Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection

Junfei Qiao[a,b,*], Fei Li[a,b], Shengxiang Yang[c], Cuili Yang[a,b], Wenjing Li[a,b], Ke Gu[a,b]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[b] Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China
[c] Center for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK

ARTICLE INFO

Article history:
Received 25 February 2019
Revised 8 August 2019
Accepted 11 August 2019
Available online xxx

Keywords:
Immune algorithm
Uniform distribution selection
Distribution enhancement
Local variation
Multi-objective optimization

ABSTRACT

In general, for the iteration process of an evolutionary algorithm (EA), there exists the problem of uneven distribution of individuals in the target space for both multi-objective and single-objective optimization problems. This uneven distribution significantly degrades the population diversity and convergence speed. This paper proposes an adaptive hybrid evolutionary immune algorithm based on a uniform distribution selection mechanism (AUDHEIA) for solving MOPs efficiently. In AUDHEIA, the individuals in the population are mapped to a hyperplane, which is correlated with the objective space and are clustered to increase the diversity of solutions. To improve the distribution of the solutions, the mapped hyperplane is evenly sectioned. With the constantly changing distribution during the iteration, a threshold as a standard for judging the distribution level is adjusted adaptively. When the threshold is not satisfied in the corresponding interval, the distribution enhancement module is activated. Then, the same number of individuals should be selected in each interval. However, sometimes, there are insufficient or no individuals in the interval during the iterative process. To obtain sufficient individuals, the limit optimization variation strategy of the best individual is adopted. Experiments show that this algorithm can escape from local optima and has a high convergence speed. Moreover, the distribution and convergence of this algorithm are superior to the peer algorithms tested in this paper.

1. Introduction

In engineering and scientific applications, such as industrial production, urban transportation, sewage treatment, and capital operations, there are real multi-objective problems for almost every important decision. These objectives are often incommensurable, or even conflicting. Thus, multi-objective optimization problems (MOPs) have been one of the most important research topics in recent years.

To solve these problems, multi-objective evolutionary algorithms (MOEAs) have been studied extensively. The most famous and advanced MOEAs include the nondominated Sorting Genetic Algorithm II (NSGA-II) [9], Strength Pareto Evolutionary Algorithm (SPEA2) [10], MOEA based on decomposition (MOEA/D) [44], and multi-objective particle swarm optimization (MOPSO) [18,25]. In addition, as nature-inspired algorithms, multi-objective immune algorithms (MOIAs) have been pro-

* Corresponding author at: Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China.
E-mail address: junfeiq@bjut.edu.cn (J. Qiao).

https://doi.org/10.1016/j.ins.2019.08.032
0020-0255/

posed [37,41]. Because they are highly evolutionary, parallel and adaptively distributed, MOIAs have attracted more and more attention. In MOIAs, only a few individuals with better convergence and diversity are selected to propagate, thus producing multiple cloned individuals [7,28,45]. Moreover, each clone evolves better individuals through hypermutation. In this manner, excellent individuals have more evolutionary opportunities. Therefore, MOIAs have a competitive advantage in terms of population diversity and convergence speed compared with other MOEAs [36]. However, most MOIAs use a single hypermutation strategy [14,29,36], which can lead to unsatisfactory results when dealing with complex MOPs [14]. This may be because it is rather difficult to balance the proximity and diversity. This conforms to the no-free-lunch theorem [2]. To tackle this problem, Lin et al. [30] proposed a novel hybrid evolutionary framework for MOIAs (HEIA). This algorithm employs multiple evolutionary strategies to combine their advantages and overcome the shortcomings of any single strategy. However, [36] the purpose of many multi-objective optimization algorithms is to find a certain number of Pareto optimal solutions uniformly distributed along the PF, to better represent the entire PF [2,14,29,46]. Therefore, it is necessary and meaningful to find a solution that can converge better and be distributed more widely and evenly to the PF front for the next generation to increase the diversity of the population. In fact, there still exist two issues to be addressed for all MOEAs during the iteration process, and HEIA is no exception, in this regard: 1) how to select individuals in the population to increase the distribution of individuals during the iteration procedure [44]; in general, the individuals are severely unevenly distributed ., i.e., there may be many individuals residing at some areas of the objective space, while there are few or even no individuals in other areas; and 2) supplement deficiency when there are no sufficient individuals on the Pareto front during the solving process, which is a challenging problem.

For the first question, the objective of multi-objective optimization is to make the solution converge uniformly as widely as possible at the Pareto front [10]. This is because an uneven individual distribution can lead to poor population diversity in the iteration procedure. Therefore, it is easy for the solutions to fall into a local optimum. Some parts of the Pareto front may be empty and the convergence speed decelerates. To overcome these issues, the sharing mechanism proposed by Goldberg and Richardson [13] can be used. This approach to producing new individuals considers the similarity levels among individuals in the population. However, it requires extensive calculation resources. Zhu and Chen [47] adopted the population distribution entropy to portray the diversity and distribution of a population. However, this method lacks the characterization of relations between individuals within groups. Therefore, it is not easy to regulate the diversity and distribution during the evolutionary process. Corne et al. [3,6] presented grid technology in which the individuals with high packing density in the grid are deleted. Nonetheless, the poles cannot be deleted. Knowles and Corne [24] suggested an adaptive grid technique. This algorithm adaptively adjusts the boundary according to the current individual distribution in every evolution. Morse [33] proposed a clustering analysis method to maintain the diversity of the population. Han et al. [16] proposed a method based on the population spacing and population distribution entropy. However, the above methods just delete the individuals with small crowding distances. They cannot resolve the problem of no or insufficient individuals in some regions during the iterations. In addition, MOEA/D can obtain a set of well-distributed solutions by its diversity maintenance among subproblems, which is implemented by a uniform distribution of weight vectors. In MOEA/D, the number of weight vectors is fixed, and only one individual is chosen in each weight vector interval. For simple MOPs, MOEA/D exhibits good performance. However, the results are not satisfactory for complex MOPs [21].

Based on the above discussions, this paper presents an adaptive hybrid evolutionary artificial immune algorithm based on a uniform distribution mechanism (AUDHEIA). In this algorithm, HEIA is used for its ability to solve complex MOPs [30]. However, the problem of the individuals being unevenly distributed during the evolution process significantly affects the population diversity and convergence speed. To solve this problem, the hyperplane is evenly divided, and an equal number of individuals are selected from each interval. However, most Pareto fronts are curves or surfaces. How dimension reduction is performed is very important. For this purpose, motivated by literature [38], the individuals are mapped to a hyperplane that is correlated with the objective space, and they are clustered to increase the diversity of solutions. The results show that this can achieve faster convergence toward the Pareto-optimal front without loss in diversity. However, how to choose individuals from each interval remains a challenging question, and there is no experience to be found in the literature. An indicator is required to measure the distribution of individuals in each interval, and a threshold to judge the distribution standard is also necessary. Because the distribution of individuals changes unceasingly during the iterations, the threshold needs to be adjusted adaptively in this paper. When clustering individuals, if the distribution levels in the corresponding intervals do not meet the threshold, the distribution enhancement module is activated. When the threshold is satisfied, a certain number of superior individuals are selected from each cluster. Then the distribution of individuals can be improved.

However, in the selection process, there are sometimes insufficient individuals or even no individuals in some intervals during the iteration process. Regarding this problem, the individuals with poor diversity values are deleted in most of the literature, and this issue has only rarely been studied. To solve this problem, a limit optimization variation strategy of the best individual is utilized in this paper to produce new antibodies. To verify the performance of AUDHEIA, the inverted generation distance (IGD) function and spacing (SP) function are selected to test diversity and convergence. The experimental results show that our proposed algorithm's inverted generational distance (IGD) and spacing (SP) values are higher than those of other strategies. Therefore, the modified algorithm has better population diversity and distribution, and it converges faster than peer algorithms.

The main contributions of this paper are summarized as follows:

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

1) A distribution measurement indicator is proposed to measure the distribution of individuals in a population. The individuals are mapped to a hyperplane that corresponds to the objective space, and they are clustered on this hyperplane. The hyperplane is equally divided, and the number of segmentations is adjusted adaptively in the iteration process. When the number of species in the interval is less than the threshold, the distribution enhancement module is activated.
2) A distribution enhancement module is introduced to increase the distribution of the individuals. In the population selection process, individuals with the same number are selected in every interval. Furthermore, the choice mechanism is offered.
3) A limit optimization variation strategy of the best individual is taken to generate local solutions. Sometimes the individuals are insufficient or not present at all in the iteration process, so two variation strategies are used to supplement the insufficient individuals. The first strategy can effectively improve the local search ability. The other can prevent the algorithm from falling into local optima, thus improving the search speed.
4) The threshold for judging the distribution level of individuals is adjusted adaptively. During the iterations, the distribution of individuals changes unceasingly. Therefore, this paper designs an adaptive threshold adjustment strategy based on the evolutionary state of the population and the distribution of individuals.

The remainder of this article is organized as follows. In Section 2, MOPs and MOIAs are briefly reviewed. In Section 3, the details of the AUDHEIA framework are described. In Section 4, six experiments are conducted. Finally, some conclusions are presented.

