Iranian Journal of Operations Research
Vol. 3, No. 2, 2012, pp. 66-82

A Multi-objective Immune System for a New Bi-objective Permutation Flowshop Problem with Sequence-dependent Setup Times

R. Tavakkoli-Moghaddam[*,1], H. Amin-Tahmasbi[2]

We present a new mathematical model for a permutation flowshop scheduling problem with sequence-dependent setup times considering minimization of two objectives, namely makespan and weighted mean total earliness/tardiness. Only small-sized problems with up to 20 jobs can be solved by the proposed integer programming approach. Thus, an effective multi-objective immune system (MOIS) is specially proposed to solve the given problem. Finally, the computational results are reported showing that the proposed MOIS is effective in finding solutions of large-sized problems.

Keywords: *Multi-objective immune system, Bi-objective flowshop scheduling, Sequence-dependent setup times, Earliness/tardiness, Make span.*

Manuscript received on 4/10/2009 and accepted for publication after revision on 25/10/2012.

1. Introduction

A standard flowshop scheduling problem can be formally stated as follows. A set (N= {1, 2, …, n} of n jobs is to be processed on m stages sequentially. There is one machine at each stage, in which all machines are continuously available. A job is to be processed on one machine at a time without preemption, and a machine processes no more than one job at a time. The objective is to schedule the jobs minimizing some performance measures such as make span, total completion time, maximum tardiness, total tardiness, weighted tardiness, and weighted sum of earliness and tardiness. The above-mentioned flowshop problem is an extension of the two-machine flowshop problem first considered and solved by Johnson [15] having several simplifying assumptions [9] for its formulation and solution.

---

[*]Corresponding Author.
[1]Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran. E-mail: tavakoli@ut.ac.ir
[2]Young Researchers and Elites Club, Science and Research Branch, Islamic Azad University, Tehran, Iran. E-mail: h.amintahmasbi@gmail.com

Here, we deal with a bi-objective permutation flowshop scheduling problem considering setup times. The objectives are to minimize the sum of the makespan and weighted mean total earliness/tardiness for all the *n* jobs. An effective multi-objective immune system (MOIS) is used to solve the given problem. The rest of this paper is organized as follows. Section 2 gives a literature review. Section 3 gives the problem definition. In Section 4, the proposed MOIS algorithm is given. The experimental results are provided in Section 5. Finally, Section 6 gives our conclusions.

2. Literature Review

2.1. Objective Functions

Some objective functions used in the literature are as follows: Makespan, weighted completion time, maximum tardiness, total weighted tardiness, maximum earliness, total weighted earliness, maximum flow

time, total weighted flow time, number of tardy job, and total earliness/tardiness. Some of these objectives

are related to the best utilization of the system minimizing the work-in-process of the job, and a few of them are concerned with the due date of the delivery. Studies in bi-objective flowshop problems mostly concentrate on the makespan and total flow time. Varadharajan and Rajendran [30] considered a problem of permutation flowshop scheduling with the objectives of minimizing the makespan and total flow time of jobs, and presented a multi-objective simulated-annealing (MOSA) algorithm. Neppalli et al. [18] considered the two-stage bi-criteria flowshop scheduling problem with the objective of minimizing the total flow time to obtain the optimal sequence. In the view of the NP-hardness of the given problem, two genetic algorithms (GA)-based approaches are proposed to solve the problem.

In bi-objective problems, there is a special concern about minimization of the makespan and total completion time. T'kindt et al. [26] developed mathematical programming formulations, a branch-and-bound algorithm, and a heuristic algorithm for solving the two-machine flowshop scheduling problem with the objective of minimizing the total completion time. Framinan and Leisten [8] considered the problem of the total flow time and makespan minimization in a permutation flow shop. They introduced a multi-criteria iterative greedy search algorithm iterating over a multi-criteria constructive heuristic approach to yield a set of Pareto-efficient solutions (a posteriori approach). Two other objectives mostly considered in bi-objective problems are makespan and maximum tardiness. Framinanand Leisten [7] considered the problem of makespan minimization in a permutation flowshop where the maximum tardiness is limited by a given upper bound. They focused on approximate approaches to obtain good heuristic solutions to the given problem. Allahverdi [1] addressed the *m*-machine no-wait flowshop scheduling problem with a weighted sum of the makespan and maximum lateness criteria. This problem reduces to the single criterion of the makespan or maximum lateness when a weight of zero is assigned to the makespan or maximum lateness. They proposed a hybrid simulated annealing and a hybrid genetic algorithm, which can be used for the single criterion of the makespan or maximum lateness, or the bi-criteria problem. They also proposed a dominance relation (DR) and a branch-and-bound algorithm. Extensive computational experiments show that the proposed heuristics perform much better than the best existing heuristics for the make span, and perform very well for the single criterion of the maximum lateness and the bi-criteria problem.

