Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007)

Solving a Bi-Criteria Permutation Flow Shop Problem Using Immune Algorithm

Reza Tavakkoli-Moghaddam[1], Ali-Reza Rahimi-Vahed[2], and Ali Hossein Mirzaei[3]

***Abstract*—A flow shop problem as a typical manufacturing challenge has gained wide attention in academic fields. In this paper, we consider a bi-criteria permutation flow shop scheduling problem, in which the weighted mean completion time and the weighted mean tardiness are to be minimized simultaneously. Since a flow shop scheduling problem has been proved to be NP-hard in strong sense, an effective multi-objective immune algorithm (MOIA) is proposed for searching locally Pareto-optimal frontier for the given problem. To prove the efficiency of the proposed algorithm, a number of test problems are solved and the efficiency of the proposed algorithm, based on some comparison metrics, is compared with a distinguished multi-objective genetic algorithm, i.e. SPEA-II. The computational results show that the proposed MOIA performs better than the above genetic algorithm, especially for large-sized problems.**

I. INTRODUCTION

FLOW-SHOP scheduling problems address a determination of sequencing a number of jobs that have to be processed on a number of machines so that performance measures, such as makespan, tardiness, etc., are optimized. In flow shop scheduling, the processing routes are the same for all the jobs. Recently, the flow shop scheduling problems have been one of the most prevalent problems in the area of scheduling and there are numerous papers that have investigated this issue [1]. A comprehensive review of flow shop related papers over the last 50 years are provided in [2].

In the general flow shop, passing is allowed. In other words, the job sequence on each machine may be different. However, in some practical cases the sequencing of different jobs that visit a set of machines must be in the same order. This class of flow shop problems is referred as permutation flow shop [3]. Some papers considered the permutation flow shop scheduling problem. Nowicki and Smutnicki [4] addressed this problem with respect to makespan criterion and proposed an improved tabu search to solve it. Suliman [5] proposed a two-phase heuristic approach to tackle the same problem. Cheng et al. [6] addressed the three machine permutation flow shop problem with release times where the objective is to minimize makespan. They proposed a branch and bound algorithm for solving this problem. Grabowski and Wodecki [7] proposed a tabu search based algorithm for the permutation flow shop problem with makespan criterion. Lian et al. [8] proposed a particle swarm optimization algorithm for the same problem.

While there are many studies treated a single objective, consideration of multiple criteria is more realistic practically [1]. Murata et al. [1] proposed a multi objective genetic algorithm to tackle flow shop scheduling problem. They considered the problem with two objectives of minimizing makespan and total tardiness and then they investigated the problem with respect to minimizing makespan, total tardiness and total flowtime as objectives. Ponnambalam et al. [9] proposed a TSP-GA multi objective algorithm for flow shop scheduling where they use a weighted sum of multiple objectives (i.e. minimizing makespan, mean flow time and machine idle time). Toktas et al. [10] considered the two machine flow shop scheduling problem by minimizing makespan and maximum earliness as objectives. Ravindran et al. [11] proposed three heuristic algorithms to solve the flow shop scheduling problem which in makespan and total flow time have been considered as objectives.

In this paper, we deal with a bi-criteria permutation flow shop scheduling problem where the weighted mean completion time and the weighted mean tardiness are considered as objective functions. To tackle this problem, an improved multi-objective immune algorithm (MOIA) is developed for searching locally Pareto-optimal frontier. The remainder of this paper is organized as follows: Section 2 gives the problem definition. In Section 3, the background of immune algorithm is described and then the proposed algorithm is given. The experimental results are provided in Section 4. Finally, Section 5 consists of conclusions.

II. PROBLEM DEFINITION

A. Permutation flow shop scheduling problem

The addressed scheduling problem can be described as follows: Consider an n job m machine permutation flow shop scheduling problem where each of n jobs has to be processed, without preemption, on m machines in the same order, i.e. passing is not allowed. Job j, $j \in n$, consists of a sequence of m operations O_{j1}, O_{j2}, … , O_{jm} that operation O_{jk} corresponds to the processing of job j on machine k during an uninterrupted processing time p_{jk}. The problem is considered under the following assumptions: All jobs are

[1]Department of Industrial Engineering, University of Tehran, P.O. Box: 11365/4563, Tehran, Iran. Corresponding author e-mail: tavakoli@ut.ac.ir (R. Tavakkoli-Moghaddam).

