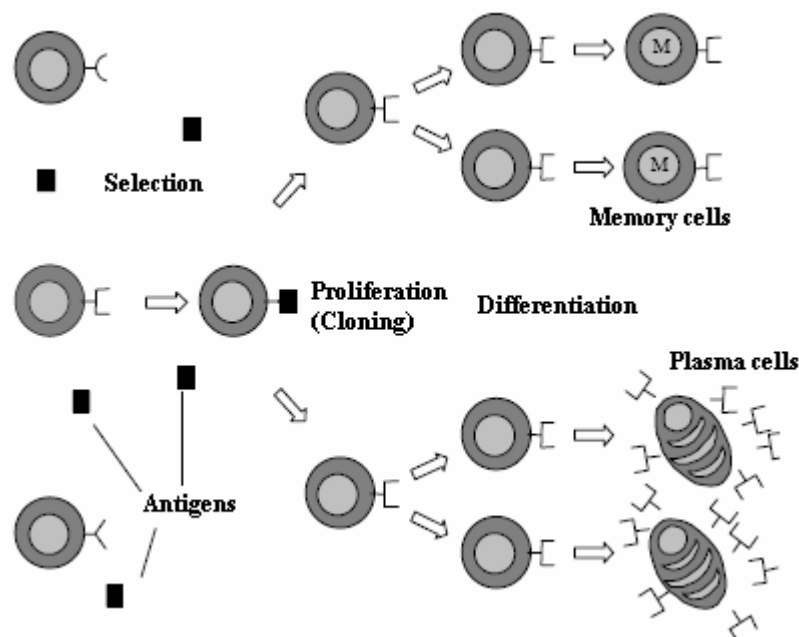


Figure 1. The Clonal Selection Principle (De Castro and Von Zuben, 2000).

## 2.2 Affinity Maturation

Affinity maturation is the whole mutation process and the selection of the variant offspring that better recognizes the antigen (De Castro and Timmis, 2003). The two basic mechanisms of affinity maturation are those: hypermutation and receptor editing (De Castro and Von Zuben, 2002).

Random changes (mutations) take place in the variable region genes of antibody molecules. That random changes are mutational events and cause structurally different cells. Occasionally one such change will lead to an increase for the affinity of the antibody. The mutation processes on lymphocytes are named as somatic hypermutation. Somatic hypermutation rate is inversely proportional to the cell-affinity: the higher the affinity a cell receptor has with an antigen the lower the mutation rate and vice-versa. With this strategy, the immune system keeps in hand the high affinity offspring cells and also ensures large mutations for the low affinity ones to get better affinity cells. Due to the random nature of the somatic mutation processes, a large proportion of mutating genes become non-functional or develop harmful anti-self cells. Those cells are eliminated by a programmed death process. But all cells with low affinities and anti-self specifications are not deleted, there is a process known as receptor editing: B-cells delete their self reactive receptors and develop entirely new receptors (De Castro and Von Zuben, 2000). This process of receptor editing may cause a receptor with a better or worse affinity. Point mutations are good for exploring local regions, while editing may rescue immune responses stuck on unsatisfactory local optima (De Castro and Von Zuben, 1999).

### 3. Artificial Immune Systems (AIS)

The vertebrate immune system is one of the most complex systems of the body. The progression of science ensures to understand its working principles more easily in recent years. The operative mechanisms of immune system are very efficient from a computational standpoint. Similarly to the artificial neural networks which were inspired from nervous system, AIS were developed as a novel computational intelligence approach. AIS are defined as computational systems inspired by theoretical immunology and observed immune functions, principles, and models, applied to solve problems (De Castro and Timmis, 2002a). The main application domains of AIS are optimization, pattern recognition, computer and network security, scheduling, anomaly detection and data mining. Although there is a wide range of application areas of AIS, still there is not so many studies related with it. But the number of interested people has been increasing in the recent years. Forrest et al. (1994) used the r-contiguous bit rule and compared the problem of protecting computer systems to that of learning to distinguish between self and nonself, and proposed the negative selection algorithm. Dasgupta and Forrest (1996) proposed to apply the negative selection algorithm of AIS to detect novelties in time series data. Dasgupta and Forrest (1999) proposed an AIS algorithm for tool breakage detection. Method is inspired by negative selection algorithm that enables to distinguish self and nonself cells. In the study, self is defined to be normal cutting operation and the nonself is any deviation beyond allowable variation of the cutting force. De Castro and Timmis (2002b) presented the use of AIS in pattern recognition applications. Taranakov and Dasgupta (2000) developed a mathematical model of lymphocyte cell interactions between each other. Forrest and Hofmeyr (2001) used an AIS approach to protect a network of computers from illegal intrusions due to the properties of being a distributed, robust, dynamic, diverse, and adaptive system. Nasaroui et al. (2002) used an AIS for web mining. AIS's ease of adaptation to the changing/dynamic environment that characterizes the world wide web was a strong advantage for the application. De Castro and Von Zuben (2000) applied the clonal selection algorithm to solve multi-modal optimisation, pattern recognition tasks and the travelling salesman problem. Timmis and Neal (2000) presented an AIS for data analysis. Trojanowski and Wierzchon (2002) used AIS for non-stationary function optimization.