2. Background

2.1. Multi-objective optimization problems

Without loss of generality, an MOP that consists of n decision variables and m objective functions can be described as follows:

$$\begin{aligned} \min F(\mathbf{x}) &= \left(f_1(\mathbf{x}), \cdots, f_j(\mathbf{x}), \cdots, f_m(\mathbf{x})\right)^T \\ \mathbf{x} &= (x_1, \cdots, x_i, \cdots, x_n) \in \Omega \\ l_i &\le x_i \le u_i, \quad i = 1, 2, \cdots, n \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a vector of decision variables, Ω is an n-dimensional decision space, $F(\mathbf{x}) \in \mathbb{R}^m$ is an m-dimensional objective space, $f_j(\mathbf{x})$ ($j = 1, 2, \ldots, m$) is the jth objective function, x_i is the ith decision variable. l_i and u_i are the upper and lower bounds of the ith decision variable, respectively.

Definition 1 (Pareto Dominance)**.** For two given decision vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$ if and only if the formulas (2a)-(2b) hold, and this is recorded as $\mathbf{x} > \mathbf{y}$.

$$\begin{aligned} &(\forall i \in \{1, 2, \cdots, m\} : f_i(\mathbf{x}) \le f_i(\mathbf{y})) \\ &\wedge (\exists k \in \{1, 2, \cdots, m\} : f_k(\mathbf{x}) < f_k(\mathbf{y})) \end{aligned} \tag{2}$$

Definition 2 (Pareto-Optimal)**.** A solution $\mathbf{x}$ is said to be Pareto-optimal if and only if

$$\neg \exists \mathbf{y} \in \Omega : \mathbf{y} > \mathbf{x} \tag{3}$$

Definition 3 (Pareto-Optimal Set)**.** The collection $\mathbf{P_S}$ that is composed of all Pareto-optimal solutions is described as the Pareto-optimal set of solutions.

Definition 4 (Pareto-Optimal Front)**.** The set $\mathbf{P}_F$ includes the values of all the objective functions corresponding to the Pareto-optimal solutions in $\mathbf{P_S}$, that is:

$$\mathbf{P}_F = \left\{ F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots f_m(\mathbf{x}))^T \middle| \mathbf{x} \in \mathbf{P}_s \right\} \tag{4}$$

In this paper, $\mathrm{PF}_{\mathrm{true}}$ is used to refer to the true (or optimal) $\mathbf{P}_F$ as defined in (4), while $\mathrm{PF}_{\mathrm{known}}$ is employed to represent the best solutions produced by an algorithm.

2.2. Immunology terms in MOIAs

An artificial immune system is mainly based on the information processing mechanism of a biological immune system [48]. It is developed to solve complex problems. To describe the algorithm better, several common immunological terms for an artificial immune system are described as follows:

Definition 5 (Antigen)**.** An antigen refers to the problem and constraints to be solved. It is defined as the objective function(s) in an MOP.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

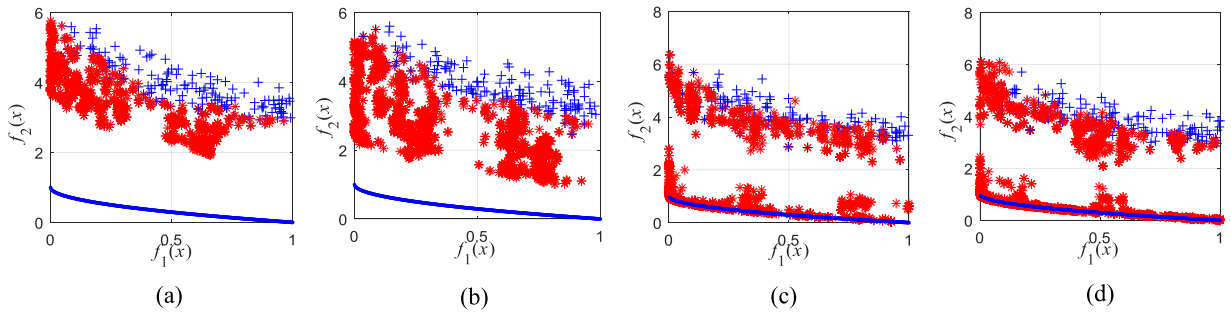

Fig. 1. Nondominated solution sets of objective function $f_1(x)$ and $f_2(x)$ found on the ZDT1 problem. (a) from the 1st to the 10th iteration–HEIA. (b) from the 1st to the 20th iteration–HEIA. (c) from the 1st to the 10th iteration–AUDHEIA. (d) from the 1st to the 20th iteration–AUDHEIA.

Definition 6 (Antibody)**.** An antibody refers to a candidate solution of the problem. It is described as a candidate solution in the decision space for an MOP.

Definition 7 (Affinity)**.** An affinity refers to the adaptive measure of candidate solutions or problems that correspond to the value(s) of the objective function(s).

Yoo and Hajela [40] introduced the first related work on MOIA, and the concept of antibody and antigen affinity was introduced. To imitate the biological immune system, the clone operator is usually chosen by MOIAs to select and clone antibodies with high affinities. Then, the decision variables are altered by the hypermutation. In the process, the aim is to evolve solutions toward having better and better affinities. Since then, many other MOIAs have been designed, most of which have excellent performance. According to the characteristics of the immune system, MOIAs can be divided into three categories: 1) clonal selection methods; 2) immune network methods; and 3) hybrid methods (i.e., combinations of immune systems and other heuristic methods).

Among them, Coello and Cortés [4] proposed a clonal selection based multi-objective immune algorithm (MISA). In this method, only antibodies with high affinity can multiply and produce multiple clonal antibodies, and the adaptive grid method is used to maintain the diversity of the population. In [5], the performance of MISA is further improved [22]. The immune dominant cloning multi-objective algorithm was introduced. The method uses antibody anti-affinity to reflect the similarity between antibodies. This will guide cloning operations to select the most efficient search area (i.e., the least crowded area). In addition, Hu [17] proposed a new MOIA using a multimodal model. Six affinity assignment methods were used in this method, i.e., cloning, hypermutation and immunosuppression. Immunosuppression refers to the removal of similar antibodies in variable and target spaces.

In addition, an artificial immune system based on vectors [11] has been extended by the artificial immune network algorithm (opt-ainet) to solve MOPs. In this case, two cycles of evolution are carried out. The purpose of internal circulation is to utilize the search space, while the purpose of external circulation is to avoid redundancy caused by similar antibodies. In [12], a new weight-based MOIA was proposed. This method uses a random weighted sum method as a fitness allocation scheme and combines it with a new truncation algorithm to eliminate similar individuals. The results show that the method has low computational complexity and can obtain Pareto-optimal solutions with good distributions.

However, most of the above MOIAs only use simple hypermutation operators to evolve antibodies [12,36,42,43]. The use of simple evolutionary methods in MOIAs may lead to monotonous search patterns, which makes existing MOIAs unable to deal with complex MOPs (for example, the UF test problem [32]). In fact, the hybrid mutation method has been studied in immune algorithms [19,20,27,39], and good results have been achieved. For example, Sindhya et al. [38] introduced a hybrid framework for MOEAs, which uses local search modules to speed up convergence. Tang and Wang [39] proposed a new hybrid MOEA that combines the concepts of individual optimum and global optimum from PSO. In addition, Lin et al. [30] proposed a novel hybrid evolutionary framework for MOIAs (HEIA), which selects multiple high-affinity antibodies from antibodies for cloning and then performs SBX and crossover operations. This hybrid evolutionary strategy can overcome the limitations of using a single strategy and has better results in solving different types of MOP problems. Even though HEIA has advantages in terms of convergence speed and population diversity [30], the individuals in the population exhibit the uneven distribution problem during the iteration process. This problem occurs in many MOPs.

For the ZDT test problems, Fig. 1 demonstrates the individual evolution process at different stages. Fig. 1(a) and (b) present HEIA's nondominant solutions to the ZDT1 problem. Fig. 1(a) is the nondominant solution of the objective functions $f_1(x)$ and $f_2(x)$ in the first to tenth iterations of HEIA. Because of the initial stage of the algorithm, the population is randomly initialized. Therefore, as seen from Fig. 1(a), individuals in the population are evenly distributed in the target space at the beginning of the iteration. However, with the continuous evolution of individuals in the population, if the distribution of individuals in the population is not taken into account when selecting the elite solution, it is likely that some individuals will be concentrated in some regions and that some regions will have very few or no individuals. This will seriously affect the diversity of the population, which may cause a Pareto solution to fall into a local optimum and affect the convergence

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

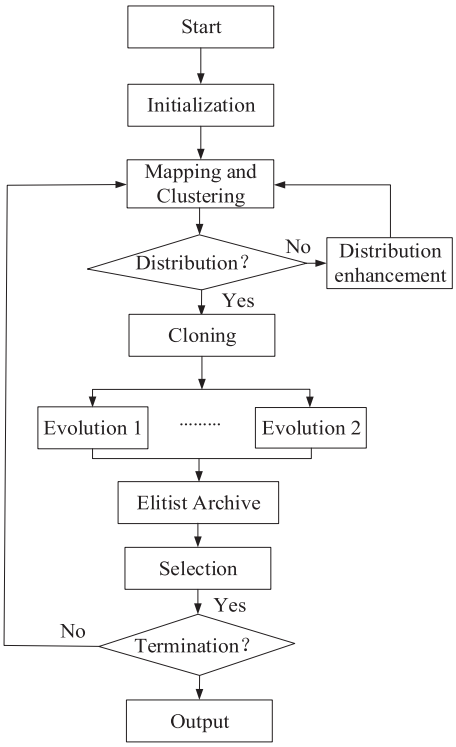

Fig. 2. Proposed framework of AUDHEIA.

of the algorithm. Fig. 1(b) gives the nondominated solutions of the objective functions $f_1(x)$ and $f_2(x)$ of HEIA in the 1st to 20th iterations. It can be seen that in the 10th to 20th iterations, due to the uneven distribution of individuals in the population and the insufficient diversity of the population, the individuals in the population spend a long time searching in the target space with dense individual aggregation. It was not until the 20th iteration that a very few individuals began to look for other regions. In this manner, the convergence rate of Pareto solution is greatly reduced, and the difficulty of extending and evenly distributing Pareto solution to PF front is also increased. This phenomenon exists in almost all of the test problems. Due to the page limitation, they are not introduced here. However, most MOEAs just delete the individuals with smaller crowding distances to increase the diversity of the population. When the Pareto front is continuous, how to solve the empty or insufficient problem in some regions during iterations is a particularly challenging problem. In addition, there are few studies concerning this aspect.

