

## A New Artificial Immune System Algorithm for Multiobjective Fuzzy Flow Shop Problems

**Cengiz Kahraman**

*Department of Industrial Engineering, İstanbul Technical University, İstanbul, Turkey.*

*E-mail: kahramanc@itu.edu.tr*

*www.itu.edu.tr*

**Orhan Engin and Mustafa Kerim Yılmaz**

*Department of Industrial Engineering, Selçuk University, Konya, Turkey.*

*E-mail: orhanengin@yahoo.com, m\_kerim@hotmail.com*

*www.selcuk.edu.tr*

Received: 03/03/09

Accepted: 30/06/09

### Abstract

In this paper a new artificial immune system (AIS) algorithm is proposed to solve multi objective fuzzy flow shop scheduling problems. A new mutation operator is also described for this AIS. Fuzzy sets are used to model processing times and due dates. The objectives are to minimize the average tardiness and the number of tardy jobs. The developed new AIS algorithm is tested on real world data collected at an engine cylinder liner manufacturing process. The feasibility and effectiveness of the proposed AIS is demonstrated by comparing it with genetic algorithms. Computational results demonstrate that the proposed AIS algorithm is more effective meta-heuristic for multi objective flow shop scheduling problems with fuzzy processing time and due date.

*Key words:* Fuzzy flow shop, new artificial immune system, multi objective, engine cylinder liner manufacturing process.

### 1. Introduction

In a static permutation flow shop scheduling problem,  $n$  independent jobs  $\{J_1, \dots, J_n\}$  have to be processed in the same order on  $m$  different machines  $\{M_1, \dots, M_m\}$  and the objective is to minimize the total completion time (makespan) of  $n$  jobs<sup>1</sup>.

In a static permutation flow shop, the processing time for each job and due dates are usually assumed to be known exactly, but in many real world applications, processing times and due dates vary dynamically due to human factors or operating faults. In the literature,

fuzzy sets are used to model the uncertain processing times and due dates for the flow shop scheduling problems. The recent research works in terms of fuzzy processing times and due dates for flow shop scheduling are given in the following:

Nezhad and Assadi<sup>2</sup> proposed a method to approximate maximum operator number as a triangular fuzzy number and developed an algorithm based on this method. This approach was applied in flow shop scheduling and they modified Campbell Dudek and Smith<sup>3</sup> algorithm by using this operator.

Kiliç<sup>4</sup> presented a flowshop scheduling problem with fuzzy processing times and flexible due dates. The author proposed an ant colony optimization metaheuristic approach to solve the problems. Petrovic and Song<sup>5</sup> developed an optimization algorithm based on Johnson's algorithm for two machine flow shop problem with uncertain processing times. The solutions were compared with McCahon and Lee's<sup>6</sup> algorithm. Zhu et al.<sup>7</sup> studied fuzzy flow shop scheduling problems with distinct due window. Fuzzy time is denoted using triangular fuzzy numbers. They transformed the object function into the deterministic function and the artificial immune algorithm based information entropy is used to solve the model. Niu and Gu<sup>8</sup> proposed a genetic-based particle swarm optimization for no idle permutation flow shops with fuzzy processing time. Several benchmarks with fuzzy processing time are used to test genetic particle swarm optimization algorithm. The results are compared with genetic algorithm. Li and Wang<sup>9</sup> proposed a genetic algorithm for solving flow shop scheduling problem with fuzzy due window. A fuzzy logic controller that adjusts crossover probability and mutation probability in a real-time manner is designed to improve the algorithm's optimization performance. Song and Petrovic<sup>10</sup> proposed a fuzzy approach to capacitated flow shop scheduling problems. They used fuzzy sets to model the processing times of jobs and the setup times of machines required a new job starts processing. They developed a simulated annealing to search the best schedule. Temiz and Erol<sup>11</sup> proposed a fuzzy branch and bound algorithm for flow shop scheduling. The branch and bound algorithm of Ignall and Schrage<sup>12</sup> was modified and rewritten for three-machine flow shop problem with fuzzy processing time. They used fuzzy arithmetic and fuzzy numbers to determine the minimum completion time. Yao and Lin<sup>13</sup> investigated an approach for incorporating statistics with fuzzy sets for flow shop sequencing problems. They constructed a fuzzy flow shop sequencing model based on statistical data, which uses level  $(1-\alpha, 1-\beta)$  interval-valued fuzzy numbers to present the unknown job processing time. Ishibuchi et al.<sup>14</sup> formulated fuzzy flowshop scheduling problems with fuzzy processing time. They illustrated that a fuzzy flowshop scheduling problem with a single scheduling criterion had multiple non-dominated solutions.

Artificial immune system is a recent and hopeful metaheuristic algorithm for combinatorial optimization problems. Some previous publications have focused on applying AIS to different scheduling problems, for example, flow shop scheduling problem<sup>15</sup> the job shop scheduling problem<sup>16,17,18</sup> the hybrid flow scheduling problem<sup>19</sup> and the multiprocessor scheduling problem<sup>20, 21</sup>.

Due to human factors or operation faults, the processing times and due dates are unknown exactly in the multi objective flow shop problems. For this reason in the study, fuzzy sets are used in the model of multi objective flow shop problem and a new artificial immune system is developed for solving the multi objective flow shop scheduling problems with fuzzy processing time and due date.

The reminder of this study is organized as follows. In Sections 2 and 3, the proposed new AIS heuristic and genetic algorithms are presented, respectively. Multiobjective flow shop with fuzzy processing time and due date problem is defined in Section 4. The computational results of applying the new AIS to an engine cylinder liner manufacturing process are given in Section 5 and conclusions are derived in Section 6.