The efficient mechanisms of immune system which are the clonal selection, learning ability, memory, robustness and flexibility make artificial immune systems useful for scheduling problems. Mori et al. (1997) proposed an AIS to control a semiconductor production line. The control of the production line was done by a set of agents. Each agent interacted with the production line and with other agents. This system was a copy of the immune system. Hart et al. (1998) used an AIS approach to solve job shop problems. They used problems where each job has associated release and due dates. Their goal was finding schedules that minimizes maximum tardiness. Each solution (a complete schedule) was an antibody. They build a number of libraries. Each library contains a number of genetic strings, each string being a part of solutions to a set of job shop problems. By concatenating strings from each library, an antibody (schedule) is constructed. 1000 clones of the best individual found were generated. The clones were mutated and the best clone found was selected as the solution of the problem. Russ et al. (1999) give an AIS model for task allocation in computer systems with the goal of designing a system that is able to adapt changing conditions. They used a set of agents, agents are interacted

with the system and themselves. Costa et al. (2002) proposed an immune based approach to minimize makespan on parallel processors. They defined an affinity function based on makespan values of the schedules. Also they give a function to calculate the number of clones that will be proliferated. They compared the performance of their algorithm with heuristics: Longest Processing Time, Multifit, Local Search and Simulated Annealing.

#### 4. Experimentation

In this section, Simulated Annealing (SA) algorithm and the proposed AIS approach will be described. These algorithms are implemented for solving flow shop scheduling problems when the objective is makespan minimization.

##### 4.1. Simulated Annealing (SA) Algorithm

SA is a nature-based stochastic algorithm which produces good suboptimal solution (Tian et al.,1999). The SA is an iterative search procedure based on a neighborhood structure. Recently, much research has been done on the application of SA to flow shop scheduling problem (Osman and Potts,1989; Ogbu and Smith, 1990). The SA algorithm can be described as follows (Tian et al.,1999).

*Step 1.*

Find an initial solution  $i \in S$ ;

Set simulation temperature values  $T_0 > T_i > 0$ ; Set an iterative counter  $k=0$ ;

*Step 2.*

Generate a random solution  $j \in N(i)$ ; Find  $\Delta f = f(j) - f(i)$ .

*Step 3.*

If Metropolis criterion is satisfied, i.e.,  $\min \{1, \exp(-\Delta f / T_k)\} > \eta \in [0,1)\}$ , then  $i=j$ ;

*Step 4.*

If Metropolis equilibrium under  $T_k$  is realised, then go to Step 5; Else go to Step 2.

*Step 5.*

If stop criterion is not satisfied, i.e.,  $T_k > T_s$ , then reduce the temperature

$T_{k+1} = T_k - \Delta T_k$ ,  $\Delta T_k > 0$  and set  $k = k+1$ , go to Step 2; Else output  $i_{opt} = i$ .

*End*

##### 4.2. Proposed AIS Approach

###### 4.2.1. Algorithm

In this study, possible schedules are represented by integer-valued strings of length  $n$ . The  $n$  elements of the strings are the jobs which will be sequenced, so the strings are composed of permutations of  $n$  (*jobs*) elements. Those strings are accepted as antibodies of the AIS. The algorithm goes to solution by the evolution of these antibodies. Evolution based on two basic principles of the vertebrate immune system: Clonal Selection and Affinity Maturation.

