Solving a Multi-Objective No-Wait Flow Shop Problem by a Hybrid Multi-Objective Immune Algorithm

R. Tavakkoli-Moghaddam[1] , A. Rahimi-Vahed[2] and A. Hossein Mirzaei[3]
[1] Department of Industrial Engineering, Faculty of Engineering, University of Tehran,
[2] Department of Informatics and Operations Research, University of Montreal,
[3] Department of Industrial Engineering, Tarbiat Modares University,
[1,3] Iran, [2] Canada

1. Introduction

A frequently occurring operational problem is one of processing a given number of jobs (commodities) on a specified number of machines (facilities) - referred to by various investigators as scheduling, dispatching, sequencing, or combinations thereof (Gupta and Stafford, 2006). In most manufacturing environments, a set of processes is needed to be serially performed in several stages before a job is completed. Such system is referred to as the flow shop environment that is one class of scheduling problems. In a flow shop problem, we consider n different jobs that need to be processed on m machines in the same order. Each job has one operation on each machine and the operation of job i on machine j has processing time p_{ij} (Baker, 1974).

The early groups of flow shop researchers were quite small and these people were concentrated in a few US academic and research institutions. However, today's flow shop research community is global and from every continent and every geographical region (Gupta and Stafford, 2006). Recently, flow shop scheduling problems have been one of the most renowned problems in the area of scheduling and there are numerous papers that have investigated this issue (Murata et al., 1996). For instance, Gupta and Stafford (2006) investigated the evolution of flow shop scheduling problems and possible approaches for their solution over the last fifty years. They introduced the current flow shop problems and the approaches were used to solve them optimally or approximately. Suliman (2000) considered the permutation flow shop scheduling problem by makespan as objective and he proposed a two-phase heuristic approach to solve it. Cheng et al. (2001) addressed the three machine permutation flow shop scheduling problem with release times where the objective is to minimize maximum completion time. They proposed a branch and bound algorithm for solving this problem. Grabowski and Wodecki (2004) proposed a tabu search based algorithm for the permutation flow shop problem with makespan criterion. Solimanpur et al., (2004) proposed a neural networks-based tabu search method for the flow shop scheduling problem, in which the objective is to minimize makespan. Whang et al. (2006) dealt with a two machine flow shop scheduling problem with deteriorating jobs by minimizing the total completion time.

Source: Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner,
ISBN 978-3-902613-02-8, pp.436, December 2007, Itech Education and Publishing, Vienna, Austria

The multi-objective flow shop scheduling problem has been addressed by some of papers. Murata et al. (1996) 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. (2004) 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. (2004) considered the two machine flow shop scheduling problem by minimizing makespan and maximum earliness as objectives. Ravindran et al. (2005) proposed three heuristic algorithms to solve the flow shop scheduling problem which in makespan and total flow time have been considered as objectives.

This chapter deals with a multi-objective no-wait flow shop scheduling problem. The weighted mean completion time and the weighted mean tardiness are to be optimized simultaneously. To tackle this problem, an effective multi-objective immune algorithm (MOIA) is designed for searching locally Pareto-optimal frontier. The rest of this chapter is organized as follows: Basic definitions of multi-objective optimization are presented in Section 2. Section 3 gives the problem definition. In Section 4, the background of immune algorithm is described and then the proposed algorithm is given. The experimental results are provided in Section 5. Finally, In Section 6 we conclude.

2. Multi-Objective Optimization

A single objective optimization algorithm is terminated upon obtaining an optimal solution. However, it is always difficult to find out a single solution for a multi-objective problem. So, it is natural to find out a set of solutions depending on non-dominance criterion. In the following, we provide a summary of some basic definitions in order to better understand the multi-objective optimization problem. Without loss of generality, let us consider a general multi-objective minimization problem with p decision variables and q objectives (q>1) as follows:

$$\text{Minimize } y= f(x) = (f_1(x), f_2(x),..., f_q(x)) \tag{1}$$

where $x \in R^p$, and $y \in R^q$.