## 2. Artificial Immune System Algorithms

The vertebrate immune system is one of the most intricate bodily systems and these mechanisms demonstrated to be very interesting not only from a biological standpoint, but also under a computational perspective<sup>22</sup>.

AISs are defined as computational systems inspired by theoretical immunology and observed immune functions, principles, and models, applied to solve problems<sup>22</sup>. In the recent years, AIS algorithms have been extensively applied to optimization, computer and network security, pattern recognition, anomaly detection, data mining and scheduling problems.

The efficient mechanisms of an immune system, which are the clonal selection, learning ability, memory, robustness and flexibility make artificial immune systems useful for scheduling problems<sup>15</sup>. The proposed AIS is built on the clonal selection and affinity maturation principles.

### 2.1. Clonal Selection

The immune system is a complex system of cells and molecules distributed throughout our bodies that

provides us with a basic defense against bacteria, viruses, fungi, and other pathogenic agents. The immune system performs pattern recognition tasks, learns, and retains a memory of the antigens that it has fought<sup>23</sup>. The immune system mostly consists of the immune cells. The most common type of immune cells is white blood cells known as lymphocytes (B-cells and T-cells). These cells help in recognizing an almost limitless range of antigenic patterns. When a non-self antigen with threshold affinity is recognized by immune cell receptors, the antigen stimulates the B-cell to proliferate (divide) and mature into terminal antibodies (non-dividing) secreting cells, known as plasma cells<sup>24</sup>. The process of clonal selection is schematically shown in Fig. 1.

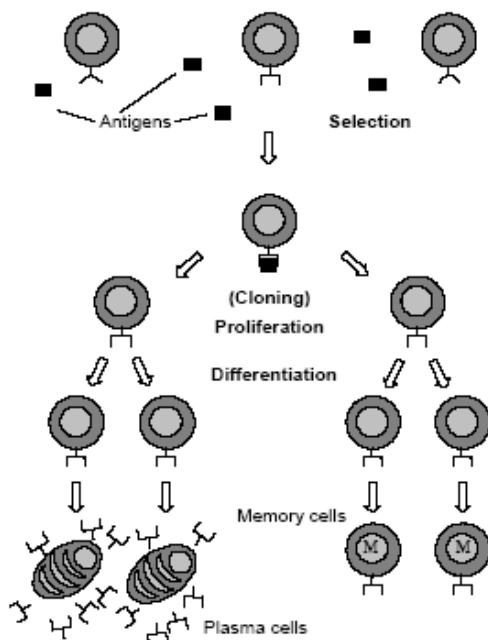


Fig. 1. The clonal selection principle<sup>25</sup>.

De Castro and Von Zuben<sup>24</sup> proposed a clonal selection algorithm. The steps of the algorithm are as follows:

1. Generate a set ( $P$ ) of candidate solutions, composed of subset of memory cells ( $M$ ) added to the remaining ( $P_r$ ) population.
2. Select the  $n$  best individuals of the population ( $P_n$ ), based on an affinity measure.
3. Clone (reproduce) these  $n$  best individuals of the population, giving rise to a temporary population of clones ( $C$ ). The clone size is an increasing function of the affinity with the antigen.

4. Submit the population of clones to a hypermutation scheme, where the hypermutation is proportional to the affinity of the antibody with antigen. A matured antibody population is generated ( $C^*$ ).
5. Re-select the improved individuals from  $C^*$  to compose the memory set  $M$ . Some members of  $P$  can be replaced by other improved members of  $C^*$ .
6. Replace  $d$  antibodies by diversity introduction. The lower affinity cells have higher probabilities of being replaced.

## 2.2. Affinity Maturation

Affinity maturation is the whole mutation process and the selection of the variant offspring that better recognizes the antigen<sup>26</sup>. The two basic mechanisms of affinity maturation are those: hypermutation and receptor editing<sup>25</sup>. Mutations take place in the variable region genes of antibody molecules. The mutation processes on lymphocytes are named as somatic hypermutation. The 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<sup>19</sup>. By the help of this strategy, the immune system keeps in hand the high affinity offspring cells and also ensures large mutations for the low affinity ones in order 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 developed harmful anti-self specificities. Those cells are eliminated by a programmed death process. But all cells with low affinities and anti-self specificities are not deleted, there is a process known as receptor editing: B-cells delete their self reactive receptors and developed entirely new receptors<sup>24</sup>.

## 3. Genetic Algorithm

Genetic algorithm (GA) is a stochastic search method developed to solve combinatorial optimization problems<sup>27</sup>. GA was invented by John Holland. GA is one of the best known metaheuristic methods for solving flow shop scheduling problems<sup>28</sup>.

A collection of solutions in the GA is called a population. The better solutions (individuals, strings or chromosomes) can be produced by GA's operators. The basic GA's operators are reproduction (selection) and recombination (crossover and mutation). These operators are used to construct new solutions (offsprings) from individuals of the current population, and

the solutions steadily improve from generation to generation<sup>29</sup>.

There is a few studies on GA for solving multiobjective flow shop scheduling problems in the literature. Pasupathy et al.<sup>30</sup> proposed a Pareto genetic algorithm for the problem of permutation flow shop scheduling with the objectives of minimizing the makespan and total flow time of jobs. Chang et al.<sup>31</sup> developed a sub-population genetic algorithm with mining gene structures for multiobjective flow shop scheduling problems.