The proposed algorithm is presented below:

```

Create a population of  $B$  antibodies ( $B$  is the size of antibody population);
 $x=0$ ;
For each generation do;
 $x=x+1$ ;
  For each antibody do:
    Decode the antibody;
    Determine the makespan (affinity) of antibody;
    Calculate the selection probability (rate of cloning);
    Cloning (generate copies of antibodies);

    For each generated clone do;
      inverse mutation (generate a new string);
      decode the new string;
      calculate the makespan of the new string;
      if makespan (new string) < makespan (clone) then clone=new string;
      else do pairwise interchange mutation (generate a new string);
      decode the new string;
      calculate the makespan of the new string;
      if makespan(new string)< makespan(clone) then clone=new string;
      else clone=clone;
    antibody=clone;

  if  $x = A$  (frequency of elimination steps) then
    eliminate worst  $C$  number of antibodies in the population ( $C$  is elimination
      ratio of antibodies);
    create  $C$  new random antibodies;
    change the new created ones with the eliminated ones;
   $x=0$ ;
end if;
while stopping criteria=false (stopping criteria is the number of generations
  which is user defined)

```

#### Cloning selection processes of the algorithm

Each schedule (antibody) has a makespan value that refers to the affinity value of that antibody. Affinity value of each schedule is calculated due to the affinity function. Equation 1 shows the affinity function, where  $z$  represents the antibody.

$$Affinity(z) = \frac{1}{makespan(z)} \quad (1)$$

From the equation, it can be noticed that the lower the makespan value the higher the affinity value. In the algorithm the cloning of antibodies is done directly proportional to their affinity function values, which is also the case in the vertebrate immune system.

We propose a cloning procedure for the algorithm. The procedure is a special version of the roulette wheel method (Goldberg, 1989). Here makespan values of schedules are used rather than an objective function value. Opposite to maximization

problems, the procedure gives more chance to the ones with lower makespan ( $C_{max}$ ) for selecting and cloning. Thus, there will be more clones of antibodies with lower  $C_{max}$  values than high  $C_{max}$  valued antibodies in the new generated clone population.

The selection probability of each antibody is calculated due to the following procedure:

- a) For each antibody in the population calculate the makespan value,
- b) Find the maximum makespan value ( $Max C_{max}$ )
- c) For each antibody calculate the fitness value due to Equation 2.

$$Fitness\ value = (MaxC_{max} + 1) - (makespan\ of\ antibody) \quad (2)$$

- d) For each antibody find the selection probability by using Equation 3.

$$selection\ probability = \frac{fitness\ value\ of\ antibody}{total\ of\ fitness\ values\ of\ antibodies\ in\ the\ population} \quad (3)$$

In the algorithm a fixed size of antibody population was used, also we generated the set of clones with the same size of antibody population. The number of clones which are generated from each antibody, changes due to the selection probability of the antibody. It is expected that the antibodies with greater selection probabilities will have more clone (copy) in the clone set. Because of the fixed size of the clone set, some of the antibodies with high  $C_{max}$  values may have no clones in the clone set, while the antibodies with lower  $C_{max}$  values may have lots of clones.

#### **Affinity maturation processes of the algorithm:**

**Mutations:** In the study, a two phased mutation procedure was used. The generated clones undergo an inverse mutation procedure firstly.

*Inverse Mutation:* Given a sequence  $s$ , let  $i$  and  $j$  be two positions in the sequence  $s$ . A neighbor of  $s$  is obtained by inverting the sequence of jobs between positions  $i$  and  $j$ . If the makespan value of mutated sequence (after inverse mutation) is smaller than that of the original sequence (a generated clone from an antibody) then the mutated one is stored in place of the original one. Otherwise, the sequence will be mutated again with random pairwise interchange mutation method.

*Pairwise Interchange Mutation:* Given a sequence  $s$ , let  $i$  and  $j$  be randomly selected two positions in the sequence  $s$ . A neighbor of  $s$  is obtained by interchanging the jobs in positions  $i$  and  $j$ . If the makespan value of mutated sequence (after pairwise interchange mutation) is smaller than the original sequence, then store the mutated one in place of the original one.

In the case of the algorithm could not find a better sequence after the two mutation procedure then store the original sequence (generated clone). In the inverse mutation phase, the considered sequence is not allowed to be mutated if  $|j - i| < 2$ .

This constraint maintains a higher mutation rate for inverse mutation than the pairwise interchange mutation because the constraint allows to change the places of jobs of more than two. In the early steps of our algorithm, it is a much possibility to find a better sequence by employing the inverse mutation because the algorithm still



far away from the good solutions and the large mutations may cause to find better job sequences. In later steps the algorithm will have good solutions. The possibility of finding better sequences by the use of making large mutations is low, because large mutations may cause to lose good partial job sequences and escape from optimal. So, in the later steps it is more efficient to make relatively small mutations. In the proposed algorithm this efficiency is secured by using the pairwise interchange mutation method when the inverse mutation does not give a better solution.