[2]Department of Industrial Engineering, University of Tehran, P.O. Box: 11365/4563, Tehran, Iran.

[3]Department of Industrial Engineering, Tarbiat Modares University, P.O.Box: 14115/111, Tehran, Iran.

available at zero time, Machines are always available, The processing time of each job on each machine is known and constant, Setup times and removal times are included in processing times, Preemption is not allowed, Passing is not allowed, Transportation times are negligible, Each job may have its own due date, Each machine can process only one job at the same time, A job can not processed on more than one machine at the same time, Jobs can wait between two successive machines and intermediate storage exists.

B. Objective functions

Assume that C_i, d_i and w_i denote the completion time, the due date and a possible weight associated to job *i*, respectively. The first considered objective function is the minimization of the weighted mean completion time. This objective can be calculated by the following equation:

$$\frac{\sum_{i=1}^{n} w_i C_i}{W} \tag{1}$$

Where W is the sum of jobs' weights; that is $W = \sum_i w_i$. Another objective considered is the minimization of the weighted mean tardiness. To calculate the value of this objective, the subsequent equation is used:

$$\frac{\sum_{i=1}^{n} w_i T_i}{W} \tag{2}$$

Where W is as the same equation (1) and T_i is the tardiness for job *i* and equals to $\max\{0, C_i - d_i\}$.

III. IMMUNE ALGORITHM

A. Immune algorithm in general

The biological immune system is a robust, complex, adaptive system that defends the body from foreign pathogens. The clonal selection and affinity maturation principles are used to explain how the immune system reacts to pathogens and how it improves its capability of recognizing and eliminating pathogens [12]. Clonal selection states that by pathogen invasion, a number of immune cells (lymphocytes) that recognize these pathogens will proliferate; some of them will become effecter cells (plasma cells), while others will be maintained as memory cells. The effecter cells secrete antibodies in large numbers, and the memory cells have long life spans so as to act faster and more effectively in future exposures to the same or a similar pathogen [13]. During cellular reproduction, the cells suffer somatic mutations at high rates, together with a selective force; the cells with higher affinity to the invading pathogen differentiate into memory cells. Generally, cells with low affinity receptors are mutated at a higher rate, whereas cells with high affinity receptors will have a lower mutation rate [14]. This whole process of somatic mutation plus selection is known as affinity maturation [13].

A novel computational intelligence technique, inspired by immunology, has emerged, known as Artificial Immune Systems (AIs). Recently, it has advocated special attention to itself in order to various applications. Luh et al. [15] proposed an immune based algorithm for finding Pareto optimal solutions to multi-objective optimization problems. Coello Coello and Cortes [16] applied clonal selection principle to solve multi-objective optimization problems. Khoo and Situmdrang [14] dealt with the design of assembly system for modular products by using an approach based on the principles of natural immune systems. Engin and Doyen [17] dealt with the hybrid flow shop scheduling problem where they applied clonal selection principle and affinity maturation mechanism in order to solve the problem. Kumar et al. [18] used artificial immune system to tackle a continuous flow shop problem with total flow time as criterion. Zandieh et al. [13] used the immune algorithm for solving the hybrid flow shop scheduling problems where setup times depended on sequence.

{Initialize search parameters
Create the initial antibody repertoire with elite tabu search
Initialize the adaptive Pareto archive set so that is empty
For 1 to *MaxIter* (the maximum number of iterations)
  Perform non-dominated sorting
  Update the adaptive Pareto archive set
  While (pool size is not reached)
    The high affinity antibodies, including both dominated and non-dominated antibodies, are cloned and added to the Pool
  End While
  While (Hypermutation rate is not satisfied)
    Perform swapping mutation on selected antibody
  End While
  While (Combination rate is not met)
    Select a prespecified number of antibodies from the pool
    Perform linear combination method on the selected antibodies to generate a new antibody
  End While
End For}

Fig. 1. The general scheme of MOIA

B. The Proposed Multi-Objective Immune Algorithm

The proposed algorithm is based on the clonal selection principle, modeling the fact that only the highest affinity antibodies will proliferate. The distinguishing criterion between antigens and antibodies is Pareto dominance. In other words, non-dominated solutions are the antigens and dominated solutions are the antibodies. The multi-objective immune algorithm (MOIA) implementation is described in the following sections. The general scheme of the proposed algorithm is provided in Fig.1.