There are also some other works on fuzzy GA: Li and Kwan<sup>32</sup> presented a hybrid GA for the bi-objective public transport driver scheduling problem. They used a greedy heuristic, which constructs a schedule by sequentially selecting shifts, from a very large set of pre-generated legal potential shifts, to cover the remaining work. Tang et al.<sup>33</sup> proposed using a fuzzy-genetic algorithm (GA) intelligent framework embedded with performance measurement. A fuzzy-GA approach was developed to include fuzzy rule sets with the associated membership functions in one chromosome. This approach was composed of two phases: knowledge representation and knowledge assimilation.

In this paper, the feasibility and effectiveness of the proposed AIS is demonstrated by comparing with the genetic algorithm. The proposed genetic algorithm is based on a permutation representation of the  $n$  jobs<sup>34</sup>. A direct coding approach is used. In this coding a chromosome represents a schedule directly. The initial population is randomly generated. The population size is kept constant through the generations.

In the study the GA parameters are selected as follows<sup>35</sup>. Initial population is 40; selection operator is roulette wheel; crossover operator is order crossover; mutation operator is inversion mutation; probability of crossover is 0.90; probability of mutation is 0.60; and termination condition is 250.

The selected parameter sets for GA are summarized as follows.

*Roulette wheel selection*

1. Let *pop-size*, number of strings in *pop*.
2. *nsum*, sum of all of the fitness value of the strings in *pop*; form *nsum* slots and assign string to the slots according to the fitness value of the string.
3. Do step 4 (*pop-size* - 1) times.

4. Generate a random number between 1 and *nsum*, and use it to index into the slots to find the corresponding string: add this string to *newpop*
5. Add the string with the highest fitness value in *pop* to *newpop*.

*Order Crossover*

1. Select a substring from one string at random,
2. Produce a new string by copying the substring into the position corresponding to those in the string,
3. Delete all of the symbols from the second string. The resulting sequence contains the symbols the new string needs,
4. Place the symbols into unfixed positions in the new string from left to right according to the order of the sequence to produce an offspring.

*Inversion mutation*

The inversion mutation can be seen from Fig. 2.

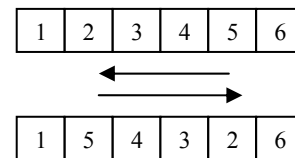


Fig. 2. The illustration of the inversion.

**4. Multiobjective Flow Shop with Fuzzy Processing Time and Due Date**

The formulation of a multiobjective fuzzy permutation flow shop scheduling problem is given in the following.

- $n$  number of jobs to be scheduled
- $m$  number of machines in the flow shop
- $\tilde{t}_{ij}$  fuzzy processing times of job  $i$  on machine  $j$
- $\tilde{d}_j$  fuzzy due date of job  $j$
- $\tilde{T}_j$  fuzzy tardiness of job  $j$
- $\tilde{C}_j$  fuzzy completion time of job  $j$

Fuzzy processing times  $\tilde{t}_{ij}$  are modeled by triangular membership functions represent by a triplet  $(t_{ij}^1, t_{ij}^2, t_{ij}^3)$ , where  $t_{ij}^1$  and  $t_{ij}^3$  are lower and upper bounds of the processing time. The triangular fuzzy processing time membership function is shown in Fig. 3. Also the due date  $\tilde{d}_j$  of each job is modeled by a trapezoidal fuzzy set and represented by a doublet  $(d_j^1, d_j^2)$ , where  $d_j^1$  is the crisp due date and  $d_j^2$  is the upper bound due date. The trapezoidal fuzzy membership function is shown in Fig. 4.

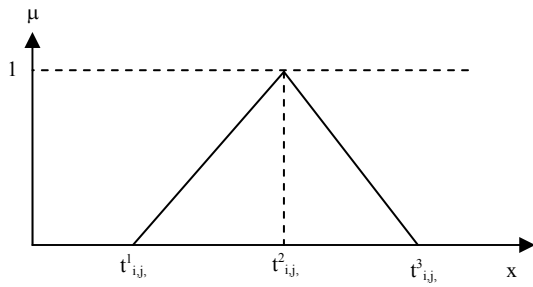


Fig. 3. Fuzzy processing time.

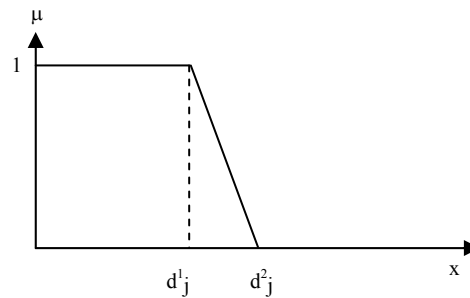


Fig. 4. Fuzzy due date.

In this study we consider minimizing the following two objectives<sup>36</sup>:

$$(1) \quad \text{to minimize the average tardiness } C_{AT}: \\ C_{AT} = \frac{1}{n} \sum_{j=1}^n T_j; \quad T_j = \max\{0, \tilde{C}_j - \tilde{d}_j\} \quad j = 1, \dots, n \quad (1)$$

$$(2) \quad \text{to minimize the number of tardy jobs } C_{NT}: \\ C_{NT} = \sum_{j=1}^n u_j \quad u_j = 1 \text{ if } T_j > 0, \text{ otherwise } u_j = 0 \quad (2)$$

For fuzzy multiobjective flow shop scheduling problems, we have the following assumptions.