As described above, by the time the algorithm ensures to make relatively small mutations contrary to the relatively increase (decrease in makespan values) in affinity function values. It was explained in Section 2.2 that, in the vertebrate immune system the mutations on receptor molecules are done inversely proportional to the affinity degree of that receptor molecule. The approach in the algorithm fits well with this mechanism.

**Receptor Editing:** For some steps of the algorithm, a number of worst antibodies in the antibody population are eliminated and randomly created antibodies at the same number are replaced with them. This mechanism is also a vertebrate immune system mechanism, named receptor editing and described in Section 2.2. This mechanism allows to find new schedules, what means of new search regions in the total search space. Exploring new search regions may help the algorithm to escape from local optimals. This process is applied in every  $A$  ( $A$  is a user defined parameter) generations in the algorithm. A counter, named  $x$ , is used to count the generation number and when  $x=A$  algorithm starts the receptor editing procedure.

In summary; all the clones in the set of clones which are the copies of antibodies with good affinity degrees undergo a mutation process. This is a two phased mutation process: firstly the inverse mutation is applied. In the case of not having a better solution, then the pairwise interchange mutation method is applied. Also after the second mutation method if there is no improvement then the original schedule (copy of an antibody in the copies set) remains with no change. And in some steps of the algorithm a model of receptor editing mechanism of the immune system was used: a proportion of the worst schedules eliminated and new ones are generated in place of them.

The clone set is accepted as an antibody population set for the next generation after these cloning and mutation processes. Thus, the clones which had the mutation process is assigned as antibodies for the next generation. In the next generation the clones will be copied from these antibodies. In the algorithm this statement was given as *antibody=clone*:

#### 4.2.2. Parameter optimization for AIS

For obtaining optimal or near-optimal solutions of any combinatorial optimization problem in a shorter time, one should use the optimal set of the parameters. In this study the parameters of the AIS algorithm were improved by Multi Step Experimental Design Approach (MSEDA) (Fıglalı et al., 2002). The experimental design uses an Orthogonal Array (OA) which prescribes series of trials to perform. In our algorithm there are three parameters which effect the solution. Two levels of each factor are used. The three variable factors are frequency of elimination steps (A), size of antibody population (B) and elimination ratio of antibodies (C)

respectively. In Table 1 the two level L8 design for experiments is presented. In this study, all possible combinations of factor A,B, and C factor at each of the levels 1 and 2 are tested as a full factorial experiment. The lower and upper bounds of parameter ranges correspond to the factor levels. The ranges of each parameter are given in Table 2.

**Table 1. L8 OA for Experiments**

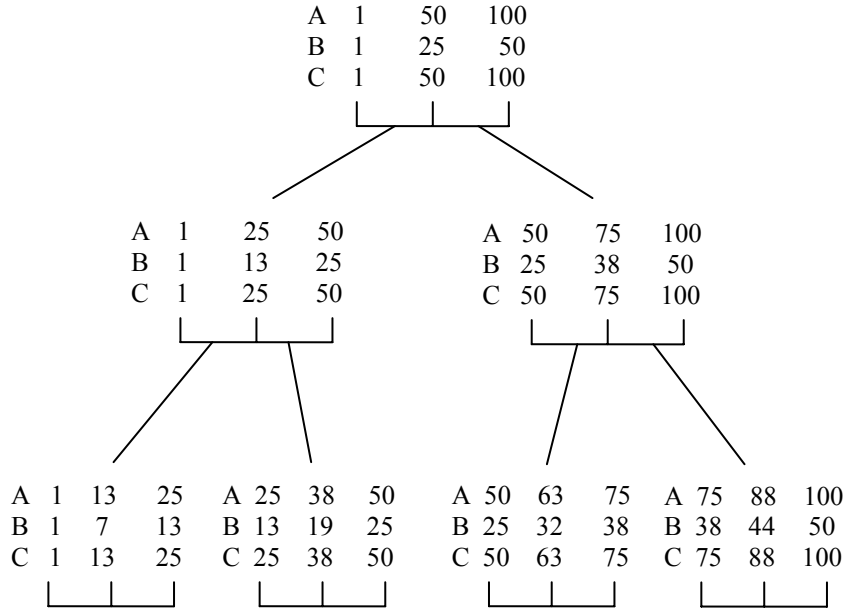
Trial no	A	B	C
1	1	1	1
2	1	1	2
3	1	2	1
4	1	2	2
5	2	1	1
6	2	1	2
7	2	2	1
8	2	2	2

**Table 2. The parameter ranges of AIS for Flowshop scheduling Problems**

Parameter	Factor Levels
Frequency of Elimination Steps (A)	1-100
Size of Antibody Population (B)	1-50
Elimination Ratio of Antibodies (C)	1-100 (%)

The used MSEDA (Fırlalı et al., 2002) for determining the optimal parameter set is explained as follows.