C. Antibody Representation

Two kinds of different antibody representations are used simultaneously in this paper. Job-to-position and continuous

representation. Each antibody concurrently has a job-to-position and continuous representation, each of them is used in different steps in our algorithm.

In the Job-to-position representation, as shown in Fig. 2, a single row array of the size equal to the number of the jobs to be scheduled is considered. The value of the first element of the array shows which job is scheduled first. The second value shows which job is scheduled second and so on.

Location in a sequence	1	2	3	4	5	6	7
Job to be scheduled	1	2	4	3	5	6	7

Fig.2. Job-to-position representation for a flow shop scheduling problem

Tasgetiren et al [19] devised a new way of representation for scheduling problems using continuous values. In this paper, a modified version of this representation is provided.

Consider the sample job-to-position representation illustrated in Fig. 2. To construct the continuous version of this representation, we first need to generate 7 (as many as the number of the jobs to be produced) random numbers between $[0, x_{\max}] = [0,4]$, then these numbers will be sorted and the first smallest of them will be assigned to the position that contain the first job, that is job number 1, the next smallest will be assigned to position that contain the second job, that is job number 2 and so on. Suppose the numbers shown in Table I are the random numbers obtained.

TABLE I
A SAMPLE SET OF RANDOM NUMBERS

No.1	No.2	No.3	No.4	No.5	No.6	No.7
0.46	2.96	1.77	2.49	1.54	3.61	2.88

To build the continuous representation, we have to assign 0.46 to job number 1, 1.54 to job number 2, 1.77 to job number 3 and so on. Thus, Fig. 3 shows the associated representation.

Location in a sequence	1	2	3	4	5	6	7
Continuous representation	0.46	1.54	2.49	1.77	2.88	2.96	3.61

Fig.3. Continuous representation of Fig. 2

To illustrate how the job-to-position representation is obtained from the representation shown in Fig. 3, we just need to schedule the first job in the place of the first smallest values of the continuous representation, the second job in the place of the next smallest values of the continuous representation and so on.

D. Antibody Initialization

Most of the evolutionary algorithms use a random procedure to generate an initial set of solutions. However, since the output results are strongly sensitive to the initial set, we propose a new Elite Tabu Search (ETS) mechanism to construct this set of solutions. The main purpose of applying this meta-heuristic is to build a set of potentially diverse and high quality antibodies in the job-to-position representation form. Before describing the elements of the proposed Tabu Search, the following definition must be provided:

Ideal Point-Ideal point is a virtual point that its coordinates are obtained by separately optimizing each objective function.

Finding the ideal point requires separately optimizing each of the objective functions of the problem. However, as the problem in question is non-linear, even optimizing it considering only one objective is a time-demanding task. To overcome this obstacle, we first linearize the problem so that each of the objective functions can be possibly solved to optimality with available optimization software such as LINGO 8. Another problem, even after linearization, is the NP-hardness of the large-size linearized problems due to their large feasible space and our inability to find the global optimum (even a strong local optimum) in reasonable time. The following approach is adopted to solve this problem: when finding the exact ideal point is not easy, an approximation of it called the Dynamic Ideal Point (DIP) is used instead. The approximation requires interrupting the optimization software (LINGO 8) after ξ seconds after the first feasible solution is found and report the best solution found up to that time as the respective coordinate of the ideal point. The value of ξ is determined after running various test problems. To improve this approximation and to prevent it from impairing our algorithm, DIP must be updated at the end of each iteration of the proposed HMOSFLA algorithm.

1) ETS Implementation

The desired size of the antibody repertoire, which is shown by *N*, remains constant during the optimization process. To construct N diverse and good antibodies, the proposed elite Tabu Search (ETS) must be done $\alpha \times N$ times where α is an integer greater than or equal to 1. The Tabu Search starts from a predetermined point called the Starting Point which can be set to be the related sequence of any one of the two values obtained for coordinates of the ideal point. Here, the string of objective function 1 is considered as the starting point. Then, the current solution is saved in a virtual list and will be replaced by a desired solution in its neighborhood that meets the acceptance criterion. This process must be continued until the prespecified termination criterion is met. The detailed description of implementation of the proposed tabu search is as follows:

a) Move Description

The proposed move procedure, which is used to generate a neighborhood subset μ, is based on an implementation of what is known in the GA literature as the inversion operator. An example of the inversion operator is presented below:

Before inversion: 2 1 3 | 4 5 6 7 | 9 8
After inversion: 2 1 3 | 7 6 5 4 | 9 8

b) Tabu List

The move mechanism uses the intelligent Tabu Search strategy, whose principle is to avoid returning to the solution recently visited by using an adaptive memory called Tabu List. The proposed tabu list is attributive and made of a list of pairs of integers (i, j), where $i, j \in \{1,..., n\}$. It means that it is forbidden to inverse the subsequence of jobs between the position i and the position j, if the pair (i, j) exists in the tabu list. The size of tabu list, which is shown by ψ, is a predetermined and sufficiently large value. To diversify the search, a long-term memory is deployed and the Tabu Tenure (T_{max}) will be considered infinite. Besides that, the frequency-based memory is used.

c) Search Direction

In order to simultaneously maintain suitable intensification and diversification, we introduce a new function based on Goal Attainment method. This Function can be shown as follows:

$$\zeta = \sum_{i=1}^{k} \frac{| f_i - F_i |}{w_i} \quad (3)$$

Where f_i is the i^{th} objective function value of the solution F_i is the i^{th} coordinate value of the ideal point and w_i is the weight of i^{th} objective function. The motivation to use this metric is that a solution is efficient for a given set of weights w if it minimizes ζ.

The main difference of the proposed function with the existing ones is that it allows working with a set of solutions which is not necessarily convex. This advantage makes the proposed ETS very popular that can be implemented in every optimization problem with every search space pattern. Another advantage is achieved by generating w_i randomly. According to this approach, the proposed ETS can search the solution space in various directions, so the high diversification is maintained.

To explain the acceptance criteria of a new solution, the variable η is defined as follows:

$$\eta = \zeta_B - \zeta_A \quad (4)$$

Where A is the current solution and B is generated from A by a recent move. So the acceptance criteria can be defined in the following way:

1) If $\eta \le 0$ and the move is not found in the tabu list, solution A will be replaced by B.

2) If $\eta \le 0$ but the move is found in the tabu list, the aspiration strategy is used and solution A will be replaced by B.

3) If $\eta > 0$ and the move is not found in the tabu list, solution A will be replaced by B when solution B is not dominated by solution A.

4) If $\eta > 0$ and the move is found in the tabu list, solution A does not change.

d) Stopping Criteria

The proposed tabu search must be done $\alpha \times N$ times. After running the ETS, We have $\alpha \times N$ number of antibodies that are selected among the whole set of visited solutions to be as near to the Pareto front as possible. To construct N initial antibodies, we select the N best solutions among $\alpha \times N$ according to their distances to the ideal point.

E. Adaptive Pareto archive set

In many researches, a Pareto archive set is provided to explicitly maintain a limited number of non-dominated solutions. This approach is incorporated to prevent losing certain portions of the current non-dominated front during the optimization process. This archive is iteratively updated to get closer to correct Pareto-optimal front. When a new non-dominated solution is found, if the archive set is not full, it will enter the archive set. Otherwise it will be ignored. When a new solution enters the archive set, any solution in the archive dominated by this solution will be removed from the archive set.

When the maximum archive size is exceeded, removing a non-dominated solution may destroy the characteristics of the Trade-off front. There exist many different and efficient methods which deal with the updating procedure when the archive size is exceeded. Among them the most widely adopted techniques are: Clustering methods and k-nearest neighbor methods. But most of these algorithms do not preclude the problem of temporary deterioration, and not converge to the Pareto set.

In this study, we propose an adaptive Pareto archive set updating procedure that attempts to prevent losing new non-dominated solutions, found when Pareto archive size has reached its maximum size.

The archive size, which is shown by *Arch_size*, is a prespecified value and must be determined at the beginning of the algorithm. When a new non-dominated solution is found, one of the two following possibilities may occur for updating the Pareto archive set:

1) Number of the solutions in the archive set is less than *Arch_size*, thus this solution joins the archive set.
2) Number of the solutions in the archive set is equal to (or greater than) *Arch_size*, thus the new solution will be added if its distance to the nearest non-dominated solution in the archive is greater-than-or-equal-to the "Duplication Area" of that nearest non-dominated solution in the archive and the size of Pareto archive increases.

Duplication area of a non-dominated solution in the Pareto archive is defined as a bowl of center of the solution and of radius λ. This area is used as a measure of dissimilarity in order to find diverse non-dominated solutions. The distance between the new non-dominated solution and the nearest non-dominated solution in the archive is measured in the Euclidean distance form. To put it another way, if the new non-dominated solution is not

located in the duplication area of its nearest non-dominated solution in the archive, it is considered as a dissimilar solution and added to the Pareto archive set.