Definition 2.1. A solution a is said to dominate solution b if and only if:

$$1) \quad f_i(a) \le f_i(b) \qquad \forall i \in \{1,2,...,q\} \tag{2}$$

$$2) \quad f_i(a) < f_i(b) \qquad \exists i \in \{1,2,...,q\} \tag{3}$$

Solutions which dominate the others but do not dominate themselves are called non-dominated solutions. Local and global optimality are defined as follows:

Definition 2.2. A solution a is locally optimal in the Pareto sense if there exists a real $\varepsilon > 0$ such that there is no solution b which dominates the solution a with $b \in R^p \cap B(a,\varepsilon)$, where $B(a,\varepsilon)$ shows a bowl of center a and radius ε.

Definition 2.3. A solution a is globally optimal in the Pareto sense if there does not exist any vector b such that b dominates the vector a.

The main difference between this definition and the definition of local optimality lies in the fact that we do not have any restriction on the set R^p anymore.

When, we have a globally optimal solution which is not dominated by any other solution in the feasible space, we call it Pareto- optimal. The set of all Pareto-optimal solutions is also termed the Pareto-optimal set or efficient set. Their corresponding images in the objective space are called the Pareto-optimal frontier.

The development of various methodologies, in order to solve multi-objective problems, has been a continuing effort by researchers. There exists various methods for optimizing the multi-objective optimization problems and we have classified them into five sets as follows (Collette and Siarry, 2003):

- Scalar methods,
- Interactive methods,
- Fuzzy methods,
- Methods with use a metaheuristics,
- Decision aid methods,

Among the above mentioned methods, meta-heuristic methods seem to be practically suitable to solve multi-objective optimization problems. Different approaches appear in the literature, for example vector evaluated genetic algorithm (Schaffer, 1985), multi-objective genetic algorithm (MOGA) (Fonseca & Fleming, 1993), niched Pareto genetic algorithm (NPGA) (Horn et al., 1994), non-dominated sorting genetic algorithm (NSGA and NSGA-II) (Deb, 1999; Deb et al., 2002), Pareto Stratum- Niche Cubicle genetic algorithm (PS-NC GA) (Hyun et al., 1998), Multiple Objective Genetic Local Search (MOGLS) (Jaszkiewicz, 1999), strength Pareto evolutionary algorithm (SPEA and SPEA2) (Zitzler et al., 2001a; 2001b), Micro-Genetic Algorithm (Coello Coello & Toscano, 2001), Pareto archive evolution strategy (PAES) (Knowles and Come, 1999), multi-objective tabu search (MOTS) (Pilegaard, 1997), multi-objective scatter search (MOSS) (Beausoleil, 2006).

3. Problem Definition

3.1 No-Wait Flow Shop Scheduling Problem

In this chapter, a no-wait flow shop scheduling problem is considered. The addressed scheduling problem can be described as follows: Consider an n job m machine no-wait flow shop scheduling problem where the machines are ceaselessly ready to be used from time zero onwards. At any time, every job can be processed at most one machine and every machine can process at most one job. Preemption is not permitted; i.e., once an operation is started, it must be completed without interruption. Given the known uninterrupted processing time of job i on machine j, P_{ij}, and due date of job i, d_i and the precedence constraint, the objective is to seek a schedule that minimizes the weighted mean completion time and the weighted mean tardiness of the manufacturing system.

The problem is considered under the following assumptions: (1) All jobs are available at zero time; (2) machines are always available; (3) processing time of each job on each machine is known and constant; (4) setup times and removal times are included in processing times; (5) preemption is not allowed; (6) passing is not allowed; (7) Transportation times are negligible; (8) each job may have its own due date; (9) each machine can process only one job at the same time; (10) a job cannot processed on more than one machine at the same time; and (11) jobs cannot wait between two successive machines and intermediate storage does not exist.