To address this problem, a hybrid evolutionary artificial immune algorithm with adaptive uniform distribution is used to solve the multi-objective optimization problem. At the same time, to compare with HEIA, the nondominant solutions for the ZDT1 problem are also given in Fig. 1(c) and (d). As seen from Fig. 1(c), individuals in the population have been relatively uniformly distributed in the target space. In this manner, the diversity and distribution of the population have been greatly improved. This reduces the possibility of a Pareto solution falling into a local optimum. Fig. 1(d) shows the nondominated solutions of the objective functions $f_1(x)$ and $f_2(x)$ of AUDHEIA in the 1th to 20th iterations. It can be seen that, because the individual distribution in the previous population is more uniform, when very few better individuals find the Pareto front, the rest of the individuals can be more quickly and evenly distributed on the Pareto front. Therefore, in the process of iteration, it is necessary to increase the uniformity of the individual distribution in the population, to improve the diversity of the population.

To summarize, it is necessary to ensure that individuals are distributed uniformly and increase the population diversity when individuals in the population are unevenly distributed during the iteration process.

3. Proposed framework and implementation

The framework of AUDHEIA is shown in Fig. 2. The algorithm starts by initializing the population and setting some related parameters. Then, mapping and clustering operations are performed to increase the diversity of solutions. At the same time, to increase the distribution of solutions, a distribution judgment module is added. When the distribution condition is not satisfied, the distribution enhancement module is activated to obtain uniformly distributed solutions in the iteration process. Afterwards, individuals with higher affinity are cloned and randomly divided into subpopulations of equal size. To avoid the

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 1
Initialization.

```
Set g = 0                                                    //generation number
for i = 1 to N
    for i = 1 to n
      x_i = l_i + rand () ×(u_i - l_i);                      //initialize each variable of x_i
    end for
    Evaluate the objective functions;
end for
Add the nondominated antibodies to the elitist archive E_a;
Calculate the crowding distance for each antibody in E_a;
```

limitation of adopting a single strategy, each subgroup evolves independently with different evolutionary strategies. Finally, the nondominated antibodies of each subgroup are stored in an elitist archive to prepare for the next iteration [30].

According to the AUDHEIA framework shown in Fig. 2, antibodies mainly undergo six important steps: mapping and clustering, distribution judgment, distribution enhancement, cloning, evolution and selection to approach the PF. In addition, Table 1 gives the initial pseudocode, where N is the size of the population and n is the number of decision variables. All nondominated antibodies in the initial population are added to the elite archive E_a to calculate the crowding distance. The other main steps are shown below.

3.1. Mapping and clustering

In each iteration process, to increase the distribution of individuals, the approach described in this paper needs to divide the target space evenly into equal intervals and select the same number of individuals from each subinterval. In the optimization process, the target space is usually a surface or curve. When the target space is evenly partitioned, all coordinate axes need to be segmented, which increases the computational complexity of the algorithm. In the later stage, when the distribution judgment module decides that the individual distribution in a region is insufficient or empty, a certain number of solutions must be added to the region. If the area is a curve or a surface, it will be difficult to determine the exact location. In this module, motivated by literature [38], all individuals in the population are mapped vertically to the hyperplane H. In this manner, only the x-axis (two-dimensional) or x-axis and y-axis (three-dimensional) are segmented evenly, which reduces the computational complexity. H is an (m-1) dimensional linear subspace in m-dimensional Euclidean space, and m is the number of objective functions. The specific expressions are as follows:

$$H = \left\{ \mathbf{F} \in \mathbf{R}^k : \langle \mathbf{W}^s, \mathbf{F} \rangle + b^s = 0 \right\} \tag{5}$$

where b^s is a constant (here $b^s = -1$), $<>$ is the Euclidean inner product, $\mathbf{F}$ is the target vector, and

$$\mathbf{W}^s = \left(\frac{1}{f_1^{\max}}, \cdots, \frac{1}{f_i^{\max}}, \cdots, \frac{1}{f_m^{\max}} \right) \tag{6}$$

where $f_i^{\max}$ is the maximum value of objective function f_i in population P. If $f_i^{\max} = 0$, then $f_i^{\max} = 10^{-06}$. The orthogonal mapping PI of the target vector $\mathbf{F}$ corresponding to the individual in the population on the hyperplane H is as follows:

$$PI = \frac{1 - \langle \mathbf{W}^s, \mathbf{F} \rangle}{\|\mathbf{W}^s\|^2} \mathbf{W}^s + \mathbf{F} \tag{7}$$

From formula (7), we can obtain population P' by mapping all individuals in population P to H. Then the individuals in the mapped population P' are distributed in a straight line or in a plane. At the same time, to increase the diversity of the population, P' is clustered. The corresponding subpopulation of each cluster is P_i, $i = 1, 2, \ldots, K$, where K is the number of clusters. At the same time, the cluster quality index Q is used to evaluate the diversity of P'. The formula is as follows:

$$Q = \sum_{i=1}^{K} \frac{1}{|P_i|} \sum_{C_j \in P_i} E\left(\sigma_i, C_j\right) \tag{8}$$

where $|P_i|$ is the number of individuals of cluster i. σ_i is the centroid of cluster i, C_j is the jth individual of cluster i, and $E(\sigma_i, C_j)$ is the Euclidean distance from the individual C_j to the centroid σ_i of cluster i. Fig. 3 illustrates the method in detail, in which circles 1, 2, 3, 4, 5 and 6 represent different individuals in the population P. Squares a, b and c correspond to the orthogonal mappings of individuals 1, 2-3, and 5-6 on the hyperplane H, respectively. As seen from the graph, although individuals 2, 3, and 4 in P are different, when they map to H, they are represented by the same individual b in population P'. In addition, a, b and c belong to the same cluster of individuals, and when searching locally, they will produce the same Pareto-optimal solution, which reduces the diversity of the population. Therefore, it is very important to maintain a well-distributed population P', and produce different Pareto-optimal solutions in different areas of the Pareto front. To obtain better descriptors for this model, the pseudocode is shown in Table 2.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

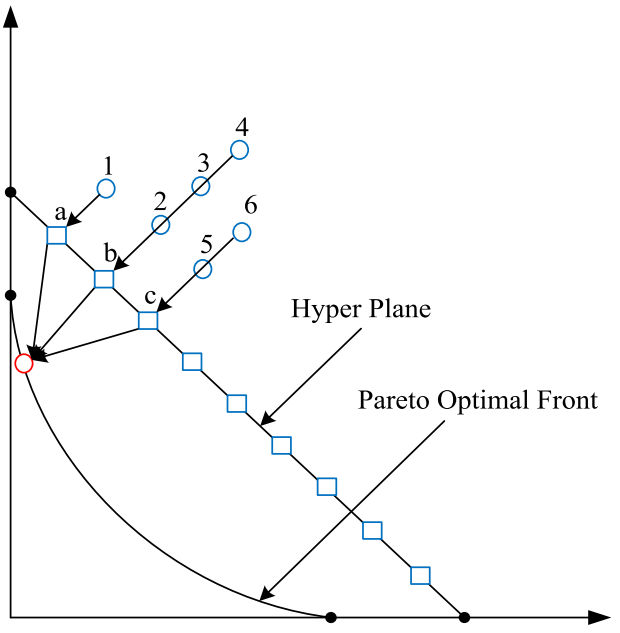

Fig. 3. Mapping of individuals on a Hyperplane.

Table 2
Mapping and cluster module.

Input: size of population, *N*; number of clusters, *K*; and population, *P*.	
Output: Cluster quality index at generation t, *Q*; clusters, P_i^t at generation *t*.	
Step 1. Map the population *P* to the hyperplane *H*, get population *P*′.	%(5)–(7)
Step 2. Cluster the population *P*′ into *K* clusters.	
Step 3. Evaluate the diversity of *P*′.	%(8)
Step 4. Replace the subpopulation in each cluster on the hyperplane *H* with corresponding individuals in *P*.	