A few papers address the completion time and total tardiness. T'kindt and Billaut [25] considered a flowshop problem that minimizes the maximum completion time and total tardiness. Basseur et al. [3] used an innovative multi-objective genetic memetic algorithm (MGMA). Besides the criterion of the maximum completion time, in multi-objective problems, the most usable function is the criterion of the total completion time. Ravindran et al. [22] dealt with the multi-criteria approach to flowshop scheduling (FSS) problems by considering the makespan and the total flow time. Three heuristic algorithms, namely HAMC1, HAMC2 and HAMC3 have been proposed for the problem. Rahimi-Vahed et al. [21] considered a bi-criteria no-wait flowshop scheduling problem (FSSP) that minimizes the weighted mean completion time and weighted mean tardiness. They proposed a new multi-objective scatter search (MOSS) to find the locally Pareto-optimal frontier of the given problem. Tavakkoli-Moghaddam et al. [28] presented a novel multi-objective model for a no-wait flowshop scheduling problem that minimizes both the weighted mean completion time and weighted mean tardiness. They proposed a new hybrid multi-objective algorithm based on the features of a biological immune system (IS) and bacterial optimization (BO) to find Pareto optimal solutions for the given problem.

Naderi et al. [17] investigated an extended problem of jobshop scheduling with sequence-dependent setup times and machine availability constraints to minimize the total completion time. They proposed a novel meta-heuristic method based on the artificial immune algorithm (AIA). Gupta et al. [10] developed and compared different local search heuristics for a two-stage flowshop problem with the makespan minimization as the primary criterion and the minimization of either the total flow time, total weighted flow time, or total weighted tardiness as the secondary criterion.

A limited number of bi-objective scheduling problems are as follows: Maximum earliness, machines idle time, and the number of tardy jobs. Qian et al. [19] proposed an effective hybrid algorithm based on differential evolution (DE), namely HDE, to solve a multi-objective permutation flowshop scheduling problem (MPFSSP) with limited buffers between consecutive machines. The convergence property of HDE is analyzed by using the theory of finite Markov chains. Finally, simulations and comparisons based on benchmarks demonstrate the effectiveness and efficiency of the proposed HDE.

Make span, total completion time, and total tardiness are mainly used in the literature. Eren and Guner [6] considered a tri-criteria two-machine flowshop scheduling problem with the tardiness criterion. The objective is to minimize the weighted sum of the total completion time, total tardiness, and make span. Yagmahan and Yenisey [31] considered the flowshop scheduling problem with multi-objectives of the make span, total flow time, and total machine idle time. Their proposed algorithm is based on an ACS meta-heuristic method. To verify the performance of the proposed algorithm, computational experiments were conducted on the benchmark problems. The related results showed that the proposed ant colony algorithm performed better than the HC and NEH heuristics for the multi-objective flowshop scheduling problem. Tavakkoli-Moghaddam et al. [29]

proposed an efficient memetic algorithm (MA) combined with a novel local search engine, namely nested variable neighborhood search (NVNS), in order to solve the flexible flow line scheduling problem (FFLB) with processor blocking and without intermediate buffers. The results obtained by the proposed MA were compared with the classical genetic algorithm (CGA). Chen et al. [5] proposed an integrated model based on a hybrid flowshop scheduling problem with precedence constraint, setup times and blocking for a container handling system. They also proposed a tabu search algorithm to solve the problem. Han et al. [11] considered a scheduling problem of two uniform machines with different speeds. They proposed optimal online algorithms to find a sequence and a schedule minimizing the makespan and the machine activation cost. Jiang et al. [14] considered an online scheduling problem of two parallel and identical machines with some different grades of service (GoS) levels. They proposed an optimal algorithm minimizing the make span.

Nowadays, the scheduling issue of incorporating the earliness/tardiness (E/T) measure has been attracting intensive research. This measure is, in fact, compatible with the concept of the just-in-time (JIT) production system [2]. Considering the active research interests in the last few decades, production operation managers have made their efforts not only to reduce all kinds of waste in production processes but also to meet delivery times of goods. However, under such JIT scheduling circumstances, any job completed earlier than its due date can cause an opportunity cost to be charged for its carrying in the inventory till the due date, while any job not completed till its due date can cause unhappiness of the customer. Therefore, it is desired to get all the jobs finished on their assigned due dates as exactly as possible. This has provided motivations for the E/T measure consideration in recent scheduling research to pursue production commitment to due date [24]. Many scheduling research considering such scheduling problems with the E/T measure have been reported in the literature. Most of them have considered only a single machine or parallel discrete processing machines. The early research is traced back to the late 1970s and early 1980s as evidenced by the literature [2].