3.2 Objectives Functions

3.2.1 Minimizing the Weighted Mean Completion Time

We consider the no-wait flow shop scheduling problem by minimizing the weighted mean completion time (i.e., $\overline{C}^{(w)} = \frac{1}{W}\sum_i w_i C_i$), where C_i is the completion time for job *i* and w_i is a possible weight related to job *i* and $W = \sum_i w_i$.

3.2.2 Minimizing the Weighted Mean Tardiness

The second objective is to minimize the weighted mean tardiness (i.e., $\overline{T}^{(w)} = \frac{1}{W}\sum_i w_i T_i$), where w_i is a possible weight associated to job *i*, $T_i = \max\{0, C_i - d_i\}$ is the tardiness related to job *i*, and d_i indicates the due date of job *i*.

4. Immune Algorithm

4.1 Artificial Immune Systems in General

The biological immune system is a robust, complex, adaptive system that defends the body from foreign pathogens. It is able to categorize all cells (or molecules) within the body as self-cells or nonself cells. Depending on the type of the pathogen, and the way it gets into the body, the immune system uses different response mechanisms either to neutralize the pathogenic effect or to destroy the infected cells (Aickelin and Dasgupta, 2005). The immune defense mechanism is either nonspecific (innate), which is obtained through evolutions from generation to generation, or specific (acquired), which is learnt through its own encounters with antigens (Khoo and Situmdrang, 2003). 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 (Ada and Nossal, 1987). The immune system mostly consists of the immune cells. The most common type of immune cells is lymphocytes (B-cells and T-cells). Both cells have receptor molecules on their surfaces that they are able to recognize disease causing pathogens (antigens). The B-cell receptor molecule also called as antibody (Engin and Doyen, 2004). 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 (Zandieh et al., 2006). 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 (Khoo and Situmdrang, 2003). This whole process of somatic mutation plus selection is known as affinity maturation (Zandieh et al., 2006).

A novel computational intelligence technique, inspired by immunology, has emerged, known as Artificial Immune Systems (AIs). Several concepts from immunology have been extracted and applied for the solution of real world science and engineering problems (Aickelin and Dasgupta, 2005).

4.2 Previous Work in Artificial Immune System

Recently, the artificial immune system has advocated special attention to itself in order to various applications. For instance, Dasgupta and Forrest (1995) considered tool breakage detection in milling operations by using a negative selection algorithm. Aickelin and Dasgupta (2005) suggested using AIs for the intrusion detection systems and collaborative filtering and clustering in datamining. Other applications of AIs are for solving optimization problems and pattern recognition tasks. De Castro and Von Zuben (2000, 2002) used the clonal selection principle to perform machine learning and pattern recognition tasks and to solve optimization problems. Luh et al. (2003) proposed an immune based algorithm for finding Pareto optimal solutions to multi-objective optimization problems. Coello Coello and Cortes (2005) applied clonal selection principle to solve multi-objective optimization problems. Also artificial immune algorithm has been used to tackle scheduling problems by some papers, Such as, by using the immune algorithm Alisantoso et al. (2003) considered the scheduling of a flexible PCB flow shop. Khoo and Situmdrang (2003) 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 (2004) 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. (2005) used artificial immune system to tackle a continuous flow shop problem with total flow time as criterion. Zandieh et al. (2006) used the immune algorithm for solving the hybrid flow shop scheduling problems where setup times depended on sequence.

4.3 The Proposed Multi-Objective Immune Algorithm (MOIA)

```
{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}
```

Figure 1. The general scheme of MOIA

In this chapter, 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. Fig. 1 presents the pseudo-code of the proposed MOIA.

4.3.1 Antibody Representation

One of the most important decisions in designing a metaheuristic lies in deciding how to represents solutions and relate them in an efficient way to the searching space. Solution representation must have a one-to-one relation with searching space and besides that should be easy to decode to reduce the cost of the algorithm. Two kinds of different antibody representations are used simultaneously in this study, namely 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 next sections we discuss how and when they are used.