The main advantage of this procedure is to save dissimilar non-dominated solutions, without losing any existing non-dominated solutions in the archive. It must be noticed that, the Pareto archive is updated at the end of each iteration of the proposed immune algorithm.

F. Cloning

In clonal selection, only the highest affinity antibodies will be selected to go to the pool. In this paper, antibodies gain membership to the pool to their quality or their diversity. In other words, the pool is a subset of both diverse and high quality antibodies that consists of an approximation to the Pareto-optimal set.

The construction of the pool starts with the selection of all non-repeated non-dominated antibodies from Pareto archive set. If the number of such non-dominated antibodies is smaller than the required pool size, the remaining antibodies are selected among the dominated antibodies. For this purpose, the dominated antibodies are divided into various fronts and the required number of antibodies is selected with the selection mechanism which depicted in Fig. 4.

{**For** 1 to the required number of antibodies*)*
Tournament selection between two dominated antibodies
If candidate 1 is dominated by candidate 2:
  Select candidate 2
If candidate 2 is dominated by candidate 1:
  Select candidate 1
If both candidates are non-dominated:
  Find the minimum hamming distance of each candidate to the non-dominated antibodies in the Pareto archive set.
  Select the candidate with the larger distance
End for}

Fig. 4. The general scheme of clonal selection mechanism

In this study, the hamming distance is used as a measure to diversify the solution space. This measure is the number of positions in two strings of equal length for which the corresponding elements are different. Put another way, it measures the number of substitutions required to change one into the other.

G. Hypermutation

The high affinity antibodies selected in the previous step are submitted to the process of hypermutation. This process consists of two phases that are implemented in a sequential manner.

1) Swapping Mutation

The proposed immune algorithm uses a swapping mutation for each of the clones. In other words, each clone in its related job-to-position representation is subjected to be mutated.

2) Antibodies Combination

The combination method that we implemented is based on linear combination. Each time the combination procedure is to be used, the prespecified number of the mutated clones, (β), are selected randomly and linearly combined together to produce a new antibody. Let β be 3 and x_i, x_j and x_k be the selected antibodies being combined, then the new antibody x_l is obtained with the following line search:

$$x_l = w_1 x_i + w_2 x_j + w_3 x_k \qquad \sum_{i=1}^{3} w_i = 1 \tag{5}$$

It must be noted that the selected clones must be in their continuous representations and w_i, $i = 1,2,3$ are randomly generated.

H. Stopping Criterion

The proposed immune algorithm must be repeated during a prespecified number of times.

IV. Experimental Results

The performance of the proposed multi-objective immune algorithm is compared with a well-known multi-objective genetic algorithm, i.e. SPEA-II [20]. These two algorithms have been coded in the Visual Basic 6 and executed on an AMD Athlon™ XP 64 bit, 3.0 GHz, and Windows XP using 512 MB of RAM.

A. Algorithms' assumptions

The experiments are implemented in two folds: first, for the small-sized problems, the other for the large-sized ones. For both of these experiments, we consider the following assumptions:

- General assumptions: (1) The processing times (P_{ij}) are integers and are generated from a uniform distribution of $U(1,40)$, (2) The due dates (d_i) are uniformly distributed in the interval $[P(1\text{-}T\text{-}R/2),P(1\text{-}T\text{+}R/2)]$ where $P = (n+m-1)\overline{P}$ with $\overline{P}$ the mean total processing time. The values of T and R are set to 0.2 and 0.6 respectively, (3) The jobs' weights (w_i) are uniformly generated in the interval (1,20), (4) Each experiment is repeated 15 times.
- Multi-objective immune algorithm's assumptions: (1) The value of α is set to 10, (2) The pool size is considered to be equal with antibody repertoire, (3) The combination rate is set to 1 and (4) the value of β is fixed to 3.
- SPEA-II's assumptions: (1) The initial population is randomly generated, The binary tournament selection procedure is used, (3) The selection rate is set to 0.8, (4) The order crossover (OX) and inversion (IN) are used as crossover and mutation operators, and (5) The ratio of ox-crossover and inversion is set to 0.8 and 0.4, respectively.

B. Test problems

The test problems with different sizes generated according to Table II.