Radhakrishnan and Ventura [20] studied the parallel machine earliness/tardiness scheduling problem (PETNDDSP) with non-common due dates, sequence-dependent setup times and varying processing times, where the objective is to minimize the sum of the absolute deviations of completion times from their corresponding due dates. Iranpoor et al. [13] presented a general flexible flowshop scheduling problem to minimize the earliness and tardiness penalties. They considered a finite planning horizon including some equal periods. The objective in all the periods, except the last period, was to minimize the total penalties of E/T originated from the less or excess quantity produced as compared to the cummulative undelivered demands. Tavakkoli-Moghaddam et al. [27] presented the optimal single machine scheduling with idle insert that minimizes the sum of maximum earliness and tardiness. Sakuraba et al. [23] addressed the minimization of the mean absolute deviation from a common due date in a two-machine flowshop scheduling problem. They presented heuristics that use an algorithm based on proposed properties to obtain an optimal schedule for a given job sequence. Computational experiments showed that the developed heuristics outperformed the results found in the literature for problems up to 500 jobs. A literature review shows only a few published works on the ET measure considering the task scheduling on multiple non-similar machines with the objective of minimizing the total cost of earliness and tardiness, along with the sequence-dependent setup times needed for the machines.

2.2. Multi-objective Optimization

A multi-objective decision problem is defined as follows: Given an *n*-dimensional decision variable vector $x=\{x_1,\ldots,x_n\}$ in the solution space X, find a vector x^* that minimizes a given set of K objective functions $z(x^*)=\{z_1(x^*),\ldots,z_K(x^*)\}$. The solution space X is generally restricted by a set of constraints such as $g_j(x^*)=b_j$, for $j=1,\ldots,m$, and bounds on the decision variables. In many real-life problems, objectives conflict with each other. Hence, optimizing x with respect to a single objective often results in unacceptable results with respect to other objectives. Therefore, a perfect multi-objective solution that simultaneously optimizes each objective function is almost impossible. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. If all objective functions are for minimization, a feasible solution x is said to dominate another feasible solution y (written $x>y$) if and only if $z_i(x) \le z_i(y)$ for $i=1,\ldots,K$ and $z_j(x) < z_j(y)$, for at least one objective function j. A solution is said to be a Pareto optimal, if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one other objective. The set of all feasible non-dominated solutions in X is referred to as the Pareto optimal set. For a given Pareto optimal set, the corresponding objective function values in the objective space is called the Pareto front. For many problems, the number of the Pareto optimal solutions is enormous, maybe infinite.

In the literature concerning multi-objective scheduling problems, we can distinguish five main approaches as follows [16]:

(a) Hierarchical approach: The objectives considered are ranked in a priority order and optimized in that order.
(b) Utility approach: A utility function or weighting function - often a weighted linear combination of the objectives is used to aggregate the considered objectives in a single one.
(c) Goal programming (or satisfying approach): All the objectives are taken into account as constraints which express some satisfying levels (or goals) and the objective is to find a solution which provides a value as close as possible to the predefined goals corresponding to the objectives. Sometimes, one objective is chosen as the main objective and is optimized under the constraint related to other objectives.
(d) Simultaneous (or Pareto) approach: The aim is to generate - or to approximate in case of a heuristic method - the complete set of efficient solutions.
(e) Interactive approach: At each step of the procedure, the decision maker expresses his preferences with regards to one (or several) proposed solutions so that the method will progressively converge to a satisfying compromise between the considered objectives.

Each approach has its own advantages and drawbacks as discussed in the general literature on multi-objective optimization problems: Some approaches require more parameters or a priori information (cases (a), (b) and (c)), some are more pragmatic (cases (a) and (b), for instance); however, they are unable to generate some efficient solutions. Others are more general or theoretical (case (d)) or more oriented to real case studies (cases (c) and (e)), and the like. Clearly, the approach used depends essentially on the aim of the study and/or the context of the application treated. Independently of the approach used, Hoogeveen [12] and Chen and Bulfin [4] studied the complexity of bi-criteria and multi-criteria single machine problems. They proved that only problems including flow time as the primary criterion, in a hierarchical optimization, and problems minimizing flow time and maximum tardiness can be solved in a polynomial time. All other problems are either shown to be NP-hard or remain open as far as computational complexity is concerned. Obviously, the problems including more than one machine and two criteria are more difficult to solve.

3. Problem Definition

The main goal of the flowshop scheduling problem is to find a sequence for processing all the jobs on machines so that a given objective is optimized. There are a few articles found in the literature explaining the cost of earliness/tardiness along with another objective. Also, there are a few research works considering the sequence-dependent setup times. Here, we consider these objectives to arrive at a model closer to real-life conditions. The notations are:

P_{ij} Processing time of job j on machine i

S_{ijk} Setup time on machine i if job j is sequenced before job k

d_j Due date for job j

C_{ik} Completion time of a job positioned in sequence k on machine i

α Penalty cost of stopping job j

β Tardiness/earliness cost for any job

$$W_{jk} \begin{cases} 1, \text{ if job } j \text{ is assigned to sequence } k \\ 0, \text{ otherwise} \end{cases}$$

$C_j = C_{mj}$ Completion time of job j

$C_{max} = max_j C_j$ Makespan

$V = \sum_j v_j$ Sum of jobs' weights.