4.3.1.1 Job-to-Position Representation

One of the most widely used representations for scheduling problems is job-to-position representation. In this kind of representation, 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. Suppose that the sequence of seven jobs must be determined. Fig. 2 illustrates how this representation is depicted.

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

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

4.3.1.2 Continuous Representation

Tasgetiren et al. (2004) devised a new way of representation for scheduling problems using continuous values. Here, 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 1 are the random numbers obtained.

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

Table 1. A sample set of random numbers

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

Figure 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.

4.3.2 Antibody Initialization

Most 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. On the other hand, even optimizing a single objective non-linear problem is a demanding task. To overcome this obstacle, the problem in hand is first linearized so that each of the objective functions can be solved to optimality with available optimization software such as Lingo 8. Another problem in the process of finding the ideal point, even after linearization, is the NP-hardness of the large size problems due to their large feasible space and our inability to find the global optimum (even a strong local optimum) in a reasonable time. When finding the exact ideal point is not easy, an approximation of the Ideal Point is used instead. The approximation involves interrupting the optimization software (Lingo 8) ξ seconds after the first found feasible solution and report the best found solution as the respective coordinate of the ideal point. The value of ξ is determined after running various test problems.

4.3.2.1 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:

4.3.2.2 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. Inversion is a unary operator that first chooses two random cut points in an antibody. The elements between the cut points are then reversed. 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

4.3.2.3 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 change the job *i* with the job *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 (*Tmax*) will be considered infinite. Besides that, the recency-based memory and frequency-based memory are used.

4.3.2.4 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:

$$\zeta = \sum_{i=1}^{k} \frac{| f_i - F_i |}{w_i} \tag{4}$$

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:

$$\eta = \zeta_B - \zeta_A \tag{5}$$

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 $\eta \leq 0$ 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.

4.3.2.5 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.

4.3.3 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.

4.3.4 Cloning

In clonal selection, only the highest affinity antibodies will be selected to go to the pool. In this study, 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.

```
{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}
```

Figure 4. The general scheme of clonal selection mechanism

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.

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.

4.3.5 Hypermutation

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

4.3.5.1 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.

4.3.5.2 Antibodies Combination

The combination method that we implemented is based on linear combination. Each time the combination procedure is to be used, the pre-specified 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:

$$x_l = w_1 x_i + w_2 x_j + w_3 x_k$$

$$\sum_{i=1}^{3} w_i = 1$$

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

4.3.6 Stopping Criterion

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

5. 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. 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. At first, we present a brief discussion about the implementation of SPEA-II.

5.1 Strength Pareto Evolutionary Algorithm II (SPEA-II)

Zitzler et al., (2001b) proposed a Pareto-based method, the strength Pareto evolutionary algorithm II (SPEA-II), which is an intelligent enhanced version of SPEA. In SPEA-II, each individual in both the main population and elitist non-dominated archive is assigned a strength value, which incorporates both dominance and density information. On the basis of the strength value, the final rank value is determined by the summation of the strengths of the points that dominate the current individual. Meanwhile, a density estimation method is applied to obtain the density value of each individual. The final fitness is the sum of rank and density values. Additionally, a truncation method is used to maintain a constant number of individuals in the Pareto archive.

5.2 Algorithm 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 $\left[P\left(1-T-\frac{R}{2}\right), P\left(1-T+\frac{R}{2}\right)\right]$ 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.

5.3 Small-Sized Problems

5.3.1 Test Problems

The first experiment is carried out on a set of the small-sized problems. This experiment contains 16 test problems of different sizes generated according to Table 2. The proposed multi-objective immune algorithm (MOIA) is applied to the above problems and its performance is compared, based on some comparison metrics, with the above mentioned multi-objective genetic algorithm. The comparison metrics are explained in the next section.

5.3.2 Comparison Metrics

To validate the reliability of the proposed MOIA, five comparison metrics are taken into account.