- Step 1:* Quadripartite the whole range for each parameter and, take the end points of the first and the third quadriparts as the levels of first step of the experimental design.
- Step 2:* Solve the problem 25 times by using L8 orthogonal design-parameter set determined in Step 1.
- Step 3:* Select the best parameter set depending on the mean values of makespans for 25 solutions from 8 experiments.
- Step 4:* For each parameter in the best parameter set apply the Search Range Limitation Procedure (SRLP) for determining the new parameter set as shown in Figure 2.
- Step 5:* Calculate the individual and interaction effects of the parameters on the solution.
- a)* If there is no interaction effect, then solve the problem 25 times by using L8 orthogonal design with each parameter set determined in Step 4.
- b)* If there is interaction between the parameters, then select the parameter due to the interaction.
- Step 6:* Compare the obtained Average of Mean Values ( $AMEV_n$ ) with the former step's  $AMEV_{n-1}$ .  
If  $AMEV_{n-1} < AMEV_n$  Stop the algorithm.  
If  $AMEV_{n-1} \geq AMEV_n$  Go to Step 7.
- Step 7:* Select the best parameter set depending on the mean values of makespans for 25 solutions from the experiments. Go to Step 4.



**Figure 2. Search Range Limitation Procedure (SRLP)**

Notice that, SRLP limits the half of the search range in each step for each parameter. The half range which contains the effective level for any parameter is limited to a new search. The effective half range is quadripartited for each parameter and the end points of the first and the third quadriparts are taken as the levels of the new experiment.

The proposed AIS algorithm was coded in Visual Basic. Algorithms were run on a Pentium 4 1.7 GHz IBM PC. The generation number for the proposed AIS algorithm was selected as 150. For the experiments, a set of problems proposed by Carlier (1978) were used. The sizes of the  $n$  (jobs)  $\times$   $m$  (machines) type problems as follows: 7x7, 13x4, 11x5, 14x4, 12x5, 10x6, 8x8, 8x9. For each of the problems the MSEDAs was applied to obtain efficient parameter sets. For each of the problems 8 trials were implemented due to the L8 OA. Each trials were run 25 times for each step, and totally 33 steps (5 steps for 7x7 problem, 5 steps for 13x4 problem, 2 steps for 11x5 problem, 4 steps for 14x4 problem, 5 steps for 12x5 problem, 3 steps for 10x6 problem, 5 steps for 8x8 problem, 4 steps for 8x9 problem) were implemented among the eight problems. In the whole experiment 6600 runs were made among the 8 problems.

Table 3 shows the implementation results of the MSEDAs algorithm for 7x7 problem. Table 3 shows the mean and minimum values of makespan values for 25 runs in each of the trials. For each step the minimum mean value and the average of mean values of the eight trials (AMEV) are shown in bold style. In each step, the trial with minimum mean value is selected and SRLP procedure is applied to its parameter set. SRLP limits the half of the search range in each step for each parameter. The half range which contains the effective level for any parameter is limited to a new search. The effective half range is quadripartited for each parameter

and the end points of the first and the third quadriparts are taken as the levels of the new experiment. This procedure continues until AMEV of the following step becomes higher than the prior step's AMEV. The parameter set of the trial with the lowest mean value in the last decreasing step, is accepted as the efficient parameter set.

**Table 3. Implementation of the MSED algorithm for 7x7 Problem**

Step	Trial No.								AMEV	
	1	2	3	4	5	6	7	8		
1	mean	6668	6763	<b>6600</b>	6664	6724	6651	6610	6605	<b>6660</b>
	min.	6590	6590	6590	6590	6590	6590	6590	6590	6590
2	mean	<b>6608</b>	6656	6625	6627	6652	6625	6620	6633	<b>6631</b>
	min.	6590	6590	6590	6590	6590	6590	6590	6590	6590
3	mean	6629	6629	6614	6604	6632	6646	6614	<b>6603</b>	<b>6621</b>
	min.	6590	6590	6590	6590	6590	6590	6590	6590	6590
4	mean	6624	6622	6619	6620	6619	<b>6601</b>	6619	6624	<b>6619</b>
	min.	6590	6590	6590	6590	6590	6590	6590	6590	6590
5	mean	6633	6642	6649	6619	6639	6687	6649	<b>6603</b>	<b>6640</b>
	min.	6590	6590	6590	6590	6590	6590	6590	6590	6590