TABLE II
PROBLEM SETS

Problem	Job (n)	Machine(m)
Small-sized problems:		
1	7	15
2	7	20
3	8	5
4	8	10
5	8	15
6	8	20
7	9	5
8	9	10
9	9	15
10	9	20
Large-sized problems:		
11	200	15
12	200	20
13	300	5
14	300	10
15	300	15
16	300	20
17	500	5
18	500	10
19	500	15
20	500	20

C. Comparison metrics

There are a number of methods available to compare the performance of different algorithms. Schaffer [21] and many other researchers use *the number of Pareto solutions* (NPS) as a quantitative measure of the performance of the algorithms studied. The *Error Ratio* (ER) and the *Generational Distance* (GD) are also used as the performance measure indicators when the Pareto-optimal solutions are known [22]. Moreover, the *Spacing Metric* (SM) is utilized to express the distribution of individuals over the non-dominated region [23]. The *Diversification Metric* (DM) is also used to measure the spread of the solution set [24]. The definitions of the above-mentioned are as follow:

$$ER = \frac{\sum_{i=1}^{n} e_i}{N} \tag{6}$$

where N is the number of found Pareto optimal solutions, and

$$e_i = \begin{cases} 0 & \text{if the solution } i \in \text{Psreto - optimal frontier} \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

$$GD = \frac{\left(\sum_{i=1}^{n} d_i\right)}{N} \tag{8}$$

where d_i is the Euclidean distance between solution i and the closest which belongs to the Pareto-optimal frontier obtained from the total enumeration.

$$SM = \left[\frac{1}{N-1} \times \sum_{i=1}^{n} \left(\overline{d} - d_i\right)^2\right]^{\frac{1}{2}} \tag{9}$$

where $\overline{d}$ is the mean value of all d_i.

$$DM = \sqrt{\sum_{i=1}^{n} \max\left(\left\|x'_i - y'_i\right\|\right)} \tag{10}$$

where $\left\|x'_i - y'_i\right\|$ is the Euclidean distance between of the non-dominated solution x'_i and the non-dominated solution y'_i.

For the small-sized problems, we have used all the above mentioned comparison metrics to have a comprehensive comparisons and to show the quantitative dominance. In the other hand, for the large-sized problems, because of the large size of the problems, it is impossible to find the Pareto optimal solutions using the total enumeration algorithm. Therefore, the comparison metrics which is used in the small-sized problems must be changed. For this purpose, the following comparison metrics are used: (1) *the number of non-dominated solutions* (NPS) that each algorithm can find, (2) the *Quality Metric* (QM) that is simply measured by putting together the non-dominated solutions found by two algorithms, i.e. *A* and *B*, and reporting the ratio of the non-dominated solutions which are discovered by algorithm *A* to the non-dominated solutions which are discovered by algorithm *B*, (3) *Spacing Metric* (SM), and (4) *Diversification Metric* (DM).

D. Parameter setting

For tuning the algorithms, extensive experiments were conducted with different sets of parameters. At the end, the following set was found to be effective in terms of solution quality and diversity level:

1) Small-sized problems

MOIA's tuned parameters: (1) The size of antibody repertoire at each iteration, *N*, is set to 50, (2) The algorithm is terminated after 50 iterations, (3) Since each objective function is linear and the lingo software can obtain the best values of the coordinates of the ideal point immediately, the value of ξ is set to 0, (4) The neighborhood subset size, μ, and the tabu list size, ψ, are respectively set to 3 and 20, in both of the ETS, (5) The maximum Pareto archive size, *Arch_Size*, is fixed to 35.

SPEA-II' tuned parameters: (1) The population size is set to 50, (2) Algorithm is terminated after 50 iterations.

TABLE III
COMPUTATIONAL RESULTS FOR SMALL-SIZED PROBLEMS

Problems	NPS			ER		GD		SM		DM	
	Total Enumeration	MOIA	SPEA.II	MOIA	SPEA.II	MOIA	SPEA.II	MOIA	SPEA.II	MOIA	SPEA.II
1	6	5.73	4.07	0.05	0.26	0.12	16.52	0.12	0.87	8.27	2.03
2	7	6.6	5.4	0.04	0.14	0.09	12.67	0.23	0.4	6.93	2.69
3	7	5.47	4.47	0.27	0.23	0.36	6.77	0.2	0.67	5.8	1.5
4	7	3.73	3.33	0.36	0.43	0.74	20.23	1.79	2.34	6.27	2.81
5	10	7.6	7.27	0.12	0.11	0.38	7.14	0.29	0.64	5.53	3.57
6	4	3.67	2.27	0.02	0.36	0.07	25.6	5.31	5.89	7.27	3.47
7	9	6.2	2	0.53	0.8	0.76	22.71	1.18	1.56	7.47	3.18
8	4	2.67	1.54	0.1	0.15	0.24	6.42	5.14	7.23	7	3.43
9	7	3.67	2	0.21	0.59	0.77	38.42	3.95	4.25	8.07	3.04
10	5	3.13	2.33	0.1	0.34	0.48	27.55	8.38	8.66	9.87	3.23