The first considered objective is the minimization of the maximum completion time (i.e., make span). This objective is computed by

$$\min Z_1 = \max_{j=1}^{n} C_j \ . \tag{1}$$

The second objective is to minimize the weighted mean total earliness and tardiness of all jobs. To compute the objective function value, we use

$$\min \ Z_2 = \frac{\sum_{j=1}^{n} v_j (E_j + T_j)}{V} \ . \tag{2}$$

A job is to be processed on one machine at a time without preemption and a machine processes no more than one job at a time, that is,

$$\sum_{j=1}^{n} W_{jk} = 1 \quad , \quad \forall k \in \{1, 2, ..., n\} \tag{3}$$

$$\sum_{k=1}^{n} W_{jk} = 1 \quad , \quad \forall j \in \{1, 2, ..., n\} \cdot \tag{4}$$

The completion times for the jobs on the machines are computed by

$$C_{11} = \sum_{j=1}^{n} W_{j1} \times P_{1j} \tag{5}$$

$$C_{1k} = \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} W_{j_1,k} W_{j_2,k-1} (C_{1,j_2} + S_{1,j_2,j_1} + P_{1,j_1}) \quad , \quad \forall k \in \{2,3,...,n\} \ , \ j_1 \neq j_2 \tag{6}$$

$$C_{i1} = \sum_{j=1}^{n} W_{j1} (C_{i-1,j} + P_{ij}) \quad , \quad \forall i \in \{2,3,...,m\} \tag{7}$$

$$C_{ik} = \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} W_{j_1,k} W_{j_2,k-1} (\max\{C_{i,j_2} + S_{i,j_2,j_1}, C_{i-1,j_1}\} + P_{i,j_1}) \ , \ \forall \, i \in (2,3,...,m) \, , \forall \, k \in (2,3,...,n) \tag{8}$$

$$E_j = \max\{d_j - C_{mj}, 0\} \quad , \quad \forall j \in \{1, 2, ..., n\} \tag{9}$$

$$T_j = \max\{C_{mj} - d_j, 0\} \qquad , \qquad \forall j \in \{1, 2, \ldots, n\} \tag{10}$$

$$j_1, j_2 \in \{1, 2, \ldots, n\} \;\; , j_1 \neq j_2 \tag{11}$$

$$W_{jk} \text{ , Binary,} \qquad C_{ik} \geq 0, \forall i, j, k. \tag{12}$$

Note that the earliness and tardiness of job j are given by the constraints (9) and (10).

4. Proposed Multi-objective Immune System

An effective multi-objective immune system (MOIS) is developed here. The appropriate definition about system's components is very important for meta-heuristic algorithms, and it can be effective in finding the answer, or at least it can affect the computing time. Another main aspect of the appropriate definition of the system components is that the definition can facilitate or harden computer programming of the algorithm for solving the problem. After a careful study of the problem's structure, the following components were selected for the algorithm. The attempt is to prevent the need for studding limitations during the problem solving, and to design the components in a way that they contain limitations. The MOIS implementation is described in the following sections.

4.1. Definitions

The algorithm proposed here is based on the clonal selection principle, in which only the highest affinity antibodies proliferate. The criterion distinguishing between antigens and antibodies is Pareto dominance. In immune algorithms (IAs), antigens refer to the objective function that needs to be optimized. The antibodies refer to the candidate solutions to a problem. Usually, initial antibodies are randomly generated on a feasible space. The algorithm proposed here is based on the clonal selection principle, which recognizes that only the highest affinity antibodies will proliferate. The criterion distinguishing between antigens and antibodies is the Pareto dominance. In other words, non-dominated solutions are the antigens and dominated solutions are the antibodies.

Here, we use an adaptive Pareto archive set updating procedure proposed by Tavakkoli-Moghaddam et al. [28], which prevents losing new non-dominated solutions when the Pareto archive has reached its maximum size. When a new non-dominated solution is found, one of the two following possibilities may occur:

- The number of solutions in the archive set is less than Arch_size, and thus the new solution joins the archive set.
- The number of solutions in the archive set is equal to (or greater than) Arch_size, and 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. Consequently, the size of the Pareto archive increases.

Here, each antibody has two different forms for display at the same time, namely (1) job-to-position representation, and (2) continuous representation. Each one of the two display forms is used in special stages of the proposed algorithm. For the mentioned immune system, three operators of mutation, copy, and crossover are considered as used in genetic algorithms.

4.2. Steps of MOIS

The main MOIS steps shown in Fig. 1are described in the following six subsections.

4.2.1. Determination of the Ideal Point

The ideal point for a multi-objective problem is a virtual point whose coordinates are obtained by separately optimizing each objective function with the same Pareto solution. Determining the ideal point requires separately optimizing each of the objective functions of the problem. However, optimizing even a single objective non-linear problem is a difficult task. To overcome this obstacle, the problem is first linearized. Then, each objective function is optimized by an available optimization software package such as Lingo 8. Another difficultly in the process of finding the ideal point, even after linearization, is the NP-hardness of the problem, which prohibits finding the global optimum (even a strong local optimum) in a reasonable time. As such, an approximation of the ideal point is used instead. The approximation involves interrupting the optimization software (e.g., Lingo 8) for ω seconds after the first feasible solution is found. The best solution found in this way is considered as the corresponding coordinate of the ideal point. The value of $\omega$is determined after running various test problems.