- All processing times and due dates are fuzzy positive integer numbers;
- The number of jobs and machines are known and fixed during the schedule;
- The jobs must be carried out in a non preemptive way;
- Each machine can carry out at most one job at the same time;
- For carrying out these jobs all machines are continuously available;
- The processing times contain set up times for each job at each operation.

Set initial values:

- Number of jobs;
- Number of operations;
- Fuzzy processing times;
- Fuzzy due dates;
- Order quantity;

Set the AIS value:

- Number of iterations to determine the initial immune strings
- Initial parameter of the antibodies significance level
- Increasing the significance level of the antibody
- Number of iterations for AIS algorithm

Create a population of antibodies  $s$

## 5. Application in an Engine Cylinder Liner Manufacturing Process

### 5.1. Proposed new AIS algorithm

#### 5.1.1. Algorithm

In the study, the schedules of jobs are represented by strings of length  $s = n * m$  (jobs \* machines). The  $s$  elements of the strings are the jobs which will be sequenced.

These strings are accepted as antibodies of the AIS. The AIS algorithm goes up to solution by the evolution of these antibodies. The proposed procedure is given below;

For each generation do:

- Decode the antibodies in the antibody population;
- Determine the satisfaction grade of average tardiness;
- Determine the satisfaction grade of number of tardy jobs;
- Determine the immune strings,  $s$
- Cloning (generate copies of the antibodies)

For each generated clone do:

- Immune based Mutation (generate a new string)
- Decode the new string;
- Calculate the satisfaction grade of average tardiness;
- Calculate the satisfaction grade of number of tardy jobs;
- If the satisfaction grade of average tardiness (new string) > the satisfaction grade of average tardiness (clone) then update the table
- If the satisfaction grade of number of tardy jobs (new string) < the satisfaction grade of number of tardy jobs  $j$  (clone) then check the antibodies significance level.

The suggested original artificial immune system is formed with two basic principles. They are listed in the following;

1. The immune system is transferred to the new members or the children from the parents. The new member has the immune system of the parents: Cloning.
2. The weak immune system of the newly formed member becomes stronger in the course of time and gains immunity against viruses as in the example of vaccine.

The suggested algorithm is briefly explained in the following example.

#### 5.1.2. Example

Let's consider a 3-jobs  $\times$  2-machine type of flow shop scheduling problem. Since the problem is "flow type", our operations will be treated in the first machine then in the second machine before leaving the system. There exist six operations for this problem (number of operations = number of works  $\times$  number of machines). In this suggested new mutation method, an operation ordering will be carried out. The steps of the algorithm are presented in the following.

**Step 1** Determine the set of operations that will be scheduled. Since we have 3 works in our example and each work's first operation is required to be scheduled, the set of scheduled operations will be as in the following.

The term 11 in the set of  $S_1 = \{11, 21, 31\}$  means that the first operation is executed in the first machine, and the terms 21 and 31 indicate that the second and third

operations are respectively executed in the first machine. When it is necessary to execute all the operations of this set in the first machine, only one of the operations can be executed initially in the first machine. One of the operations existing in the  $S_1$  set is selected randomly and become the first member (gene) of the ordered operations series.

For instance, if we select the operation 21 from set  $S_1$ , the set of scheduled operations will be  $S_2$  as in the following.

That's why the operations of 11 and 31 were not executed before, they directly took place in the set of  $S_2 = \{11, 22, 31\}$ . Additionally, since the operation 21 was previously executed, the set of  $S_2$  did not involve the operation 21 but the next operation namely 22 that represents the second work's second operation.

The foregoing operations are continued until we have no member in the set of scheduled operations. At the end of Step 1, an operation series (chromosome) with which all the operations were put in order will be obtained. Now, determine the fitness function of this operation series i.e. chromosome (makespan -  $C_{max}$  or number of tardy jobs or total tardiness can be the fitness function for the scheduled problem)

**Step 2.** Similar to Step 1, form the second operation series (another chromosome), and calculate the fitness function of the obtained chromosome. Determine the percent deviation values of the fitness functions of both chromosomes by using the following expression. In this study, for the measure of deviation, we used Dubois et al.'s<sup>37</sup> and Sakawa et al.'s<sup>38</sup> approaches.

$$\text{deviation \%} = \frac{\text{fitness function of the first chromosome} - \text{fitness function of the second chromosome}}{\text{fitness function of the second chromosome}} \times 100$$

**Step 3.** Draw a table including the percent deviation values. The number of columns and rows of this table should be equal to the number of operations. For example; our problem with 3-jobs  $\times$  2-machines will have 6-rows and 6-columns. Write the operations to the rows and the columns of the table by considering the priority order and beginning from the first work. In other words, write all the operations of the first work in order, and then write all the operations of the second work.

For our example, 3-jobs  $\times$  2-machines, the following operations will take place in order in the rows and columns of the table.

Operations: 11, 12, 13, 21, 22, 23, 31, 32, 33; in this series, the term 11 is the first work's first operation, 12 is the first work's second operation, and similarly 33 is the third work's third operation.

Write the percent deviation values of each operation to the related row and column of the table considering the second operation series or chromosome.

**Step 4.** Input the number of chromosomes for the formation period of the initial immune system. Go to Step 1 as many as the assigned number of chromosomes; form the operation series (chromosomes); calculate the percent deviation values in Step 2 and add the calculated percent deviations into the percent deviation table of Step 3.

Since the percent deviation values can be positive or negative, the positive values are added to the previous percent deviation value, and the negative ones are subtracted from the previous percent deviation values.

During the determination of the percent deviation values, the comparison is always made with the series having the most appropriate fitness function that was ever found.