TABLE IV
COMPUTATIONAL RESULTS FOR LARGE-SIZED PROBLEMS

Problems	NPS		QM		SM		DM	
	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II
11	25.16	21.67	67.1	32.9	7.34	8.32	29.93	21.12
12	23.88	20.56	80.2	19.8	6.29	6.45	33.76	28.55
13	24.15	22.71	74.3	25.7	4.52	7.67	17.17	12.34
14	27.18	24.44	66.4	33.6	5.74	6.46	37.83	32.40
15	25.14	19.93	77.9	22.1	5.70	6.31	27.15	22.46
16	19.31	14.76	65.2	34.8	6.14	6.37	35.09	29.81
17	25.30	23.89	62.4	37.6	6.32	6.65	21.53	17.45
18	31.14	26.66	70.2	29.8	6.39	6.74	31.17	27.57
19	35.38	29.58	71.4	28.6	7.34	8.32	32.93	26.09
20	30.13	27.45	60.4	39.6	4.21	5.24	27.32	24.91

2) Large-sized problems

MOIA's tuned parameters: (1) The size of antibody repertoire at each iteration, *N*, increases to 200, (2) The algorithm is terminated after 500 iterations, (3) The value of ξ is set to 300 minutes, (4) The neighborhood subset size, μ , and the tabu list size, ψ , are respectively fixed to 3 and 40, in both of the ETS, (5) The maximum Pareto archive size, *Arch_Size*, is set to 100.

SPEA-II's tuned parameters: (1) The population size increases to 200, (2) each algorithm is terminated after 500 iterations.

E. Comparative results

In this section, the proposed MOIA is applied to the test problems and its performance is compared with SPEA-II. The average values of the above mentioned comparison metrics for small-sized and large-sized test problems are illustrated in Tables III and IV, respectively.

As shown in Tables III and IV, the proposed MOIA is superior to the SPEA-II in each test problems. In other words, MOIA provides the higher number of diverse locally non-dominated solutions which are closer to the true Pareto-optimal frontier.

As illustrated in Table V, the proposed MOIA consumes more computational time than SPEA-II. Since MOIA, because of the structure of the proposed elitist tabu search and antibody combination method, can search intelligently more regions of the search space, this higher value of computational time is reasonable.

V. CONCLUSION

This paper has presented a new multi-objective immune algorithm (MOIA) for solving a permutation flow shop scheduling problem with respect to the weighted mean completion time and the weighted mean tardiness.

To validate the proposed multi-objective immune algorithm, we designed various test problems and evaluated the performance and the reliability of the proposed MOIA in comparison with a conventional multi-objective genetic algorithm (i.e. SPEA II) to solve the given problems. Some

useful comparison metrics (such as, number of Pareto solutions found by the algorithm, error ratio, generational distance, spacing metric, and diversity metric) were applied to validate the efficiency of the proposed MOIA. The experimental results indicated that the proposed MOIA outperformed the SPEA II and was able to improve the quality of the obtained solutions, especially for the large-sized problems.

TABLE V
THE AVERAGE VALUES OF COMPUTATIONAL TIMES (SEC.)

Problems	MOIA	SPEA II
Small-sized problems:		
1	16	2
2	17	3
3	8	2
4	12	2
5	39	2
6	50	3
7	8	1
8	43	2
9	39	3
10	65	4
Large-sized problems:		
11	342	137
12	283	145
13	381	129
14	380	155
15	470	159
16	483	162
17	378	181
18	483	185
19	451	188
20	483	192

REFERENCES

[1] T. Murata., H. Ishibuchi, and H. Tanaka, "Multi-Objective Genetic Algorithm and its Applications to Flow Shop Scheduling", *Computers and Industrial Engineering*, vol.30, pp. 957-968, 1996.

[2] J.N.D. Gupta and Jr. E.F. Stafford, "Flowshop scheduling research after five decades", *European Journal of Operational Research*, vol. 169, pp. 699–711, 2006.