4.2.2. Initialization

If the number of available solutions in each repetition is N, then N primary solutions should be produced for each objective function. For this, we should generate an initial set of solutions randomly and search other solutions besides the initial solutions by a mutation operator with $P_m = 0.05$. Since the initial population has a major effect on the produced solutions, this action will be repeated £ times, and N solutions with shortest distances to the ideal point will be selected.

4.2.3. Pareto Archive

In many investigations, 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.

4.2.4. Distance Evaluation of Antibodies

In clonal selection, only the highest affinity antibodies are selected to go into the pool. In our proposed approach, antibodies gain membership to the pool according to their qualities or diversities. In other words, the pool is a subset of both diverse and high quality antibodies consisting of an approximation of the Pareto-optimal set. The construction of the pool starts with the selection of all non-repeated, non-dominated antibodies from the Pareto archive set. If the number of such non-dominated antibodies is smaller than the required pool size, then the remaining antibodies are selected from among the dominated antibodies. For this purpose, the dominated antibodies are divided into various fronts, and the required numbers of antibodies are selected with a selection mechanism shown below.

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. It allows for measuring the number of substitutions required to change one into another.

4.2.5. Antibodies Improvement

X percent of selected antibodies will transfer to the next generation by reproduction, Y percent of the next generation antibodies will be produced from the selected antibodies by mutation, and Z percent will transfer by crossover.

4.2.6. Termination

The proposed immune system is repeated a pre-specified number of times.

5. Experimental Results

The performance of the multi-objective immune system is compared with the best solution obtained by the branch-and-bound (B-B) method and the lower bound (LB) on the integer programming model. These algorithms are coded in Matlab7 and executed on a 4.0 GHz ®Pentium, with a Windows XP operating system using 1 GB of RAM. For optimal solving time limitation, the adjustment of CPU allowable maximum processing time is regarded in the Lingo software package as 18000 seconds (i.e., five hours). The experiments are implemented in two folds: small and large-sized problems.

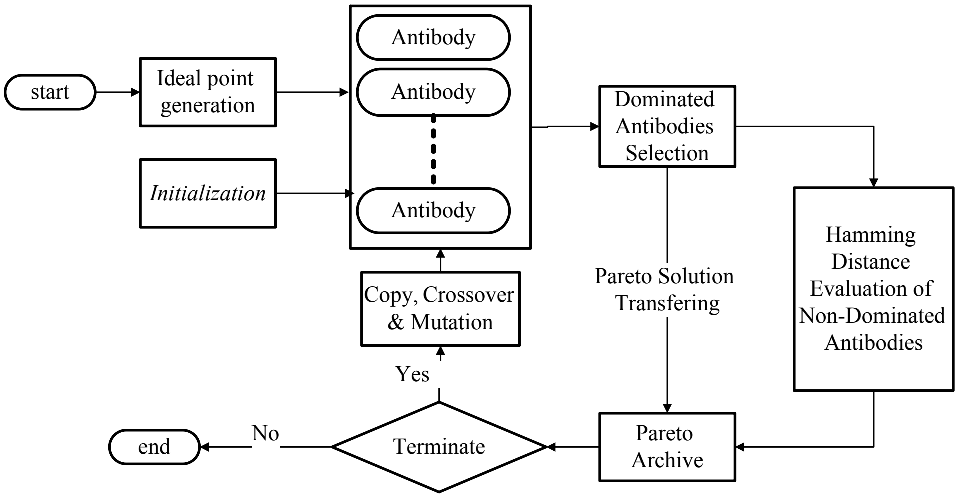

Figure 1. Steps of the proposed MOIS

5.1. Design of Experiments

For the given problems, the processing times on machines are generated from a uniform distribution in the interval [1, 100]. Setup times are randomly generated from uniform distribution on the integers between [1, 20]. Due dates (d_j) are defined to be

$$d_j = \hat{S}^i_{jk} + \sum P_{ij}\,, \tag{13}$$

where $\hat{S}^i_{jk}$ is the average setup time. The experimental set is illustrated in Table 1. Nine and eleven test problems are solved as small and large-sized problems, respectively.

5.2. Comparison Metrics

Results obtained by the algorithms are compared with results of the best solution obtained by the integer model. Considering the disability of the Lingo software package for solving the multi-objective models, we regard the linear composition of multi-lateral objectives on the form of the single objective function. The normalized coefficients are used here for getting the optimal solution of the problems and the study of accuracy and inaccuracy of the presented model. Also, for unification of the range of objective function changes, each one of the functions are normalized by the equation (14) below in order to appropriate the comparability of solutions with the MOIS's outputs:

$$Z_i = \frac{f_i}{\max\{f_i\}}\ , \ i=1, 2, 3, \tag{14}$$

where max f_i is the maximum value of the ith objective function due to the single objective problem, as regarded by the B-B algorithm.