3.2. Distribution judgment

To solve the above problems, the target space corresponding to the population P' mapped to the hyperplane H is evenly divided according to the results obtained in section A. Details are shown in Fig. 4. The target space is evenly divided into six intervals (the number of intervals is adjusted adaptively according to the actual range of the target space corresponding to the population P'). We can clearly see that there are five individuals in the first interval (C_{a1}, C_{b1}, C_{b2}, C_{c1} and C_{c2}) and three clusters (C_a, C_b, and C_c are the centers of clusters, respectively). C_{a1} belongs to cluster C_a, C_{b1} and C_{b2}, belong to cluster C_b, and C_{c1} and C_{c2} belong to cluster C_c. The second interval has an individual, a cluster, while the third interval has no individuals. Therefore, individuals in the population are unevenly distributed, and the types of clusters are also very uneven, so it is difficult to produce different Pareto-optimal solutions in different areas of the Pareto front. To increase the distribution of individuals in the population, a certain number of individuals of different types are selected in each interval. The specific description is as follows:

$$D = (D_1, \cdots, D_i, \cdots, D_q) \tag{9}$$

where D is the set of intervals, D_i is the ith interval, $i \in (1, q)$, and q is the number of intervals. $\beta = (\beta_1, \ldots \beta_j, \ldots \beta_m)$ is the set of interval ranges corresponding to the objective function value, and $\beta_j = f_j^{\max} - f_j^{\min}$ is the interval range of the jth objective function value. $f_i^{\max}$ and $f_j^{\min}$ are the maximum and minimum values of the jth objective function in population P', respectively. $\beta_j = \varphi \times 10^{\tau}$, φ is the effective value of the scientific counting method of β_j, and τ is its power exponent; $q = \lceil \varphi$. In interval D_i, the set of clusters is as follows:

$$D_i = \left(P_{i1}, \cdots, P_{ij}, \cdots, P_{ir}\right) \tag{10}$$

In the equation above, P_{ij} is the jth cluster in the ith interval, $0 \le j \le r$, and r is the number of clusters in the interval D_i. The individuals in cluster P_{ij} are as follows:

$$P_{ij} = \left(C_{j1}, \cdots, C_{jo}, \cdots, C_{jl}\right) \tag{11}$$

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

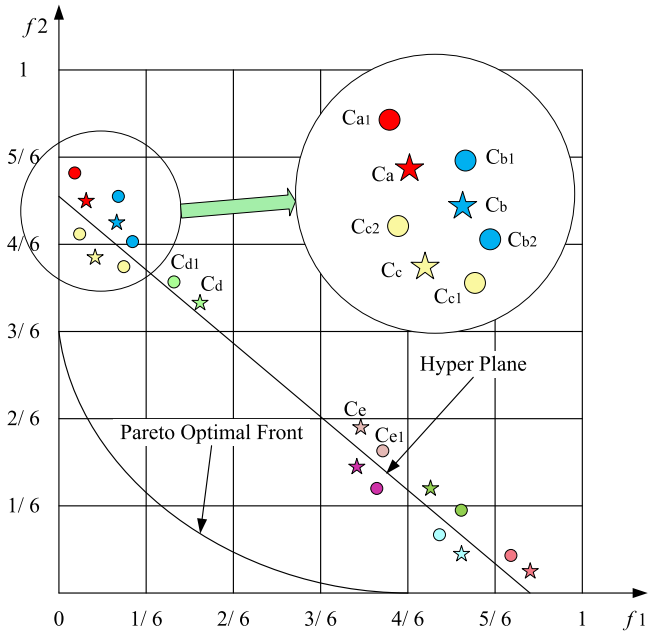

Fig. 4. The distribution of the population.

C_{jo} is the oth individual in the jth cluster P_{ij}, $0 \le o \le l$, and l is the number of individuals in cluster P_{ij}. To judge the distribution of individuals in different interval populations, the mechanism of distribution judgment is as follows:

$$\begin{cases} r_i(t) < \theta(t) & case\ 1 \\ r_i(t) = \theta(t) & case\ 2 \end{cases} \tag{12}$$

where $r_i(t)$ is the number of clusters in the ith interval D_i of the tth iteration. $\theta(t)$ is the threshold of the tth iteration.

*Case*1: The distribution enhancement module is activated.
*Case*2: The individuals of clusters are sorted from large to small in accordance with the crowding distance, and the first $\theta(t)$ individuals of the interval are selected.

Because the distribution of individuals is constantly changing during the iteration process, the value of $\theta(t)$ adaptively changes according to the distribution information *PS* of the population. The specific formula is as follows:

$$PS(t+1) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\bar{M}(t+1) - M_i(t+1)\right)^2} \tag{13}$$

where $PS(t+1)$ is the distribution information of the population in the $(t+1)$th iteration, and $M_i(t+1)$ is the minimum Manhattan distance of the ith antibody and other individuals. $\bar{M}(t+1)$ is the average of the minimum Manhattan distances for all antibodies. Then, the threshold adjustment strategy is as follows:

$$\theta(t+1) = \begin{cases} \theta(t)+1 & PS(t+1) > PS(t) \\ \theta(t)-1 & PS(t+1) < PS(t) \\ \theta(t) & PS(t+1) = PS(t) \end{cases} \tag{14}$$

$$\theta(1) = ceil\left(\frac{N}{K}\right) \tag{15}$$

$$\min\theta(t) = \max r \tag{16}$$

where $\theta(t+1)$ is the threshold of the $(t+1)$th iteration, $\theta(1)$ is the initial threshold, and $\min\theta(t)$ is the minimum of $\theta(t)$. To improve the convergence rate, the quotient of population number and cluster number is rounded as the initial threshold because of the poor distribution of individuals in the early evolution stage. In addition, to ensure that each cluster has individuals to be selected, the maximum number of clusters in all intervals is used as the minimum threshold. As shown in Fig. 4, if the number of individuals of the population in the entire target space is 11 and the number of clusters is 9, then $\theta(1)=1$ (the value of $\theta(1)$ is far greater than $\min\theta(t)$ in the actual iteration process); $\min\theta(t)=3$, and assuming that the current threshold is $\min\theta(t)$, we can see that the number of clusters in the first interval is 3, which satisfies the

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 3
Distribution Judgment.

Input: The range of *X*-axis and *Y*-axis on the hyperplane *H*, Xml and Yml; center points of clustering, σ_i, size of population, *N*, the individuals number of each interval, $N(D_i)$, the population on the hyperplane *H*, *P*′.	
Output: The threshold $\theta(t+1)$, the number of clusters in the *i*th interval, $r_i(t)$; the individuals of the *j*th cluster in the *i*th interval, P_{ij}, the index of cluster centers in the *i*th interval, μ_i.	
Step 1. The hyperplane *H* is evenly divided.	%(9)
Step 2. Calculate the number of clusters in each interval.	%(10)
Step 3. Record the individuals which correspond to every cluster in each interval.	%(11)
Step 4. Calculate the PS information $PS(t+1)$ and threshold $\theta(t+1)$.	%(13)–(16)
Step 5. Judge whether the individuals can meet the distribution in each interval.	%(12)

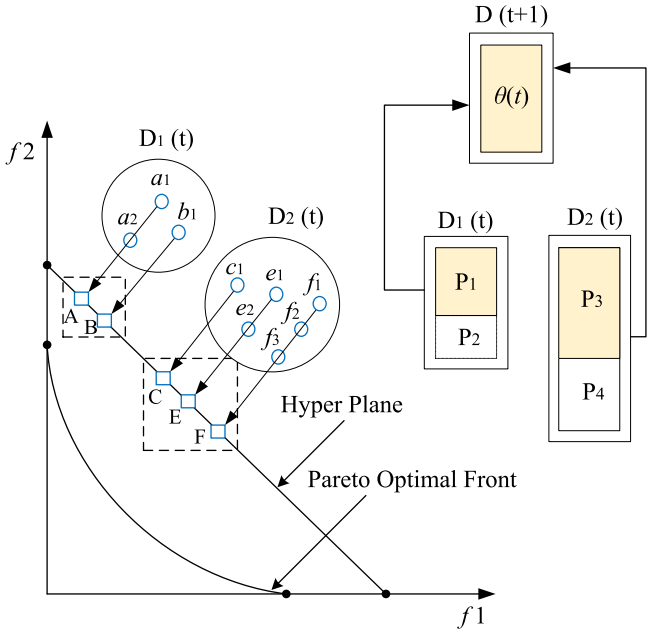

Fig. 5. Diversity enhancement.

distribution; and the number of clusters in the second to sixth interval is less than 3, which does not satisfy the distribution, and the distribution enhancement module will be activated. The details of the distribution judgment module are shown in the pseudocode in Table 3.

3.3. Distribution enhancement

As shown in Fig. 4, some intervals have fewer clusters than others, and some intervals are even empty. How to select individuals is a problem that will need to be solved. Therefore, to maintain a good distribution of individuals in the population, each interval chooses the same number of individuals belonging to different clusters. As shown in Fig. 5, D_1(t) and D_2(t) are two different intervals of the *t*th iteration. There are three individuals in interval D_1(t), of which a1 and a2 belong to cluster *A* and b_1 belongs to cluster *B*. There are six individuals in interval D_2(t). Among them, c_1 belongs to cluster *C*, e_1 and e_2 belong to cluster *E*, and f_1, f_2 and f_3 belong to cluster *F*. At time *t*, $\theta(t)$ individuals are selected for each interval. As shown in the figure, every interval in $D(t+1)$ should contain θ(t) individuals when iterating the $(t+1)$th. However, in the actual optimization process, the number of interval individuals is larger than $\theta(t)$, such as interval $D_2(t)$. In some cases, the number of interval individuals is less than $\theta(t)$, such as interval $D_1(t)$. For interval $D_1(t)$, how to supplement the remaining θ(t)-D_1(t) individuals will be a challenging problem. In this paper, we use the limit optimization mutation strategy to supplement the insufficient individuals in the interval. However, in addition, for the interval D_2(t) in the graph, each cluster selects the top $\lfloor\theta(t)/3\rfloor$individuals according to the crowding distance, and the remaining $\theta(t)-3^*\lfloor\theta(t)/3\rfloor$ individuals are selected from the three clusters randomly. To describe the detailed process, the pseudocode is shown in Table 4. In addition, according to the actual distribution, the methods are as follows:

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 4
Distribution enhancement.