The procedure for 7x7 problem becomes as following. In the beginning (first step), the parameter A ranges from 1-100; parameter B ranges from 1-50, parameter C ranges from 1-100. After the whole ranges for each parameter are quadripartited, the end points of the first and third quadriparts are taken as the first and second level of the experimental design ( for parameter A the levels are; 25 and 75, for B; 13 and 38 and for C; 25 and 75). Eight trials are prepared due to L8 OA. The third trial gives the minimum mean that corresponds to 1-2-1 (25-38-25) parameter levels according to L8 OA. Then SRLP procedure is applied to that parameter set and new parameter levels for the following step are obtained. For parameter A, the effective level was chosen as 25. The search range is lowered to 1-50, as 25 is the mid point of the range. This new range is quadripartited and new levels are determined (which is 13 and 38). Similarly, for the parameters B and C also the new levels are determined for the second step experimental design levels. This process continues until there is an increase in the average of mean values of the eight trials (AMEV) from the previous step's AMEV. The fifth step AMEV is higher than AMEV of the fourth step so process is stopped here. The parameter set of the trial with the lowest mean value in the fourth step (this is the sixth trial, as seen in Table 3) is taken as the efficient parameter set for the 7x7 problem. In Table 4, the best parameter set found in each step is seen in bold letters. The average of mean values of the eight trials decrease until the fifth step, so the best found parameter set in the fourth step is accepted as the efficient parameter set. The trial with the minimum mean in the fourth step is the sixth trial, with mean makespan value of 6601 and parameter set of 22-34-22. This parameter set is accepted as the efficient parameter set for 7x7 problem.

**Table 4. The Best Parameter Sets of Each Phase of MSED A for 7x7 Problem**

Steps	Minimum of trial mean values	Average of mean values of trials	First and second levels of A	First and second levels of B	First and second levels of C
1	6600	6660	25 75	13 38	25 75
2	6608	6631	13 38	32 44	13 38
3	6603	6621	7 19	29 35	7 19
4	6601	6619	16 22	34 37	16 22
5	6603	6640	21 24	33 35	21 24

The parameter interactions are important. If there is a positive interaction between parameters the same level of parameters must be chosen for the following step in SRLP procedure. In our study we looked for the interactions between each parameter for each problem. We could not find a meaningful interaction between parameters. In Table 5, for the 4x13 problem the interaction values of each parameter pairs are shown. The F-test value for at least 95 % confidence is calculated as 3.940158. As seen in the table there is not any value greater than 3.940158, so it can be said that there is no interaction between parameters.

**Table 5. The interactions between each parameter pairs for 4x13 problem.**

Step	A-B	A-C	B-C
1	2.84	0.08	3.12
2	1.73	0.421	0.08
3	0.16	1.6	3.27
4	0.005	0.89	0.39
5	0.62	2.19	0.0006

Also we examined the individual effects of the parameters, and found that the parameter of antibody population size has the greatest effect overall.

Use of this MSED A approach well-performed for the solutions. Table 6 presents the improvement in AMEVs. For 7x7 problem the AMEV in the beginning step was 6660, after applying MSED A the AMEV was decreased to 6619.

**Table 6. Improvement with the MSED A**

Problem	Beginning AMEV	Lasting AMEV
7x7	6660	6619
13x4	7359	7265
14x4	8149	8088
12x5	7516	7445
8x9	8706	8612
11x5	7101	7038
10x6	7878	7801
8x8	8520	8444

## 5. Comparative Results

The eight problems of Carlier (1978) were solved with classical heuristics (Engin and Fıđlalı, 2001), SA and the proposed AIS algorithm. The proposed AIS algorithm is run with the best parameters, which are found by the parameter optimization method (MSEDA).

In Table 7, the best makespan values for each problem were shown in bold style, and the second best values were shown in italic style. It can be inferred from the table that, AIS algorithm outperformed all other considered methods. The SA gives more better solutions than classical heuristic methods. Among the classical heuristics the NEH algorithm performs better than the others. From the table it is also seen that, the standart deviation between the values of different methods is very high. That means, the method used for solving flow shop scheduling problems has important effect on solution quality.