When the number of chromosomes assigned for the initial solution is reached i.e. all the chromosomes in the population are formed, then go to Step 5.

**Step 5.** The initial parameter of the antibody significance level is assigned as 0 or 1. The value 0 for the antibody significance level parameter means the unreliable sum of percent deviation values, while the value 1 means the completely reliable significance level. The sum of the percent deviation values calculated for each operation in the table including the percent deviation values is multiplied by the initial antibody significance level parameter. Therefore, the parameters with specified significance levels are

obtained for each operation of the chromosomes. Then go to Step 6.

**Step 6.** Similar to the procedure of Step 1, determine the set of operations that will be scheduled. Since we have 3 jobs and the first operation of each job should be scheduled initially, the set of scheduled operations will become as in the following;

The selection of any operation existing in the  $S_1$  set is performed in Step 1 randomly, however the parameters whose significance level is specified for the selection of the appropriate operation are considered. The appropriate operation is selected using the roulette wheel selection method (used in genetic algorithms) and considering the parameters in Step 5. The selection probability of the operations with high parameters of specified significance levels will be always greater than the others.

Applying the above procedure until no member is left in the scheduled operations set; an operation series (chromosome) is obtained in which all the operations of the works are ordered. Then the fitness function of this operation series (or chromosome) is determined (the fitness function for the scheduled problem can be makespan- $C_{\max}$  or the number of tardy jobs or total tardiness).

The table including the percent deviation values is updated as in Steps 3 and 4.

**Step 7.** The initial parameter of the antibody's significance level is usually selected as 0.1 and increased up to 1 with 0.1 increments.

This process is performed logarithmically as in the following;

There are 10 values from 0.1 to 1 with 0.1 increments, and thus  $\frac{n(n+1)}{2} = \frac{10(10+1)}{2} = 55$  is found, and

for the total number of iterations,  $\frac{100}{55} \cong 2$  is

determined. Before the first increase,  $2 \times 10 = 20$  iterations, before the second increase,  $2 \times 9 = 18$  iterations, and before the third increase,  $2 \times 8 = 16$  iterations should be executed.

The procedure in Step 6 is repeated after increasing the significance level of the antibody. Step 8 is applied when the significance level of the antibody is 1.

**Step 8.** The iteration is stopped, and the fitness function having the most appropriate series is assumed to be the solution of the problem.

In the suggested original artificial immune system, first of all, the immunity performance of the new member is formed randomly, and then this random system is strengthened with small improvements and local solutions in the subsequent steps.

5.1.3. Cloning selection process

Fitness value

The fitness value aggregates the Satisfaction Index (SI) of two objectives. The satisfaction indexes are calculated taking into consideration the completion times of the jobs. The question arises how to compare a fuzzy completion time of a job with its fuzzy due date, i.e. how to calculate the likelihood that a job is tardy<sup>36</sup>. In this study, two approaches are used;

- Based on the possibility measure introduced by Dubois et al<sup>37</sup>. and
- Based on the area of intersection introduced by Sakawa et al<sup>38</sup>.

1. The possibility measure

The possibility measure approach was also used by Itoh et al<sup>39</sup>. The possibility measure  $\pi_{\tilde{C}_j}(\tilde{d}_j)$  of a fuzzy event,  $\tilde{C}_j$  on a fuzzy set  $\tilde{d}_j$  is defined as follow.<sup>39</sup>

$$\pi_{\tilde{C}_j}(\tilde{d}_j) = \sup \min \{ \mu_{\tilde{C}_j}(t), \mu_{\tilde{d}_j}(t) \} \quad (3)$$

$j = 1, \dots, n$

It is used to measure the satisfaction grade of a fuzzy completion time  $SG_T(\tilde{C}_j)$  of job<sub>j</sub>

$$SG_T(\tilde{C}_j) = \pi_{\tilde{C}_j}(\tilde{d}_j) \quad (4)$$

Where  $\mu_{\tilde{C}_j}(t)$  and  $\mu_{\tilde{d}_j}(t)$  are the membership functions of fuzzy sets  $\tilde{C}_j$  and  $\tilde{d}_j$ , respectively.<sup>36</sup>

The possibility measure of the fuzzy due date  $\tilde{d}_j$ , it is illustrated in Fig. 5.

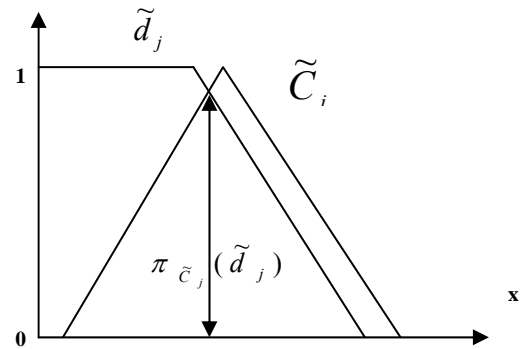


Fig. 5. The possibility measure  $\pi_{\tilde{C}_j}(\tilde{d}_j)$  of the fuzzy due date  $\tilde{d}_j$ .

2. Area of intersection measure approach

The area of intersection measures the portion of  $\tilde{C}_j$ , that is completed by the due date  $\tilde{d}_j$ .<sup>38</sup> It is shown in Fig. 6. The satisfaction grade of a fuzzy completion time of job<sub>j</sub> is defined as follow.<sup>36</sup>

$$SG_T(\tilde{C}_j) = \frac{(area \tilde{C}_j \cap \tilde{d}_j)}{(area \tilde{C}_j)} \quad (5)$$

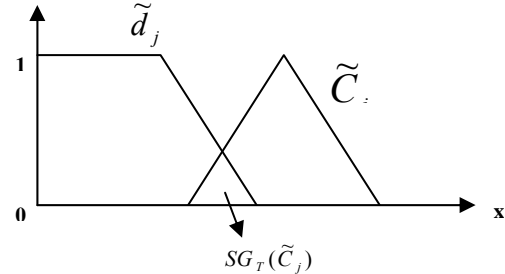


Fig. 6. Satisfaction grade of completion time using area of intersection.