Input: The number of intervals, q; the index of cluster centers in the ith interval, μ_i; The threshold $\theta(t+1)$, the number of clusters in the ith interval, $r_i(t)$; the individuals in the jth clustering in the ith interval, P_{ij}, the number of all individuals in the ith interval, $|D_i|$.
Output: The set of individuals that supplemented by individual variation, **A**; the index of the selected individuals, δ.
for $i=1$: q
  if $r_i(t) < \theta(t)$
    if $r_i(t) == 0$
      The individuals are chosen by (17)–(18)
    else
    if $k \neq 0$
      The individuals are chosen by (21);
    else
      The individuals are chosen by (20);
    end
  end
    else
      Return *case* 2 in formula (17);
end
end

1) The number of cluster centers C_j in the ith interval is zero:

$$\begin{cases} N(P_{ij}) \geq \theta(t) & case\ 1 \\ N(P_{ij}) < \theta(t) & case\ 2 \\ N(P_{ij}) = 0 & case\ 3 \end{cases} \tag{17}$$

where $N(P_{ij})$ is the number of individuals of the jth cluster in the ith interval.

*Case*1: select the first $\theta(t)$ individuals of the interval according to the crowding distance.
*Case*2: select all the individuals of the interval. The remaining $\theta(t)$-$N(P_{ij})$ individuals are supplemented by individuals local variation. Here, the individuals with the largest crowding distance are chosen to mutate.
*Case*3: because there are no individuals in the interval, two individuals that are nearest to this interval are selected from the adjacent interval to mutate, and then $\theta(t)$ individuals can be obtained. However, when the following formulas (18) are satisfied more than 10 times continuously, this interval is considered to be discontinuous.

$$\begin{aligned} & d_i = \min_{j,l} \left\{ \sum_{k=1}^{m} \left| f_k(x_j) - f_k(x_l) \right| \right\} \\ & x_j \in D_{i-1},\ x_l \in D_{i+1},\ i = 1, 2, \cdots, q \\ & d_i \geq length(|D_i|) \end{aligned} \tag{18}$$

where d_i is the distance between individuals in two adjacent intervals. D_{i-1} and D_{i+1} are the $(i-1)$th and $(i+1)$th interval, respectively.

2) The number of cluster centers C_j in the ith interval is nonzero:

When the interval is not empty, individuals are selected from clusters as follows:

$$\begin{aligned} & S_i = floor(\theta_i / r_i) \\ & k = \theta_i - S_i * r_i \\ & R_i = (I_1, \cdots, I_j, \cdots, I_k) \end{aligned} \tag{19}$$

S_i is the number of individuals who should be evenly selected from each cluster. k is the number of remaining individuals who should be selected, I_j is the identifier of the jth remaining individuals, $0 \leq j \leq k$, $k \in$ R, and R_i is the set of I_j. According to different values of k, the methods of individual selection are different. They are shown below:

a) The number of remainders, k, is zero:

$$\begin{cases} N(P_{ij}) \geq S_i & case1 \\ N(P_{ij}) < S_i & case2 \end{cases} \tag{20}$$

*Case*1: select the first S_i individuals of the jth cluster in the ith interval according to the crowding distance.
*Case*2: select all individuals of the jth cluster in the ith interval. The remaining S_i-$N(P_{ij})$ individuals are supplemented by individual variation. Here, the individual with the largest crowding distance is chosen to mutate.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

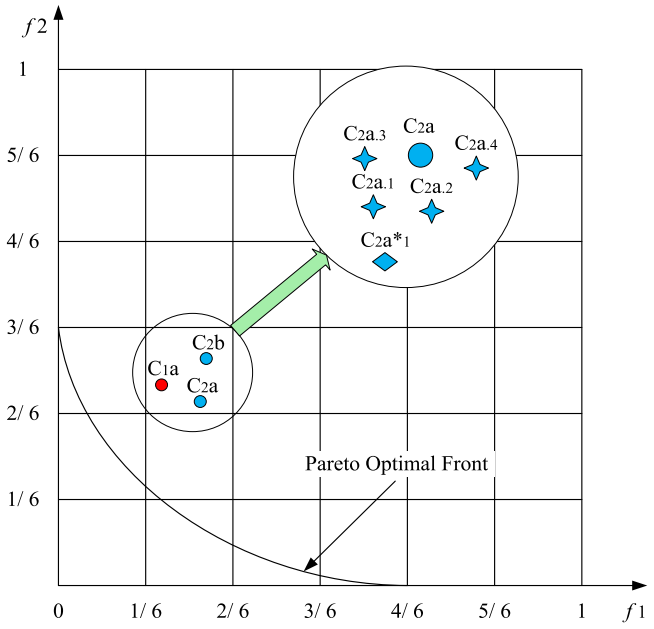

Fig. 6. Limit optimization variation strategy.

b) The number of remainders, k, is nonzero:
When the number of remaining individuals is nonzero, the methods are as defined below:

$$\begin{cases} N(P_{ij}) \geq S_i + k & case1 \\ N(P_{ij}) < S_i + k & case2 \end{cases} \tag{21}$$

*Case*1: select the first S_i individuals of the interval according to the crowding distance. The remaining k individuals are selected from cluster 1 to C_j randomly. If the cluster P_{ij} has been selected η times, then the selection starts from the $(S_i + \eta + 1)$th individual.

*Case*2: select all of the individuals of the jth cluster in the ith interval, when $N(P_{ij}) < S_i$. The remaining S_i-$N(P_{ij})$ individuals are supplemented by individual variation. When $S_i < N(P_{ij}) < S_i + k$, the former S_i individuals are selected from the cluster P_{ij}. In addition, if the individuals from P_{ij} are selected η times and $N(P_{ij}) > S_i + \eta$, then the selection starts from the $(S_i + \eta + 1)$th individual. If $N(P_{ij}) < S_i + \eta$, then reselect randomly.

3.4. Local variation strategy

To solve the problem of insufficient or zero individuals in the interval mentioned in the previous sections, this paper adopts the extreme optimization mutation strategy. When there are insufficient individuals in the interval, we need to find the individuals with the largest congestion distance in the interval. At the same time, when there are zero individuals in the interval, we need to find the two individuals closest to the interval. Then, two optimization mutation strategies are used to mutate the selected individuals. As shown in Fig. 5, the individuals in interval $D_1(t)$ are insufficient, and then the individuals with the largest crowding distance a_2 are selected to mutate. In Fig. 4, the third interval is empty, and the two individuals closest to the interval are selected, namely, C_{d1} and C_{e1}. Then, through two mutation strategies, a_2, C_{d1} and C_{e1} are adopted to generate new individuals.

The detailed mutation strategies are shown in Fig. 6, C_{1a}, C_{2a} and C_{2b} are individuals in the second interval, and C_{2a} is selected as an individual requiring variation according to nondominant ranking and crowding distance. According to the pattern indicated by the arrow, we can see that five new individuals are generated by C_{2a} mutation. $C_{2a.1}$, $C_{2a.2}$, $C_{2a.3}$, and $C_{2a.4}$ are generated by the first mutation strategy and C_{2a^*1} by the second mutation strategy.

1) The first mutation strategy
To effectively improve the local search ability and thereby improve the calculation accuracy, only one decision variable is mutated at a time for the selected object [27]. In addition, according to the distribution enhancement module, assuming that the number of individuals that need to be generated is ψ, the calculation formulas are given as follows:

$$\begin{aligned} C_{jo} &= (x_1, \cdots, x_i, \cdots x_n)0 < i \leq n \\ x'_i &= x_i + \rho \cdot x_{\max}(x_i)0 < i \leq n \end{aligned} \tag{22}$$

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

where x_i is the decision variable that needs to mutate. $x_{max}(x_i)$ is the maximum variable value of x_i. $x_i^{'}$ is the decision variable after mutation.

$$\rho = \begin{cases} (2\alpha)^{(1/(1+\varepsilon))} - 1 \\ 1 - [2(1-\alpha)]^{(1/(1+\varepsilon))} \end{cases}$$
$$x_{\max}(x_i) = \max[x_i - l_i, u_i - x_i] \quad 0 < i \le n \tag{23}$$

where α is a random variable between 0 and 1; $\varepsilon \in R$, and it is the shape parameter; here, ε is set to 11 [46]. This variation operation has strong local adjustment ability because only one decision variable can be mutated at a time. However, only a small range can be searched. To avoid falling into a local optimum, and to improve the search speed, the second mutation strategy is proposed in this paper. $20\% \times \psi$ local solutions are produced by this strategy (rounded up). The method is given as follows:

2) The second mutation strategy

$$C_{j0}^{*} = (x_1', \cdots x_i', \cdots, x_n') \quad 0 < i \le n \tag{24}$$

$$k = 1, 2, \cdots, [0.2\psi]$$
$$x'_i = \lambda x_i \quad 0 < \lambda < 1.2 \tag{25}$$

where $\lambda \in (0, 1.2)$ is a random number. $\psi + [0.2\psi]$ local solutions are generated by the above two strategies. However, when solutions approximate the Pareto front, the system may become unstable when using the second mutation strategy. To avoid this situation, only the first mutation strategy can be used, when $PS(t+1) = PS(t)$ according to formula (14), which is seen as possibly approaching the Pareto front.

3.5. Evolutionary strategies

In the proposed framework, the population is randomly divided into multiple subpopulations after cloning, and multiple evolutionary strategies are adopted for these subpopulations separately. Then, the risks of using a single strategy can be mitigated. In addition, the global search capability and its robustness can be enhanced when solving different types of complicated MOPs [30]. Here, two groups of evolutionary operators are used. The first one is the simulated binary crossover (SBX), which is followed by a polynomial-based mutation [14]. Additionally, DE is a very powerful recombination operator, which is especially suitable for complicated problems [7,28,45].