Problem	Job (*n*)	Machine (*m*)
1	6	5
2	6	10
3	6	15
4	6	20
5	7	5
6	7	10
7	7	15
8	7	20
9	8	5
10	8	10
11	8	15
12	8	20
13	9	5
14	9	10
15	9	15
16	9	20

Table 2. Problem sets for small-sized problems

5.3.2.1 The Number of Pareto Solutions (N.P.S)

This metric shows the number of Pareto optimal solutions that each algorithm can find. The number of found Pareto solutions corresponding to each algorithm is compared with the total Pareto optimal solutions which are obtained by the total enumeration algorithm.

5.3.2.2 Error Ratio (ER)

This metric allows us to measure the non-convergence of the algorithms towards the Pareto-optimal frontier. The definition of the error ratio is the following:

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

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

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

The closer this metric is to 1, the less the solution has converged toward the Pareto-optimal frontier.

5.3.2.3 Generational Distance (GD)

This metric allows us to measure the distance between the Pareto-optimal frontier and the solution set. The definition of this metric as follows:

$$G = \frac{\left(\sum_{i=1}^{n} d_i\right)}{N} \tag{7}$$

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.

5.3.2.4 Spacing Metric (SM)

The spacing metric allows us to measure the uniformity of the spread of the points of the solution set. The definition of this metric is the following:

$$S = \left[\frac{1}{N-1} \times \sum_{i=1}^{n} \left(\bar{d} - d_i\right)^2\right]^{\frac{1}{2}} \tag{8}$$

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

5.3.2.5 Diversification Metric (DM)

This metric measures the spread of the solution set. Its definition is the following:

$$D = \sqrt{\sum_{i=1}^{n} \max\left(\left\| x'_i - y'_i \right\|\right)} \tag{9}$$

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.

5.3.3 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:

Multi-objective immune algorithm'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.

5.3.4 Comparative Results

In this section, the proposed MOIA is applied to the test problems and its performance is compared with SPEA-II. Table 3 represents the average values of the above mentioned comparison metrics.

Problem	NPS		ER		GD		SM		DM	
	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II
1	3	3	0	0	0	0	2.95	3.12	5	0.72
2	4	4	0	0	0	0	4.34	6.42	6.93	1.19
3	4	4	0	0	0	0	5.65	7.08	7.8	1.25
4	3	3	0	0	0	0	5.74	5.89	7.13	1.11
5	3.8	3.6	0.12	0.08	0.16	1.81	1.42	2.53	5.8	1.25
6	5.73	5.6	0.17	0.08	0.38	3.46	0.18	1.04	6.87	1.83
7	5.73	4.07	0.05	0.26	0.12	16.52	0.12	0.87	8.27	2.03
8	6.6	5.4	0.04	0.14	0.09	12.67	0.23	0.4	6.93	2.69
9	5.47	4.47	0.27	0.23	0.36	6.77	0.2	0.67	5.8	1.5
10	3.73	3.33	0.36	0.43	0.74	20.23	1.79	2.34	6.27	2.81
11	7.6	7.27	0.12	0.11	0.38	7.14	0.29	0.64	5.53	3.57
12	3.67	2.27	0.02	0.36	0.07	25.6	5.31	5.89	7.27	3.47
13	6.2	2	0.53	0.8	0.76	22.71	1.18	1.56	7.47	3.18
14	2.67	1.54	0.1	0.15	0.24	6.42	5.14	7.23	7	3.43
15	3.67	2	0.21	0.59	0.77	38.42	3.95	4.25	8.07	3.04
16	3.13	2.33	0.1	0.34	0.48	27.55	8.38	8.66	9.87	3.23

Table 3. Computational results for small-sized problems

As shown in Table 3, the proposed MOIA is superior to the SPEA-II in each test problems. In other words:

1. MOIA could achieve the greater number of Pareto optimal solutions in comparison with SPEA-II.

2. The proposed MOIA has less error ratios in most test problems. This data suggest that the proposed MOIA has higher convergence toward the Pareto-optimal frontier.
3. The proposed immune algorithm can obtain Pareto solutions which are considerably closer to the true Pareto-optimal frontier in comparison with the benchmark algorithm.
4. MOIA provides non-dominated solutions which have less average values of spacing metric. This fact reveals that non-dominated solutions obtained by MOIA are more uniformly distributed in comparison with the other algorithm.
5. The average values of diversification metric in MOIA are considerably more than the other algorithm. In the other word, MOIA could find non-dominated solutions which are more scattered.

Table 4 represents the average of computational times that algorithms consume. As illustrated in Table 4, 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.

Problem	MOIA	SPEA II
1	9	1
2	9	2
3	15	2
4	26	3
5	8	1
6	10	1
7	16	2
8	17	3
9	8	2
10	12	2
11	39	2
12	50	3
13	8	1
14	43	2
15	39	3
16	65	4

Table 4. The average values of computational times (sec.) for small-sized problems

5.4 Large-Sized Problems

5.4.1 Test Problems

Another experiment is implemented for the large-sized problems. To construct the desired test problems, 20 test problems of different sizes generated according to Table 5.

Problem	**Job (*n*)**	**Machine (*m*)**
1	50	5
2	50	10
3	50	15
4	50	20
5	100	5
6	100	10
7	100	15
8	100	20
9	200	5
10	200	10
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

Table 5. Problem sets for large-sized problems

5.4.2 Comparison Metrics

Because of the large size of the test problems, it is impossible to find out 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 (N.P.S) 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) (the definition of the third and fourth metrics is the same as explained in Section "small-sized problems").

5.4.3 Parameter Setting

For tuning this category of problem, extensive experiments were implemented with different sets of parameters too. At the end, the following set was found to be effective in terms of the above mentioned metrics:

5.4.3.1 Multi-objective immune algorithm'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 the ETS, (5) The maximum Pareto archive size, *Arch_Size*, is set to 100.

5.4.3.2 SPEA-II's tuned parameters:

(1) The population size increases to 200, (2) each algorithm is terminated after 500 iterations.

5.4.4 Comparative Results

Table 6 represents the average values of the four above mentioned metrics.

Problem	NPS		QM		SM		DM	
	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II	MOIA	SPEA II
1	17.44	14.27	64.8	35.2	5.16	6.90	17.93	12.22
2	18.25	15.56	69.3	30.7	5.37	5.80	21.14	17.56
3	17.53	14.89	73.4	26.6	5.23	6.14	18.55	14.46
4	19.06	17.34	67.2	32.8	6.25	6.35	25.41	18.88
5	18.42	16.87	60.1	39.9	5.21	5.55	22.56	18.46
6	18.74	16.54	60.8	39.2	5.93	6.81	22.94	20.47
7	20.69	18.67	63.5	36.5	6.32	6.65	31.46	18.72
8	19.55	16.37	73.5	26.5	5.21	5.55	24.43	19.64
9	21.14	17.06	60.9	39.1	6.45	6.85	35.35	26.67
10	22.43	18.32	70.2	29.8	6.45	6.85	31.17	25.46
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	6.64	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

Table 6. Computational results for large-sized problems

As illustrated in table 6, the proposed MOIA shows better performance in all problem sets. In other words, MOIA provides the higher number of diverse locally non-dominated solutions which are closer to the true Pareto-optimal frontier. Computational time increases depending on the number of jobs which must be processed. On the average, MOIA consumes about 2.5 times more than the computational time that SPEA-II spends.

6. Conclusion

This chapter has presented a new multi-objective immune algorithm (MOIA) for solving a no wait 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 optimal solutions founded by 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.

7. References

Ada, G.L. & Nossal, G.J.V. (1987). The clonal selection theory. *Scientific American*, Vol. 257, pp. 50-57.