The equations given in (3) and (4) are transformed into the objectives to maximize their corresponding satisfaction grades as follow<sup>36</sup>:

1. Satisfaction grade of Average Tardiness

$$S_{AT} = \frac{1}{n} \sum_{j=1}^n SG_T(\tilde{C}_j) \quad (6)$$

2. Satisfaction grade of number of tardy jobs: A parameter  $\lambda$  is introduced such that a job  $J_j, j=1, \dots, n$  is considered to be tardy if  $SG_T(\tilde{C}_j) \leq \lambda, 0 \leq \lambda \leq 1$ . After calculating the number of tardy jobs  $n$  tardy, the satisfaction grade  $S_{NT}$  is given as in Eq. (7)<sup>36</sup>:



$$S_{NT} = \left\{ \begin{array}{lll} 1 & \text{if} & nTardy = 0 \\ (n'' - nTardy) / n'' & \text{if} & 0 < nTardy < n'' \\ 0 & \text{if} & nTardy > n'' \end{array} \right\} \quad (7)$$

where  $n'' = 15\%$  of the total number of jobs. In the study, three different aggregation operators are investigated. These are given in the following<sup>36</sup>:

1. Average of the satisfaction grades:

$$F_1 = \frac{(S_{AT} + S_{NT})}{2} \quad (8)$$

2. Minimum of the satisfaction grades:

$$F_2 = \min(S_{AT}, S_{NT}) \quad (9)$$

3. Average weighted sum of the satisfaction grades:

$$F_3 = 1/2(w_1 S_{AT} + w_2 S_{NT}) \quad (10)$$

where  $w_k \in [0,1]$ ,  $k = 1,2$ , are normalized weights randomly chosen used in the GA and changed in every iteration in order to explore different areas of the search space<sup>36, 40</sup>.

### 5.2. An engine cylinder liner manufacturing process

The proposed fuzzy approach for multiobjective flow shop scheduling model is tested on real world data collected at an engine cylinder liner manufacturing firm in Konya industry area, Turkey. Engine cylinder liner is the most important part working in an engine. The engine cylinder liner is shown in Fig. 7. The engine cylinder liners are processed on the machines which are equipped with the computer controlled machinery using the latest technology. They are processed 8 or 11 different operations.



Fig. 7. Engine cylinder liner.

These operations are explained roughly as shown in Fig. 8.

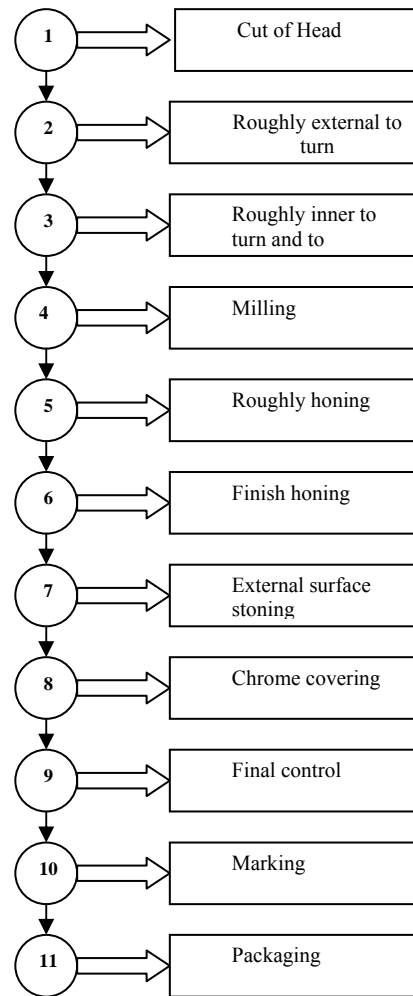


Fig. 8. Operations of engine cylinder liner.

### 5.3. Experiments and results

The developed new AIS algorithms are tested on real-world data collected at an engine cylinder liner manufacturing company. For obtaining optimal or near optimal solutions of multi objective fuzzy flow shop scheduling by new AIS algorithm in the shortest time, best set of parameters are used. These parameters are determined as follows.

5.1.3. Parameterization of the AIS algorithm by design of experiments

In the study, full factorial design of experiments (DOE) has been used. The application involves four parameters (factors) with different possible values each. These parameters are given in Table 1.

The best parameter sets for proposed new AIS are found as shown in Table 2.

Also in the study, two values are tested for  $\lambda$ : ( $\lambda=0.4$  and  $0.7$ )<sup>36</sup>. The experimental result shows that the higher

values of  $\lambda$  ( $\lambda = 0.7$  and more) find the better solution for the three aggregation operators. In the study we used the value of  $\lambda = 0.7$ .

In the study, a multi objective flow shop scheduling problem with fuzzy processing times and due dates has been formulated for an engine cylinder liner manufacturing process. The algorithm was implemented in Borland Delphi and executed with a PC of Pentium 4 with 3 GHz processor and 512 MB memory.

Table 1. Parameters of the proposed AIS for multiobjective fuzzy flow shop scheduling.