[3] M. Pinedo, *Scheduling: theory algorithms and systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[4] E. Nowicki and C. Smutnicki, "A fast tabu search algorithm for the permutation flow shop problem", *European Journal of Operational Research*, vol. 91, pp. 160-175, 1996.

[5] S.M.A Suliman., "A two-phase heuristic approach to the permutation flow shop scheduling problem", *International Journal of Production Economics*, vol. 64, pp. 143-152, 2000.

[6] J. Cheng, G. Steiner, and P. Stephenson, "A computational study with a new algorithm for the three-machine permutation flow-shop problem with release times", *European Journal of Operational Research*, vol. 130, pp. 559–575, 2001.

[7] J. Grabowskia and M. Wodecki, "A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion", *Computers and Operations research*, vol. 31, pp. 1891-1909, 2004.

[8] Z.Lian, X.Gu, and B.Jiao, "A similar particle swarm optimization algorithm for permutation flow shop scheduling to minimize makespan", *Applied Mathematics and Computation*, vol. 175, pp. 773-785, 2006.

[9] S.G. Ponnambalam, H. Jagannathan, M. Kataria, and A. Gadicherla, "A TSP-GA multi-objective algorithm for flow shop scheduling", *International Journal of Advanced Manufacturing Technology*, vol. 23, pp. 909-915, 2004.

[10] B. Toktas, M. Azizoglu, and S.K Koksalan., "Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan", *European Journal of Operational Research,* vol. 157, pp. 286-295, 2004.

[11] D. Ravindran, A. Noorul Haq, S.J. Selvakuar, and R. Sivaraman, "Flow shop scheduling with multiple objective of minimizing makespan and total flow time", *International Journal of Advance Manufacturing Technology*, vol. 25, pp. 1007-1012, 2005.

[12] G.L. Ada and G.J.V. Nossal, "The clonal selection theory", *Scientific American,* vol. 257, pp. 50-57, 1987.

[13] M. Zandieh, S.M.T. Fatemi Ghomi, and S.M. Moattar Husseini, "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times", *Applied Mathematics and Computation*, 2006, Article in Press

[14] L.P. Khoo and T.D. Situmdrang, "Solving the assembly configuration problem for modular products using an immune algorithm approach", *International Journal of Production Research*, vol. 41, pp. 3419-3434, 2003.

[15] G.-C. Luh, C.-H. Chueh, and W.-W. Liu, "Moia: multi-objective immune algorithm", *Engineering Optimization*, vol. 35, pp. 143-164, 2003.

[16] C.A. Coello Coello and N.C. Cortes, "Solving Multiobjective Optimization Problems Using an Artificial Immune System", *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163-190, 2005.

[17] O. Engin and A. Doyen, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", *Future Generation Computer Systems*, vol.20, pp. 1083-1095, 2004.

[18] A. Kumar, A. Prakash, R. Shankar, and M.K. Tiwari, "Psycho-Clonal algorithm based approach to solve continuous flow shop scheduling problem", *Expert Systems with Applications*, 2005, Article in Press.

[19] MF. Tasgetiren, M. Sevkli, YC. Liang, and G. Gencyilmaz, "Particle swarm optimization algorithm for single machine total weighted tardiness problem". *In: Proc. of the IEEE congress on evolutionary computation*, Oregon: Portland, 2004, pp. 1412-1419.

[20] Zitzler E., Laumanns M., Thiele L., "SPEA2: Improving the strength Pareto evolutionary algorithm", *in: Giannakoglou K., Tsahalis D., Periaux J., Papailou P., Fogarty T. (Eds.), EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, 2001a, pp.95-100

[21] J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms", *in: J.D. Schaffer (Ed.), Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Hillsdale, New Jersey: Lawrence Erlbaum, 1985, pp. 93-100.

[22] D.V. Veldhuizen and G. Lamont, in: J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G. Lamont (Eds.), Multi-objective Evolutionary Algorithm Test Suites 1999; pp. 351–357.

[23] N. Srinivas and K. Deb, "Multi-objective optimization using non-dominated sorting in genetic algorithms", *Evolutionary Computation*, vol. 2, no. 2, pp. 221–248, 1994.

[24] P.R. McMullen, "JIT sequencing for mixed-model assembly lines with setups using tabu search", *Production Planning & Control*, vol. 9, no. 5, pp. 504–510, 1998.