Aickelin, U. & Dasgupta, D. (2005). Artificial Immune Systems Tutorial, to appear in: *Introductory Tutorials in Optimization, Decision Support and Search Methodology*, Burke, E. & Kendall, G., Kluwer.

Alisantoso, D.L.; Khoo, P.P. & Jiang, Y. (2003). An immune algorithm approach to the scheduling of a flexible PCB flow shop. *International Journal of Advanced Manufacturing Technology*, Vol. 22, pp. 819-827.

Baker, K.R. (1974). Introduction to Sequencing and Scheduling. Wiley, New York.

Beausoleil, R.P. (2006). "MOSS" multiobjective scatter search applied to non-linear multiple criteria optimization. *European Journal of Operational Research*, Vol. 169, 426-449.

Cheng, J.; Steiner, G. & Stephenson, P. (2001). 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.

Coello Coello, C.A. & Cortes, N.C. (2005). Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, Vol. 6, pp. 163-190.

Coello Coello, C.A. & Toscano Pulido, G. (2001). A micro-genetic algorithm for multiobjective optimization, in: *First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Sciences*, Zitzler, E.; Deb, K.; Thiele, L.; Coello Coello, C.A. & Corne, D. (Eds.), No. 1993, pp. 126-140, Springer-Verlag.

Collette, Y. & Siarry, P. (2003). Multiobjective Optimization: Principles and Case Studies. Springer.

Dasgupta D. & Forrest S. (1995). Tool Breakage Detection in Milling Operations using a Negative-Selection Algorithm, *Technical Report No. CS95-5*.

De Castro, L.N. & von Zuben, F.J. (2000). The Clonal Selection Algorithm with Engineering Applications. In:Workshop *Proc. of GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, pp. 36-37.

De Castro, L.N. & von Zuben, F.J. (2002). Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, Vol. 6, 239-251.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, Vol. 7, No. 3, pp. 205-230.

Deb, K.; Pratap, A.; Agarwal, S. & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, Vol. 6, No. 2, pp. 182-197.

Engin, O. & Doyen, A. (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, Vol. 20, 1083-1095.

Fonseca, C.M. & Fleming, P.J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, Forrest, S. (Ed.), pp. 416-423, San Mateo, California, University of Illinois at Urbana-Champaign: Morgan Kaufman Publishers.

Grabowskia, J. & Wodecki, M. (2004). A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. *Computers and Operations research*, Vol. 31, pp. 1891-1909.

Gupta, J.N.D. & Stafford, Jr.E.F. (2006). Flow shop scheduling research after five decades. *European Journal of Operational Research,* Vol. 169, 699-711.

Horn, J.; Nafpliotis, N. & Goldberg, D.E. (1994). A niched Pareto genetic algorithm for multi-objective optimization. *Proceedings of 1st IEEE-ICEC Conference*, pp. 82-87.

Hyun, C.J.; Kim, Y. & Kim, Y.K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers and Operations Research*, Vol. 25, No. 7-8, pp. 675-690.

Jaszkiewicz, A. (1999). Genetic local search for multiple objective combinatorial optimization, *Technical Report RA-014/98*, Institute of Computing Science, Poznan University of Technology.

Khoo, L.P. & Situmdrang, T.D. (2003). Solving the assembly configuration problem for modular products using an immune algorithm approach. *International Journal of Production Research*, Vol. 41, 3419-3434.

Knowles, J.D. & Corne, D.W. (1999). The Pareto archieved evolution strategy: A new baseline algorithm for multiobjective optimization. *Proceedings of Congress on Evolutionary Computation*, pp.98-105, Washington, DC, IEEE Service Center.

Kumar, A.; Prakash, A.; Shankar, R. & Tiwari, M.K. (2005). Psycho-Clonal algorithm based approach to solve continuous flow shop scheduling problem. *Expert Systems with Applications*, Article in Press.