**Table 7. Best found makespan values of the test problems**

Methods	Problems							
	11x5	13x4	12x5	14x4	10x6	8x9	7x7	8x8
<b>AIS</b>	<b>7038</b>	<b>7166</b>	<b>7312</b>	<b>8003</b>	<b>7720</b>	<b>8505</b>	<b>6590</b>	<b>8366</b>
SA	<i>7038</i>	<i>7284</i>	<i>7432</i>	<i>8092</i>	<i>7785</i>	<i>8692</i>	<i>6637</i>	<i>8431</i>
NEH	<i>7038</i>	<i>7376</i>	<i>7399</i>	<i>8003</i>	<i>7838</i>	<i>8773</i>	<i>6590</i>	<i>8564</i>
CDS	<i>7202</i>	<i>7410</i>	<i>7399</i>	<i>8423</i>	<i>8627</i>	<i>9082</i>	<i>6760</i>	<i>8903</i>
Dannenbring	<i>7817</i>	<i>7509</i>	<i>7399</i>	<i>8357</i>	<i>8933</i>	<i>9179</i>	<i>6760</i>	<i>9062</i>
Gupta	<i>7274</i>	<i>7534</i>	<i>7399</i>	<i>8423</i>	<i>8773</i>	<i>9441</i>	<i>7043</i>	<i>9150</i>
Ho-Chang	<i>7202</i>	<i>7410</i>	<i>7399</i>	<i>8423</i>	<i>8627</i>	<i>9082</i>	<i>6760</i>	<i>8903</i>
Hundal-Rajgopal	<i>7175</i>	<i>7376</i>	<i>7779</i>	<i>8423</i>	<i>8520</i>	<i>9487</i>	<i>7043</i>	<i>8825</i>
Johnson	<i>7689</i>	<i>7741</i>	<i>7779</i>	<i>8679</i>	<i>8627</i>	<i>10211</i>	<i>6887</i>	<i>9696</i>
Palmer	<i>7472</i>	<i>7940</i>	<i>7779</i>	<i>8423</i>	<i>8520</i>	<i>9487</i>	<i>7043</i>	<i>8825</i>
<b>Standart deviation</b>	<b>276</b>	<b>224</b>	<b>190</b>	<b>220</b>	<b>442</b>	<b>494</b>	<b>183</b>	<b>386</b>

## 6. Conclusions

In this paper, a new AIS approach for solving permutation flow shop scheduling problems when the objective is makespan minimization was proposed. The algorithm uses simple but effective techniques for calculating cloning process, applying mutations and applying a receptor editing procedure. The procedures used in cloning and mutation phases of the algorithm has not been considered before any other AIS resarchers. A new parameter optimization approach (MSEDA) was used to obtain better parameters. The AIS algorithm was tested with the best found parameters. The AIS algorithm, SA algorithm and other heuristics were tested on the benchmark problems that were given by Carlier (1978) and a comparison was made. It was seen that the proposed AIS approach was more efficient than other heuristics.

## References

- ADA, G.L., NOSSAL, G. (1987). The clonal selection theory. *Scientific American*, vol.2, pp.50-57.
- CAMPBELL, H.G., DUDEK, R.A., SMITH, M.L (1970). A heuristic algorithm for the n-job, m-machine sequencing problem. *Management Science*, 16/B, pp. 630-637.