Parameter	Range
Number of iteration to determine the initial immune strings	500, 1000, 1500, 2000, 2500, 3000
Initial parameter of the antibodies significance level	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
The ratio of increasing the significance level of the antibody	0.1, 0.2, 0.3, 0.4
Number of iteration	10, 30, 50, 100, 175, 250

Table 2. The best parameter sets for the proposed AIS algorithm.

Parameter	Value
Number of iteration to determine the initial immune strings	1000
Initial parameter of the antibodies significance level	0.6
The ratio of increasing the significance level of the antibody	0.1
Number of iteration	100

Multi objective fuzzy flow shop scheduling problems are formulated by two objectives. These are to minimize the average tardiness and to minimize the number of tardy jobs. In the study the fitness value of proposed new AIS aggregates the satisfaction index of these two objectives. To compare the fuzzy completion time of a job with its fuzzy due date, two approaches are used. These are PM introduced by Dubois et al.<sup>37</sup> and AIM introduced by Sakawa et al.<sup>38</sup>.

In this research, three different aggregation operators are investigated. These are average of the satisfaction grades  $F_1$ , minimum of the satisfaction grades  $F_2$  and average weighted sum of the satisfaction grades  $F_3$ .

The multi objective fuzzy flow shop scheduling problems are solved by the proposed AIS and GA. These three aggregation operators' averages, standard deviations, and maximum values are presented in Table 3.

In this study, the improvement rate of the proposed new AIS with respect to GA for each aggregation operator is also presented in Table 3. The improvement rate is defined as in the following.

$$improvement\ rate = \frac{AIS - GA}{GA} \tag{11}$$

As it is seen in Table 3, the proposed new AIS found better solutions for the three aggregation operators. The minimum and maximum improvement rates are found 0.62 and 2.75 respectively. The proposed AIS found the best solutions for the minimum and maximum of the satisfaction grades and the average weighted sum of the satisfaction grades.

**Acknowledgements**

In the paper a new AIS algorithm has been proposed to solve multi objective fuzzy flow shop scheduling problems. Fuzzy sets have been used to model processing times and due dates. The considered problem is NP-hard. In the study, two objectives, which are average tardiness and the number of tardy jobs, are considered to minimize. The problem is a real world problem examined in an engine cylinder liner manufacturing process.

The results of this new AIS algorithm have been compared with GA solutions. The percentages improved from GA have been calculated. The proposed

algorithm found better AIM and PM average and max values for the average of the satisfaction grades, the minimum of the satisfaction grades and the average weighted sum of the satisfaction grades. It is a good problem solving technique for fuzzy multi objective flow shop scheduling problems.

In the future, some changes may be done in the mutation procedure and selection procedure may be improved. Also the proposed AIS algorithm may be hibritted with another metaheuristic method to improve the solution quality.

Table 3. Average and best values of satisfaction grades for AIS and GA

Fitness value		GA		New AIS algorithm		Improvement Rate	
		AIM	PM	AIM	PM	AIM	PM
Average of the satisfaction grades	Average	0.48241	0.48001	0.97899	0.98144	1.03	1.04
	Std. Dev.	0.12676	0.09894	0.00976	0.01046	AIS better	AIS better
	Max	0.61360	0.60876	0.99694	0.99947	0.62	0.64
Minimum of the satisfaction grades	Average	0.26667	0.30601	1.00000	1.00000	2.75	2.27
	Std. Dev.	0.13561	0.09220	0.00000	0.00000	AIS better	AIS better
	Max	0.26667	0.33333	1.00000	1.00000	2.75	2.00
Average weighted sum of the satisfaction grades	Average	0.59333	0.60222	1.00000	1.00000	0.69	0.66
	Std. Dev.	0.07160	0.05497	0.00000	0.00000	AIS better	AIS better
	Max	0.59333	0.61667	1.00000	1.00000	0.69	0.62

References

1. Y. Gajpal, C. Rajendran, An ant colony optimization algorithms for minimizing the completion time variance of jobs in flow shops, *International journal of production economics*, **101** (2006) 259–272.
2. S.S Nezhad, R.G. Assadi, Preference ratio based maximum operator approximation and its application in fuzzy flow shop scheduling, *Applied Soft Computing*, **8** (2008) 759–766.
3. H.G. Campbell, R. A. Dudek, M. L. Smith, A heuristic algorithm for the n-job, m-machine sequencing problem, *Management science*, **16** (1970) 630–637.
4. S. Kiliç, Scheduling a fuzzy flowshop problem with flexible due dates using ant colony optimization, *Applications of Evolutionary Computing*, **4448** (2007) 742–751.
5. S. Petrovic, X. Song, A new approach to two machine flow shop problem with uncertain processing times, *Optimization and Engineering* **7**, **3** (2006) 329–342.
6. S. McCahon, E.S. Lee, Job sequencing with fuzzy processing times, *Comput Math Appl* **19**, **7** (1990) 31–41.
7. J. Zhu, G. Du, L. Wang, Artificial immune algorithm for fuzzy flow shop scheduling problem, *Dynamics of Continuous Discrete and Impulse Systems-Series B-Applications & Algorithms* **13** (2006) 383–386.
8. O. Niu, X.,S. Gu, An improved genetic-based particle swarm optimization for no idle permutation flow shops with fuzzy processing time, *PRICAI, Lecture Notes in Artificial Intelligence*, **4099** (2006) 757–766.
9. H. Li, L. Wang, Flow shop scheduling problem with fuzzy due window, *Dynamics of Continuous Discrete and Impulse Systems-Series B-Applications & Algorithms*, **13** (2006) 370–374.
10. X. Song, S. Petrovic, A fuzzy approach to capacitated flow shop scheduling, *Proceedings of the 11<sup>th</sup> International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU)*, Paris, France, July 2–7, (2006) 486–93.
11. I. Temiz, S. Erol, Fuzzy branch and bound algorithm for flow shop scheduling, *Journal of Intelligent Manufacturing*, **15** (2004) 449–454.
12. E. Ignall, L. Schrage, Application of the branch and bound technique to some flow shop scheduling problems, *Operations research*, **13** (1965) 400–412.
13. J.S. Yao, F.T. Lin, Constructing a fuzzy flow shop sequencing model based on statistical data, *International Journal of Approximate Reasoning*, **29** (2002) 215–234.
14. H. Ishibuchi, T. Murata, K.H. Lee, Formulation of fuzzy flowshop scheduling problems with fuzzy processing time, *Proceedings of 1996 IEEE International Conference on Fuzzy Systems* (New Orleans, USA) (1996) 199–205.
15. O. Engin, A. Döyten, A new approach to solve flow shop scheduling problems by artificial immune systems, *Journal of the Doğuş University*, **8**, **1** (2007) 12–27.