3.6. Full algorithm of AUDHEIA

The main components of AUDHEIA are introduced through the above sections. They include clustering, mapping, distribution judgment, distribution enhancement and local variation strategy. Other details are described in the pseudocode of Table 5. Here, t and g are the current and the maximum number of generations, respectively. After initialization, the antibodies are randomly generated, and the value of the objective function is calculated. Then, the initial population is produced. Following this, the individual distribution judgment and distribution enhancement strategies are given. The specific methods are introduced from the above sections. After that, the main loop begins. At first, the antibodies with high affinities are copied. Then, they are randomly assigned to two subpopulations, and two evolutions are used by these subpopulations [30], respectively. Later, the antibodies with high fitness values are selected into the elite archive E. If the stopping condition is not met, recalculate 3–13 repeatedly until the maximum number of iterations is reached. At the end, the nondominated solutions in E are used as the final PF_{known}.

4. Experiments

4.1. Test problem

In this paper, various standard test functions are used to test the performance of the AUDHEAI algorithm. As a set of the most widely used test problems, ZDT problems are first selected. However, the characteristics of ZDT are not comprehensive enough, e.g., such as the problems of variable linkages and objective function multimodality cannot be inspected, and thus the ZDT problems cannot completely reflect algorithm performance. To better test the performance of the AUDHEAI algorithm, bi-objective WFG problems [26] are also adopted. These problems have the following characteristics: convexity, concavity, discontinuity, nonuniformity, and the existence of many local PFs [19,46]. Moreover, the three-objective DTLZ problems are used to further test the performance of AUDHEAI. Its performance in handling MOPs with more than two objectives [15] can be tested. In total, 21 test problems (ZDT1-ZDT4, ZDT6, WFG1-WFG9 and DTLZ1–DTLZ7) are used for our experimental studies. For ZDT1–ZDT3 problems, the number of decision variables is 30, while the number of decision variables in ZDT4, ZDT6, and all the WFG and DTLZ problems is 10. In the WFG problems, the ten decision variables consist of eight positional parameters and two distance parameters. The details of the ZDT, WFG and DTLZ test problems are available in [15,19,46,48], respectively.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 5
Complete algorithm of AUDHEIA.

1. **Initialization:** The size of population, N; iterative number, g; the decision variables number, n; mutation parameter, p_m; crossing parameter, p_c; variation range, $x_{\max}$, $x_{\min}$; the size of elitism archive, n_E.
2. Produce initial population;
3. Add the non-dominated antibodies to the elitist archive E_a;
4. mapping and clustering to the hyperplane;
5. A cluster analysis is carried out on antibodies;
6. calculate the crowding distance for each antibody;
7. calculate population distribution;
8. **if** $r_i(t) < \theta(t)$
9. activate the distribution enhancement;
10. **else**
11. select the antibodies to elitism archive E according to the crowding distance;
12. **end**
13. select n_E antibodies with high affinities;
14. **while** $t < g$
15. clone the antibodies according to the crowding distance;
16. evolve the population using SBX and DE; %[30]
17. selection for E_a
18. evaluate convergence and distribution, *IGD and SP*;
19. **if** $\mu > 10^{-4}$
20. calculate 3–13 repeatedly;
21. **else**
22. output E_a
23. **end**
24. $t = t + 1$;
25. **end while**
26. output E_a;

4.2. Performance measures

In this thesis, the following performance measures are adopted to verify the performance of our proposed AUDHEIA.

4.2.1. Inverted generational distance (IGD)

IGD [31] is used to examine the convergence and performance of algorithms simultaneously. The calculation formula is as follows:

$$IGD(P^*_{new}, P_{new}) = \frac{\sum_{x \in P^*_{new}} d(x, P_{new})}{|P^*_{new}|} \tag{26}$$

where $d(x, P_{new})$ is the Euclidean distance between the solution x and the closest solution in the set P_{new} from x. P_{new} is an approximation to the PF achieved by the algorithm. P^*_{new} is a set of known uniformly distributed solutions along the PF in the objective space. A smaller value of IGD indicates better diversity and convergence to the PF.

4.2.2. Spacing (SP)

SP is used to measure the range variance of the neighboring solutions in the known Pareto front. It is defined as:

$$SP = \sqrt{\frac{1}{N_s - 1} \sum_{i=1}^{N_s} (\bar{u} - u_i)^2} \tag{27}$$

$$u_i = \min_{j,l} \left\{ \sum_{k=1}^{m} \left| f_k(x_j) - f_k(x_l) \right| \right\}$$

$$\bar{u} = \sum_{i=1}^{N_s} u_i / N_s \tag{28}$$

where Ns is the number of Pareto-optimal solutions. $\bar{u}$ is the average of ui. $j,l = 1, 2, \ldots, Ns$, and $fk(xj)$ is the kth objective function value of the jth solution. A smaller value of SP indicates more uniform distribution of the obtained Pareto-optimal solutions.

4.3. Experimental settings

In our experiments, to evaluate the performance of AUDHEIA, six types of nature-inspired heuristic algorithms for solving MOPs are subsequently compared. They are NSGA-II [9], SPEA2 [10], AbYSS [35], MOEA/D [28], SMPSO [34] and HEIA [30].

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 6
Parameters settings of all the algorithms compared.

Algorithms	Parameter settings
NSGA-II	$N=100$, $p_c=0.9$, $p_m=1/n$, $\eta_c=20$, $\eta_m=20$
SPEA2	$N=100$, $p_c=0.9$, $p_m=1/n$, $\eta_c=20$, $\eta_m=20$
MOEA/D	$N=100$, $CR=1.0$, $F=0.5$, $p_m=1/n$, $\eta_m=20$, $T=20$, $\delta=0.9$, $n_r=2$
AbYSS	$N=100$, $N_{\text{RefSet1}}=10$, $N_{\text{RefSet2}}=10$, $p_c=0.9$, $p_m=1/n$, $\eta_c=20$, $\eta_m=20$
SMPSO	$C_1\in[1.5, 2.5]$, $C_2\in[1.5, 2.5]$, $p_m=1/n$, $\eta_m=20$
HEIA	$N=100$, $NA=20$, $p_c=1.0$, $p_m=1/n$, $\eta_c=20$, $\eta_m=20$, $CR=1.0$, $F=0.5$, $T=20$, $\delta=0.9$
AUDHEIA	$N=100$, $NA=20$, $p_c=1.0$, $p_m=1/n$, $\eta_c=20$, $\eta_m=20$, $CR=1.0$, $F=0.5$, $T=20$, $\delta=0.9$

Table 7
Performance IGD comparison of results on the ZDT test problems.

Problems		Algorithms						
		NSGA-II[30]	SPEA2[30]	AbYSS[30]	MOEA/D[30]	SMPSO[30]	HEIA[30]	AUDHEIA
ZDT1	Mean	4.95E-03	4.26E-03	3.90E-03	1.75E-02	**3.68E-03**	3.90E-03	3.72E-03
	Std	1.89E-04	1.09E-04	1.16E-04	5.51E-03	**2.61E-05**	6.57E-05	3.68E-05
	Rank	6−	5−	4≈	7−	**1+**	3−	2
ZDT2	Mean	5.06E-03	4.64E-03	4.24E-03	1.33E-02	3.81E-03	3.96E-03	**3.70E-03**
	Std	2.17E-04	1.86E-03	2.15E-03	5.41E-03	3.24E-05	5.23E-05	**2.15E-05**
	Rank	6−	5−	4−	7−	2≈	3−	**1**
ZDT3	Mean	5.68E-03	5.93E-03	1.91E-02	6.04E-02	4.48E-03	4.43E-03	**4.39E-03**
	Std	2.96E-03	5.11E-03	2.38E-02	2.27E-02	2.53E-04	5.41E-05	**2.32E-05**
	Rank	4−	5−	6−	7−	3−	2≈	**1**
ZDT4	Mean	7.28E-03	1.98E-02	1.05E-02	3.12E-01	3.77E-03	3.87E-03	**3.72E-03**
	Std	2.14E-03	2.51E-02	1.73E-02	2.34E-01	4.44E-05	2.00E-04	**3.02E-05**
	Rank	4−	6−	5−	7−	2≈	3−	**1**
ZDT6	Mean	8.29E-03	1.55E-02	3.04E-03	**2.45E-03**	3.03E-03	3.02E-03	2.68E-03
	Std	8.47E-04	2.32E-03	1.13E-04	**7.09E-04**	1.93E-04	1.29E-04	8.56E-05
	Rank	6−	7−	5≈	**1+**	4≈	3−	2−
Rank Sum		25	27	24	28	12	14	7
Final Rank		5	6	4	7	2	3	1
better/worse/similar		0/5/0	0/5/0	0/4/1	1/4/0	1/2/2	0/4/1	/

"+", "−",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

All of the algorithms have exhibited a competitive performance when solving MOPs. Thus, a comparison with the above algorithms can present a holistic assessment of the proposed AUDHEIA algorithm.