To study the accuracy of the solutions obtained by the presented model, small-sized problems were solved by the B-B algorithm, and in other sample problems, the lower bound (LB) was used for comparison with the proposed MOIS. Also, to show the effectiveness of the presented algorithm on large-sized problems, several sample problems were solved by MOIS and a set of Pareto optimal solutions as the possible solutions along with their lower bounds were obtained by Lingo 8 in five hours. Different weight compositions of objective functions were provided by the decision makers. The maximum Pareto archive size (i.e., Arch_Size) was fixed at 10, the population size for initial solution (₤) was set to 5, the algorithm was terminated after 100 iterations, 'x' was set to 30% , 'y' was set to 62%, 'z' was set to 8%, and 'ω' was fixed at 16000 seconds.

5.3. Results on Small-sized Problems

Table 2 shows the lower bound (LB) and best solutions reported by Lingo 8 in terms of the objective function value (OFV). We solved ten examples in each experiment. Since there were nine experiments for each algorithm, we had 90 examples for each algorithm. To analyze the outcome of the OFV and to prove the algorithm's effectiveness on small-sized problems as well as not having an optimal solution in a reasonable computing time, a statistical analysis of the related results obtained by the MOIS was carried out with the results obtained by the B-B algorithm and the lower bound. Table 2 shows the results in 9 categories, in which each category shows the average relative deviance for 10 samples. Results of 9 categories were obtained by the three methods with regards to coefficients of 0.75 and 0.25 for the linear composition of the objective functions. It is worth noting

Table 1. Experimental set for problems

Parameter	Small sizes		Large sizes	
	Alternative	Values	Alternative	Values
No. of jobs	3	14,10,6	3	200,100
No. of machines	3	20,10,5	2	20,10,5
P_{ij} (Processing time)	1	~U[1,100]	1	~U[1,100]
$\hat{S}^i_{jk}$ (Setup time)	1	~U[1,20]	1	~U[1,20]
D_j (Due date)	1	$d_j = \hat{S}^i_{jk} + \sum P_{ij}$	1	$d_j = \hat{S}^i_{jk} + \sum P_{ij}$
Penalties(α,β)	1	(0.25,0.75)	2	(0.25,0.75) ; (0.5, -0.5)
Total problem	3×3×1×1×1×1=9		2×3×1×1×1×2=12	

that appropriate results cannot be found by Lingo for solving some small-sized problems in the limited five hour period. Thus, they were ignored. Also, the average values for the remaining problems during this limited time are illustrated in the related columns. However, there is no solution for problems with the sizes of 14×10 and 14×20. In this case, we rely on comparing the MOIS's outputs with the obtained lower bounds during the time limit.

The objective function values of 9 categories are compared by MOIS and B-B algorithms as depicted in Fig. 2. In this figure, the proposed MOIS gives fairly good results and is quite effective on the given problems. In addition, this algorithm produces more accurate results than the B-B algorithm almost in all cases.

5.4. Results on Large-sized Problems

We carried out 15 different examples in each experiment. Since we had six categories in each experiment, where there were two different weight compositions [(0.5, 0.5) and (0.3, 0.7)] in each category, 90 examples were solved in general. It is impossible to find the optimal solution in a reasonable time by any exact optimization software package for large-sized problems. In addition, nonlinearity of the proposed model did not create a difficulty for our proposed MOIS, even though in this case the solutions close to the optimal one are acceptable to the designer. The optimization methods can present solutions close to the optimum in a short time. However, on the other hand, the MOIS nature shows its effectiveness better for large-sized problems. Because, if the possible points

Table 2. Computational results for small-sized problems

Size ($n \times m$)	Normalized obj. of LB	Normalized obj. of B-B	Normalized obj. of MOIS	Time of MATLAB	Difference of B-B&LB (%)	Difference of MOIS&LB (%)
6×5	0.293	0.685	0.318	9	133.44	8.32
6×10	0.532	0.732	0.613	11	37.72	15.27
6×20	0.379	0.651	0.492	17	71.68	29.83
10×5	0.437	0.582	0.479	11	33.27	9.75
10×10	0.348	0.569	0.390	37	63.46	12.07
10×20	0.528	0.849	0.638	65	60.80	14.46
14×5	0.363	0.626	0.448	33	72.59	23.52
14×10	0.415	unlimited	0.475	71	unlimited	20.83
14×20	0.571	unlimited	0.673	98	unlimited	17.86