16. C.A.C. Coello, D.C. Rivera, N.C. Cortes, Use of an artificial immune system for job shop scheduling, artificial immune systems, *proceedings lecture notes in computer science*, **2787** (2003) 1–10.
17. H.W. Ge, L. Sun, Y.C. Liang, Solving job shop scheduling problems by a novel artificial immune system, *Lecture Notes in Artificial Intelligence*, **3809** (2005) 839–842.
18. M. Chandrasekaran, P. Asokan, S. Kumanan, T. Balamurugan, S. Nickolas, Solving job shop scheduling problems using artificial immune system, *Int J Adv Manuf Technol*, **31** (2006) 580–593.
19. O. Engin, A. Döyten, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, *Future Generation Computer Systems*, **20**, **6** (2004) 1083–1095.
20. A. Swiecicka, F. Seredynski, A.Y. Zomaya, Multiprocessor scheduling and rescheduling with use of cellular automata and artificial immune system support, *IEEE Transactions on Parallel and Distributed Systems*, **17**, **3** (2006) 253–263.
21. G. Wojtyla, K. Rzdaca, F. Seredynski, Artificial immune systems applied to multiprocessor scheduling, Parallel processing and applied mathematics *Lecture Notes in Computer Science*, **3911** (2006) 904–911.
22. L.N. De Castro, J.I. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer-Verlag (2002).
23. A. Perelson, G. Weisbuch, Immunology for physicists, *Reviews of Modern Physics*, **69**, **4** (1997) 1219–1267.
24. L.N. De Castro, F.J. Von Zuben, The clonal selection algorithm with engineering applications, in: *Proceedings of the workshop on GECCO*, Las Vegas, July 8- 12 (2000) 36–37.
25. L.N. De Castro, F.J. Von Zuben, Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, **6**, **3** (2002) 239–251.
26. L.N. De Castro, J.I. Timmis, Artificial Immune systems as a novel soft computing paradigm. *Soft Computing, A Fusion of Foundations, Methodologies and Applications Journal*, **7** (8), (2003) 526–544.
27. D.E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley Publishing, The University of Alabama, 1989.
28. C.R. Reeves, Genetic algorithm for flow shop sequencing, *Computers and Operations Research*, **15** (1995) 5–23.
29. S.H. Yoon, J.A. Ventura, An application of genetic algorithms to lot-streaming flow shop scheduling, *IIE Transactions*, **34** (2002) 779–787.
30. T. Pasupathy, C. Rajendran, R.K. Suresh, A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs, *Int J Adv. Manuf Technol*, **27** (2006) 804–815.
31. P.C. Chang, S.H. Chen, C.H. Liu, Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems, *Expert systems with applications*, **33** (2007) 762–771.
32. J. Li, R. S. K. Kwan, A fuzzy genetic algorithm for driver scheduling, *European Journal of Operational Research* **147**, **2**, (2003) 334–344.
33. C.X.H. Tang, H.C.W. Lau, G.T.S. Ho, A conceptual fuzzy-genetic algorithm framework for assessing the potential risks in supply chain management, *International Journal of Risk Assessment and Management* **10**, **3** (2008) 263–271.
34. C. Kahraman, O. Engin, İ. Kaya, M.K. Yılmaz, An application of effective genetic algorithms for solving hybrid flow shop scheduling problems, *International Journal of Computational Intelligence Systems*, **1**, **2** (2008) 134–147.
35. O. Engin, *To increase the performance of flow shop scheduling problems solving with genetic algorithms: a parameters optimization*, PhD. Thesis, Istanbul Technical University, Institute of Science and Technology, Istanbul, Turkey (2001).
36. C. Fayad, S. Petrovic, A fuzzy genetic algorithm for real-world job shop scheduling, *Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, Bari, Italy (2005) 524–533.
37. D. Dubois, H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, New York. (1988).
38. M. Sakawa, R. Kubota, Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms, *European Journal of Operational Research* **120** (2000) 393–407.
39. T. Itoh, H. Ishii, Fuzzy due date scheduling problem with fuzzy processing time, *International transaction in operations research*, **6** (1999) 639–647.
40. T. Murata, H. Ishibuchi, H. Tanaka, Multi objective genetic algorithm and its applications to flowshop scheduling, *Computers Industrial Engineering*, **30**, **4** (1996) 957–968.