Refer to the literature [9,10,14,28,29,34–36] for the setting of the parameters of the compared algorithms. The details are shown in Table 6. It is worth noting that the parameters of the compared algorithms are adjusted to resolve most of the MOPs. To make it fair, the parameters of AUDHEIA are set depending on them. In Table 6, p_m is the mutation probability, p_c is the crossover probability, N is the size of the population, and η_c and η_m are the distribution indexes of SBX and polynomial mutation, respectively. For AbYSS, N_{RefSet1} and N_{RefSet2} are the sizes of RefSet1 and RefSet2, respectively. In MOEA/D, T defines the size of the neighborhood in the weight coefficients, n_r is the maximum number of child solutions that take over from parent solutions and δ dictates the chosen probability that parent solutions are chosen from T neighbors. The two control parameters C_1 and C_2 are randomly set in the interval [1.5, 2.5] in SMPSO.

For the ZDT problems, the settings of N and NA are shown in Table 6, and the maximum number of functional evaluations was set to 25,000. Depending on the difficulty and complexity of the MOP, the size of the population and the maximum number of function evaluations are adjusted appropriately. For more complex WFG and three-objective DTLZ problems, the sizes of the population are set to 200 and 500, respectively. In addition, the maximum numbers of function evaluations are all set to 10^5. The values of N_{RefSet1}, N_{RefSet2}, and NA are set in proportion to the size N of the population. The remaining parameters are given in Table 5. We run each algorithm on each problem instance 30 times. In addition, the mean IGD values and the corresponding standard deviations (std) are selected to reflect the algorithms' performances. The bold values describe the best entries in the table. Moreover, to evaluate the performance of the algorithms using statistics, the Wilcoxon rank sum test is used in our experiments. Here, the significance level is $\rho = 0.05$.

4.4. Results

4.4.1. Comparisons on the ZDT test problems

a) Comparison of the IGD index: The results of all algorithms on the ZDT problems are listed in Table 7. The parameters and results are all derived from [30]. From simulation results, the mean values of NSGA-II, SMPSO, HEIA and AUDHEIA are lower than 10^{-3}. Thus, they are good approximations of PF_{true} for all ZDT problems. For ZDT1, ZDT2 and ZDT3 prob-

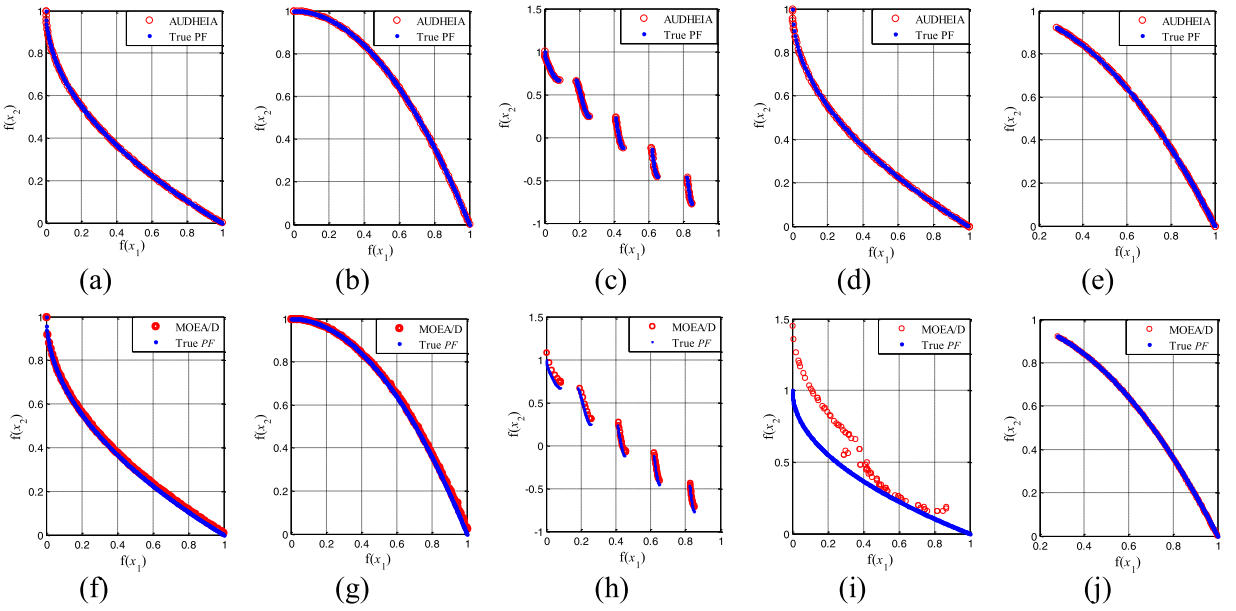

Fig. 7. Nondominated solution sets found by AUDHEIA and MOEA/D on the ZDT problems. (a) ZDT1–AUDHEIA. (b) ZDT2–AUDHEIA. (c) ZDT3–AUDHEIA. (d) ZDT4–AUDHEIA. (e) ZDT6–AUDHEIA. (f) ZDT1–MOEA/D. (g) ZDT2–MOEA/D. (h) ZDT3–MOEA/D. (i) ZDT4–MOEA/D. (j) ZDT6–MOEA/D. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

lems, SPEA2 represents good convergence. Moreover, AbYSS exhibits good performance on ZDT1, ZDT2, and ZDT6. However, MOEA/D performs poorly on ZDT1-ZDT4 problems, although it has the best performance on ZDT6. AbYSS and MOEA/D are unable to approach all of the disconnected parts of PF_{true} in some runs, because ZDT3 is a discrete problem. In addition, ZDT4 has many local PFs, which increases the difficulty of searching for PF_{true}. However, AbYSS, SPEA2 and MOEA/D could not effectively address the ZDT4 problem. For the ZDT6 problem, SPEA2 appeared to perform the worst due to it is uneven search. Under Wilcoxon's rank sum test, in Table 7, we can also see that HEIA has a similar result to AbYSS on ZDT1 and ZDT6, as well as to SMPSO on ZDT3 and ZDT6. The third-to-last line labeled "Rank Sum" expresses the final ranks of all the algorithms. In addition, it can be found that the top two ranks are AUDHEIA and SMPSO from the Final Rank row, while MOEA/D received the worst rank. The last row "better/worse/similar" indicates the number of test problems. They represent better than, worse than, or similar to that of AUDHEIA. From this row, we can see that AUDHEIA is better than any other algorithm.

When the result is close to 10^{-3}, the solution sets can sufficiently approximate the $\mathbf{PF}_{true}$. Therefore, the algorithms are relatively indistinguishable when graphed. The $\mathbf{PF}_{known}$ obtained by AUDHEIA on all of the ZDT problems are illustrated in Fig. 7. To distinguish the algorithms easily when graphed, MOEA/D is chosen as the comparison in Fig. 7. We choose the Pareto-optimal set that has the IGD value that is the closest to the mean IGD value in 30 runs. In Fig. 7, the $\mathbf{PF}_{true}$ is marked in blue, and the $\mathbf{PF}_{known}$ that is achieved by AUDHEIA is labeled in red. We notice that the set of AUDHEIA can distribute uniformly along the $\mathbf{PF}_{true}$ on all of the ZDT problems. However, MOEA/D presents bad performance in some ZDT problems, especially for the ZDT4 problem. In Table 7, the performance rankings of all the algorithms are between AUDHEIA and MOEA/D. Therefore, AUDHEIA exhibits the best performance for the ZDT problems.

To further explore the performance results for different algorithms, the box plots of the IGD results on the ZDT algorithms are presented in Fig. 8. The results demonstrate that the performance of AUDHEIA is superior to that of other algorithms for the ZDT2, ZDT3 and ZDT4 problems. It obtains the minimum mean square deviation and Std values. Although the errors of AUDHEIA are slightly larger than those of SMPSO and MOEA/D for ZDT1 and ZDT6 problems, the performance of AUDHEIA, generally speaking, is the best.

b) Comparison of the SP index: In Table 8, six algorithms are used to compare the SP performance with AUDHEIA. These algorithms are NSGA-II [9], SPEA2 [10], AbYSS [35], MOEA/D [28], SMPSO [34] and HEIA [30]. All of the parameters are derived from the reference papers. We run each algorithm 30 times. The average values of the results are listed in Table 8. One can notice that, the Mean and Std SP values of AUDHEIA are the smallest on the ZDT series test problems compared with the six other algorithms. Furthermore, the SP performance of AUDHEIA is similar to that of AbYSS on ZDT6. Again, notice that the SP performances of HEIA are superior to those of SMPSO on ZDT6. Moreover, the four test problems (ZDT1, ZDT2, and ZDT4) are similar to HEIA. In addition, AbYSS ranks in second place on the ZDT3 and ZDT6 problems. From the Rank Sum, the SP performance of AUDHEIA is obviously better than that of other algorithms. HEIA and ABYSS are the second and third ranks, respectively. However, MOEA/D has the worst result. Beyond that, the SP performances of NSGA-II, SPEA2, and SMPSO are between those of AbYSS and MOEA/D.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 8
Performance SP comparison of results on the ZDT test problems.