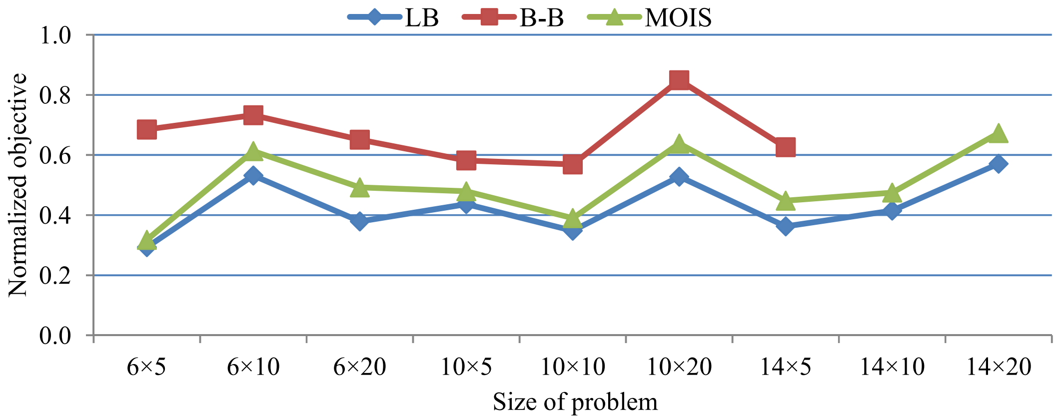

Figure 2. Comparison of the proposed MOIS and B-B algorithms with the LB

for the given problem is increased, then the fast convergence possibility of the generations is decreased and diversification of the population is protected. Therefore, the proposed algorithm has more time for finding solutions close to the optimal ones. It is worth noting that clue to the MOIS nature, the program output is a set of effective results as Pareto solutions, where different compositions can be selected from multi-objective functions as the best solutions on the basis of the decision maker's preferences.

Table 3. Computational results for large-sized problems

Size($n \times m$)	Time (sec.)	Normalized obj. of the proposed MOIS with weights of 0.3 and 0.7			Normalized obj. of the proposed MOIS with weights of 0.5 and 0.5		
		Obj. 1	Obj. 2	Norm. obj.	Obj. 1	Obj. 2	Norm obj.
10×100	310	0.574	0.387	0.443	0.434	0.58	0.507
20×100	348	0.403	0.611	0.549	0.553	0.866	0.710
10×200	335	0.344	0.537	0.479	0.392	0.685	0.539
20×200	386	0.281	0.672	0.555	0.406	0.436	0.421
10×500	407	0.592	0.752	0.704	0.439	0.821	0.630
20×500	522	0.799	0.629	0.680	0.542	0.783	0.663

Six categories of the unsolvable problems were considered to be solved in a time limit by the exact optimization software package, in which each category was the average relative deviation of 15 examples. Table 3 illustrates two kinds of different weight compositions, (0.5, 0.5) and (0.3, 0.7), defined for bi-objective functions, and the best Pareto solutions in a normalized form between 0 and 1. As shown, for large-sized problems, the convergence rate becomes slower with the increase of solution diversifications. Also, in experiments of 20×200 dimensions, the same values are found using both coefficients.

6. Conclusions

We presented a new mathematical model for a multi-objective flowshop scheduling problem with sequence-dependent setup times minimizing the makespan and weighted mean total earliness/tardiness. A multi-objective immune system (MOIS) was proposed for the given problem and the related results were compared with lower bounds and optimal solutions obtained by a conventional optimization algorithm to show the effectiveness of the proposed MOIS. To evaluate the algorithms, a design of experiments was performed. The outcomes of three algorithms in two categories indicated that the proposed MOIS is a powerful and effective tool in exhibiting almost optimal and reasonable possible solutions. Other objectives such as maximum earliness and number of tardy job with sequence-dependent family setup times can also be interesting to be considered in future research. Furthermore, more extensive comparison of the obtained solution by our proposed MOIS with other multi-objective algorithms are welcome.

References

[1] Allahverdi, A. (2004), A new heuristic for m-machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness, *Computers and Operations Research,* 31,157-180.

[2] Baker, K.R. and Scudder, G. (1990), Sequencing with earliness and tardiness penalties: A review, *Operations Research*, 38, 22-36.

[3] Basseur, M., Seynhaeve, F. and Talbi, E.G. (2003), Adaptive mechanisms for multi-objective evolutionary algorithms, In: *Congress on Engineering in System Application CESA,* Lille, France3, pages 72-86.

[4] Chen, C.L. and Bulfin, R.L. (1993), Complexity of a single machine multi-criteria scheduling problems, *European Journal of Operational Research*, 70, 115-125.

[5] Chen, L., Xi, L.F., Cai, J.G., Bostel, N. and Dejax, P. (2006), An integrated approach for modeling and solving the scheduling problem of container handling systems, *Journal of Zhejiang University - Science A*,7(2), 234-239.

[6] Eren, T. and Güner, E. (2008), The tricriteria flowshop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 36, 1210-1220.

[7] Framinana, J.M. and Leisten, R. (2006), A heuristic for scheduling a permutation flowshop with makespan objective subject to maximum tardiness, *International Journal of Production Economics*, 99, 28-40.