Luh, G.-C.; Chueh, C.-H. & Liu, W.-W. (2003). Moia: multi-objective immune algorithm. *Engineering Optimization*, Vol. 35, 143-164.

Murata, T.; Ishibuchi, H. & Tanaka, H. (1996). Multi-Objective Genetic Algorithm and its Applications to Flow Shop Scheduling. *Computers and Industrial Engineering*, Vol. 30, pp. 957-968.

Pilegaard Hansen, M. (1997). Tabu search in multiobjective optimization: MOTS. *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM_97)*, Cape Town, South Africa.

Ponnambalam, S.G.; Jagannathan, H.;, Kataria, M. & Gadicherla, A. (2004). A TSP-GA multi-objective algorithm for flow shop scheduling. *International Journal of Advanced Manufacturing Technology*, Vol. 23, pp. 909-915.

Ravindran, D.; Noorul Haq, A.; Selvakuar, S.J. & Sivaraman, R. (2005). Flow shop scheduling with multiple objective of minimizing makespan and total flow time. *International Journal of Advance Manufacturing Technology*, Vol. 25, pp. 1007-1012.

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

Solimanpur, M.; Vrat, P. & Shankar, R. (2004). A neuro-tabu search heuristic for flow shop scheduling problem. *Computers and Operations research*, Vol. 31, pp. 2151-2164.

Suliman, S.M.A. (2000). A two-phase heuristic approach to the permutation flow shop scheduling problem. *International Journal of Production Economics*, Vol. 64, pp. 143-152.

Tasgetiren, MF.; Sevkli, M.; Liang, YC. & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for single machine total weighted tardiness problem. *Proceedings of the IEEE congress on evolutionary computation*, pp. 1412-1419, Oregon: Portland.

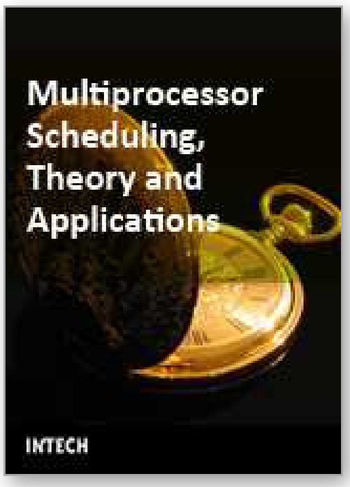
Toktas, B.; Azizoglu, M. & Koksalan, S.K. (2004). Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan. *European Journal of Operational Research*, Vol. 157, pp. 286-295.

Wang, J.-B.; Daniel, Ng C.T.; Cheng, T.C.E. & Li-Li, L. (2006). Minimizing total completion time in a two-machine flow shop with deteriorating jobs. *Applied Mathematics and Computation*, Article in Press.

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

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

Zitzler, E.; Laumanns, M. & Thiele, L.; (2001b). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Technical Report 103*, Computer Engineering and Networks Laboratory (TIK).



Multiprocessor Scheduling, Theory and Applications
Edited by Eugene Levner

ISBN 978-3-902613-02-8
Hard cover, 436 pages
Publisher I-Tech Education and Publishing
Published online 01, December, 2007
Published in print edition December, 2007

A major goal of the book is to continue a good tradition - to bring together reputable researchers from different countries in order to provide a comprehensive coverage of advanced and modern topics in scheduling not yet reflected by other books. The virtual consortium of the authors has been created by using electronic exchanges; it comprises 50 authors from 18 different countries who have submitted 23 contributions to this collective product. In this sense, the volume can be added to a bookshelf with similar collective publications in scheduling, started by Coffman (1976) and successfully continued by Chretienne et al. (1995), Gutin and Punnen (2002), and Leung (2004). This volume contains four major parts that cover the following directions: the state of the art in theory and algorithms for classical and non-standard scheduling problems; new exact optimization algorithms, approximation algorithms with performance guarantees, heuristics and metaheuristics; novel models and approaches to scheduling; and, last but least, several real-life applications and case studies.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821