- CARRIER, J. (1978). Ordonnancements a contraintes disjonctives. *R.A.I.R.O. Operational Research* 12, pp. 333-351.
- COSTA, A.M., VARGAS, P.A., VON ZUBEN, F.J.AND FRANÇA, P.M. (2002). IEEE World Congress on Computational Intelligence, In the proc. of the special sessions on artificial immune systems in the 2002 Congress on Evolutionary Computation, *Makespan minimization on parallel processors: An immune based approach*, Honolulu, Hawaii.
- DANNENBRING, D.G. (1970) An evaluation of flow shop sequencing heuristics, *Management Science*, vol. 23, pp. 1174-1182.
- DASGUPTA, D., FORREST, S. (1996). Proceedings of the ISCA'96, *Novelty detection in time series data using ideas from immunology*.
- DASGUPTA, D., FORREST, S. (1999). Proceedings of the Second International Conference on Intelligent Processing and Manufacturing Materials (IPMM) *Artificial immune systems in industrial applications*, Honolulu, July 10-15.
- DE CASTRO, L.N., VON ZUBEN, F.J. (1999). *Artificial immune systems: Part 1- Basic theory and applications*. Technical Report, TR-DCA 01/99.
- . (2000). GECCO 2000 – Workshop proceedings *The clonal selection algorithm with engineering applications*. pp.36-37. [Available form: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/gecco00.pdf>]
- . (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, vol.6, no.3, pp.225.
- DE CASTRO, L.N., TIMMIS, J.I. (2002a). Artificial immune systems: A novel paradigm for pattern recognition, In L. ALONSO J. CORCHADO, and C. FYFE (eds). *Artificial Neural Networks in Pattern Recognition*, pages 67-84, University of Paisley.
- . (2002b). *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag.
- . (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing Journal*, vol.7, Issue 7.
- ENGİN, O., FIĞLALI, A. (2001). Performance analysis of classical heuristic methods and artificial intelligence techniques used in flow shop scheduling problems: A comparative approach. *Journal of Engineering and Architecture Faculty of Selcuk University*, v.16, n.2, pp.7-17.
- FIĞLALI, A., ENGİN, O., FIĞLALI, N. (2002). International Conference on Fuzzy Systems, Soft Computational Intelligence in Management and Industrial Engineering *A systematic procedure for setting ant system parameters*, May 29-31, İstanbul, Turkey.
- FORREST, S., PERELSON, A., ALLEN, L., CHERUKURI, R. (1994). Proceedings of the IEEE Symposium on Research in Security and Privacy. *Self-nonsel self discrimination in a computer*, pp. 202-212.
- FORREST, S., HOFMEYR, S.A. (2001). Engineering an immune system. *Graft*, vol. 4:5, pp. 5-9.
- GOLDBERG, D.E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Company, USA.
- GUPTA, J.N.D. (1971). A functional heuristic algorithm for the n-job, m-machine flow shop proble., *Operational Research Quartely* , vol.22 , pp. 39-47.
- HART, E., ROSS, P.AND NELSON, J. (1998). Proc. of ICEC'98. *Producing robust schedules via an artificial immune system*, pp.464-469.
- HO, J.C, CHAN, Y.L. (1991). A new heuristic for the n-job m-machine flow shop problem, *European Journal of Operational Research*, 52, pp.194-202.

- HUNDAL, T.S., RAJGOPAL, J. (1988). An extension of Palmer's heuristic for the flow-shop scheduling problem. *International Journal of Production Research*, 26, pp. 1119-1124.
- JOHNSON, S.M. (1954). Optimal two and three stage production schedules with set up times included. *Naval Research Logistics Quarterly*, vol. 1, pp. 61-68.
- MORI, M., TSUKIYAMA, M., FUKUDA, T. (1997). Proc. of the IEEE Systems, Man and Cybernetics Conference. *Artificial immunity based management system for a semiconductor production line*, pp. 851-855.
- NASAROU, O., DASGUPTA, D., GONZALES, F. (2002). Workshop on Web Analytics at Second SIAM International Conference on Data Mining (SDM) *The promise and challenges of artificial immune system based web-usage mining: preliminary results*. Arlington VA, April 11-13.
- NAWAZ, M., ENSCORE, E.E., HAM, I. (1983). A heuristic algorithm for the machine, n-job flow shop sequencing problem. *Omega*, vol.11, pp .91-95.
- OGBU, F.A., SMITH, D.K. (1990). The application of the simulated annealing algorithm to the solution of the n/m/Cmax flow shop problem. *Computers, Operations Research*, 17, pp.243-253.
- OSMAN, I.H., POTTS, C.N. (1989). Simulated annealing for permutation flow shop scheduling. *Omega*, 17, pp,551-557.
- PALMER, D.S. (1965). Sequencing jobs through a multi stage process in the minimum total time - a quick method of obtaining a near optimum. *Operational Research Quarterly*, 16, pp., 101-107.
- RUSS, S.H., LAMBERT, A., KING, R., RAJAN, R., REESE, D. (1999). Proc. of the Symposium on High Performance Distributed Computing. *An artificial immune system model for task allocation*.
- TARANAKOV, A., DASGUPTA, D. (2000). A formal of an artificial immune system. *Biosystems*, vol. 55/1-3, pp.151-158.
- TIAN, P., MA, J., ZHANG, D.M. (1999). Application of the Simulated Annealing Algorithm to the Combinatorial Optimization problem with permutation property: An investigation of generation mechanism. *European Journal of Operational Research*, 118- 81-94.
- TIMMIS, J., NEAL, M.J. (2000). Research and Development in Intelligent Systems XVII, *A resource limited artificial immune system for data analysis*, pp.19-32.
- TROJANOWSKI, K., WIERZCHON, S.T. (2002). The Eleventh International Symposium on Intelligent Information systems. *Searching for memory in artificial immune system*, June 3-6.