[8] Framinan, J.M. and Leisten, R. (2007), A Multi-objective Iterated Greedy search for Flowshop Scheduling with Makespan and Flow time Criteria, Springer-Verlag.

[9] Gupta, J.N.D. (1979), A review of flowshop scheduling research. In: Problems in Manufacturing and Service Organizations, Ritzman LP et al. (eds.), Martinus Nijhoff, The Hague, Netherlands, pages 359-364.

[10] Gupta, J.N.D., Hennig, K. and Werner, F. (2002), Local search heuristics for two-stage flowshop problems with secondary criterion, *Computers and Operations Research*, 29, 123-149.

[11] Han, S.G., Jiang, Y.W. and Hu, J.L. (2007), Online algorithms for scheduling with machine activation cost on two uniform machines, *Journal of Zhejiang University - Science A*, 8(1), 127-133.

[12] Hoogeveen, H. (1992), Single Machine Bicriteria Scheduling, Ph.D. dissertation, University of Eindhoven.

[13] Iranpoor, M., Fatemi Ghomi, S.M.T. and Mohamadinia, A. (2007), Earliness tardiness production planning and scheduling in flexible flowshop systems under finite planning horizon, *Applied Mathematics and Computation*,184, 950-964.

[14] Jiang, Y.W., He, Y. and Tang, C.M. (2006), Optimal online algorithms for scheduling on two identical machines under a grade of service, *Journal of Zhejiang University - Science A*, 7(3), 309-314.

[15] Johnson, S.M. (1954), Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, 1(1), 61-68.

[16] Loukil, T., Teghem, J. and Tuyttens, D.(2005), Solving multi-objective production scheduling problems using metaheuristics, *European Journal of Operational Research*,161, 42-61.

[17] Naderi, B., Khalili, M. and Tavakkoli-Moghaddam, R. (2009), A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time, *Computers and Industrial Engineering*, 56(4), 1494-1501.

[18] Neppalli, V.R., Chen, C.L. and Gupta, J.N.D. (1996), Genetic algorithms for the two-stage bicriteria flowshop problem, *European Journal of Operational Research*, 95(2), 356-373.

[19] Qian, B., Wang, L., Huang, D.X., Wang, W.L. and Wang, X. (2009), An effective hybrid DE-based algorithm for multi-objective flowshop scheduling with limited buffers, *Computers and Operations Research*, 36(1), 209-233.

[20] Radhakrishnan, S. and Ventura, J.A. (2001), Simulated annealing for parallel machine scheduling with earliness/tardiness penalties and sequence-dependent set-up times, *International Journal of Production Research*, 38(10), 2233-2252.

[21] Rahimi-Vahed, A,R., Javadi, B., Rabbani, M. and Tavakkoli-Moghaddam, R. (2008), A multi-objective scatter search for a bi-criteria no-wait flowshop scheduling problem, *Engineering Optimization*, 40(4), 331-346.

[22] Ravindran, D., Noorul Haq, A., Selvakuar, S.J. and Sivaraman, R. (2005), Flowshop scheduling with multiple objective of minimizing makespan and total flow time, *International Journal of Advanced Manufacturing Technology*, 25, 1007-1012.

[23] Sakuraba, C.S., Ronconi, D.P. and Sourd, F. (2009), Scheduling in a two-machine flowshop for the minimization of the mean absolute deviation from a common due date, *Computers and Operations Research*, 36(1), 60-72.

[24] Sung, C.S. and Min, J.I. (2001), Theory and methodology scheduling in a two machine flowshop with batch processing machine(s) for earliness measure under a common due date, *European Journal of Operational Research*, 131, 95-106.

[25] T'kindt, V. and Billaut, J.C. (2002), Multicriteria Scheduling–Theory, Models and Algorithms, Springer-Verlag.

[26] T'kindt, V., Gupta, J.N.D. and Billaut, J.C. (2003), Two-machine flowshop scheduling with a secondary criterion, *Computers and Operations Research*, 30, 505-526.

[27] Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M. and Azaron, A. (2005), Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert, *Applied Mathematics and Computation*, 167(2), 1430-1450.

[28] Tavakkoli-Moghaddam, R., Rahimi-Vahed, A. and Mirzaei, H. (2007), A hybrid multi-objective immune algorithm for a flowshop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness, *Information Sciences*, 177(22), 5072-5090.

[29] Tavakkoli-Moghaddam, R., Safaei, N. and Sassani, F. (2009), A memetic algorithm for the flexible flow line scheduling problem with processor blocking, *Computers and Operations Research*, 36(2), 402-414.

[30] Varadharajan, T.K. and Rajendran, C. (2005), A multi-objective simulated-annealing algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs, *European Journal of Operational Research*, 167, 772-795.

[31] Yagmahan, B. and Yenisey, M.M. (2008), Ant colony optimization for multi-objective flowshop scheduling problem, *Computers and Industrial Engineering*, 54(3), 411-420.