Problems		Algorithms						
		NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA	AUDHEIA
ZDT1	Mean	0.3429	0.3215	0.1098	0.3752	0.0750	0.0734	**0.0551**
	Std	0.0976	0.0188	0.0169	0.0590	0.0108	0.0086	**0.0072**
	Rank	6–	5–	4	7–	3≈	2–	**1**
ZDT2	Mean	0.3778	0.3338	0.1382	0.3374	0.0733	0.0717	**0.0557**
	Std	0.0340	0.0329	0.1136	0.0956	0.0134	0.0101	**0.0082**
	Rank	7–	5–	4–	6–	3≈	2–	**1**
ZDT3	Mean	0.7463	0.7570	0.6296	0.9774	0.7182	0.7295	**0.4962**
	Std	0.0199	0.0131	0.1232	0.0281	0.0369	0.0695	**0.0183**
	Rank	5–	6–	2–	7–	3–	4≈	**1**
ZDT4	Mean	0.3564	0.5129	0.1859	0.9669	0.0888	0.0857	**0.0728**
	Std	0.0248	0.3404	0.1190	0.1366	0.0103	0.0090	**0.0067**
	Rank	5–	6–	4–	7–	3≈	2–	**1**
ZDT6	Mean	0.3613	0.6496	0.0998	0.1609	0.3033	0.1579	**0.0920**
	Std	0.0330	0.3393	0.0105	0.0315	0.4503	0.0288	**0.0067**
	Rank	6–	7–	2≈	4–	5–	3–	**1**
Rank Sum		29	29	16	31	17	13	5
Final Rank		6≈	5	3	7	4	2	1
better/worse/similar		0/7/0	0/7/0	0/6/1	0/7/0	0/7/0	0/7/0	/

"+", "−",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

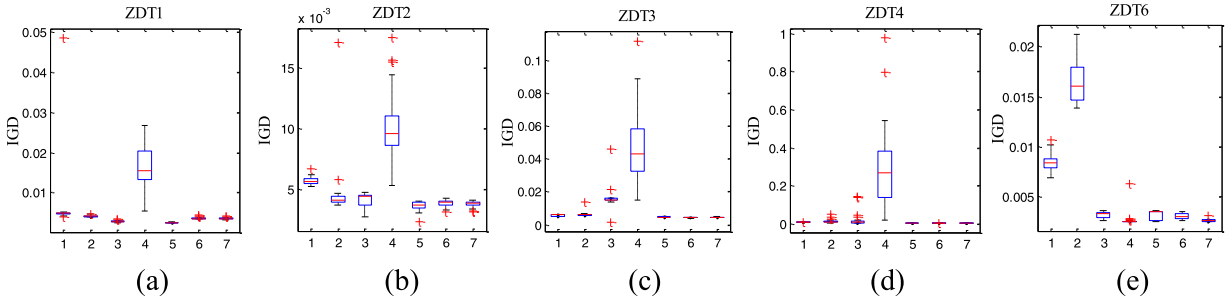

Fig. 8. Box-plots of the IGD results obtained by AUDHEIA on (a) ZDT1, (b) ZDT2, (c)ZDT3, (d) ZDT4, (e) ZDT6. (1, 2, 3, 4, 5, 6 and 7 in the horizontal axis stand for NSGAII, SPEA2, AbYSS, MOEA/D, SMPSO, HEIA, and AUDHEIA.).

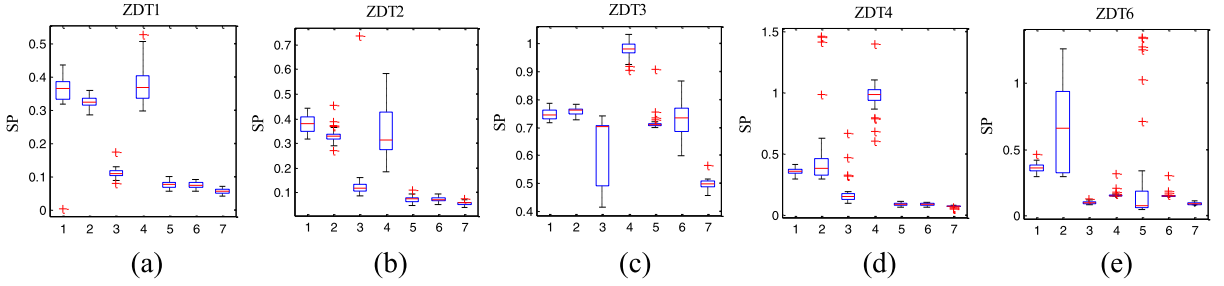

Fig. 9. Box-plots of the SP results obtained by AUDHEIA on (a) ZDT1, (b) ZDT2, (c)ZDT3, (d) ZDT4, and (e) ZDT6. (1, 2, 3, 4, 5, 6, and 7 in the horizontal axis stand for NSGAII, SPEA2, AbYSS, MOEA/D, SMPSO, HEIA and AUDHEIA, respectively.).

To further explore the SP performances of the various algorithms, box-plots are plotted in Fig. 9. The results show that AUDHEIA has the smallest SP values and that the box distances are also the least. Furthermore, outliers seldom occur. In addition, the results of AUDHEIA are close to those of SMPSO and HEIA on the ZDT2, and ZDT4 test problems. However, the SP performance of AUDHEIA is obviously better than that of these two algorithms on ZDT3. Finally, according to the above analysis, we can draw the conclusion that AUDHEIA exhibits a better distribution and more stable performance.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

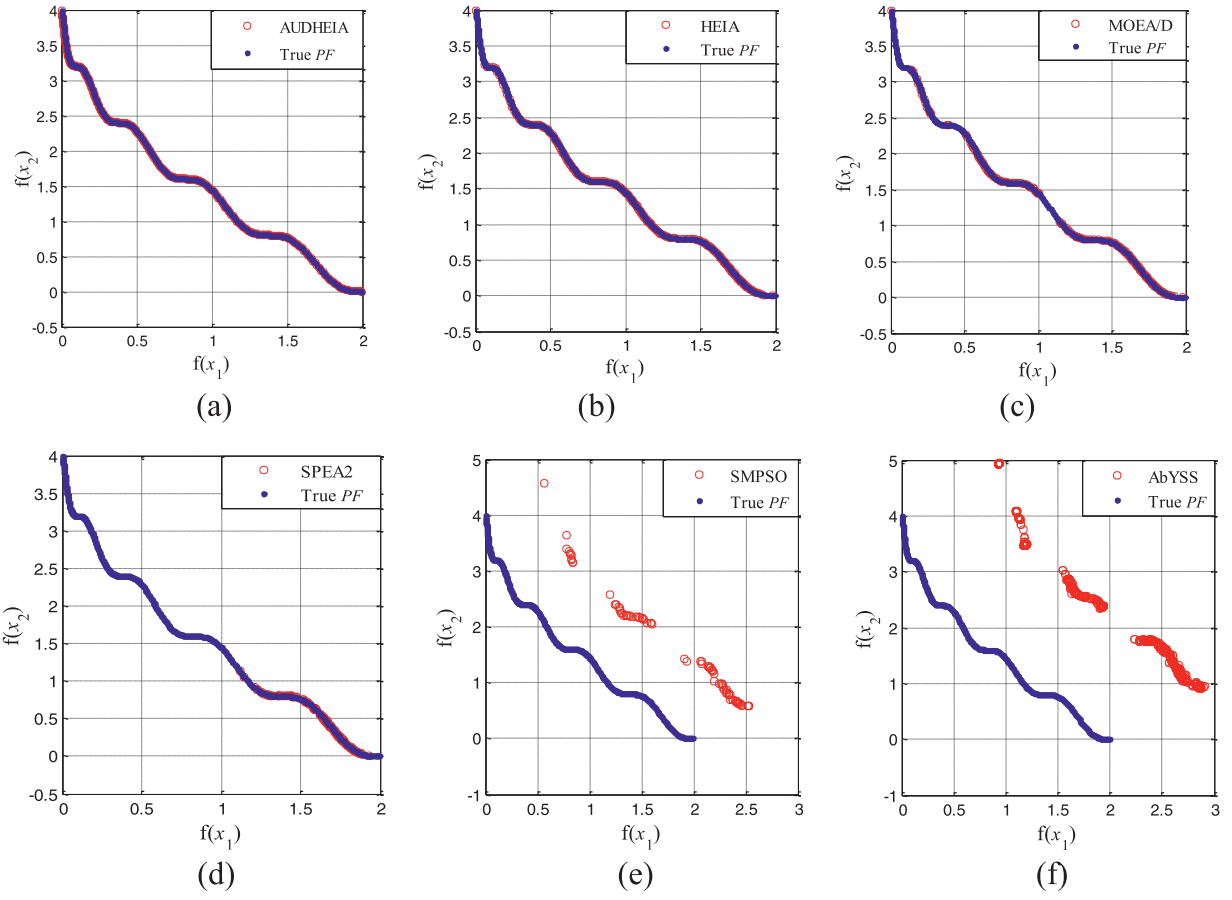

Fig. 10. Nondominated solution sets found by all the algorithms on WFG1. (a) WFG1–HEIA. (b) WFG1–MOEA/D. (c) WFG1–SPEA2. (d) WFG1–SMPSO. (e) WFG1–AUDHEIA. (f) WFG1–AbYSS.

4.4.2. Comparisons on the WFG test problems

a) Comparison of the IGD index: In Fig. 10, the convergences of all the algorithms on WFG1 are verified with simulation results. After 30 independent runs, the nearest set from the mean IGD is plotted in Fig. 10. It is observed that some algorithms can hardly approach the PF_{true} because there are many local minima. SMPSO and AbYSS fail to find the PF_{true}. SPEA2 can just approach half of PF_{true}. HEIA and MOEA/D can approximate PF_{true}, but the final sets of PF_{known} appear with nonuniform distribution along the PF_{true}. One can notice that the PF_{known} of AUDHEIA is distributed uniformly across the PF_{true}. In Table 9, for the WFG problems, the comparative results of different algorithms are illustrated. The parameters and results are all derived from [30]. The results show that AUDHEIA performs best on WFG1, WFG6, and WFG8. On WFG2, WFG3, and WFG7, SMPSO achieves the best results. In addition, AbYSS exhibits the optimal performance for WFG4. It can be seen from Wilcoxon's rank sum test that AUDHEIA achieves similar results compared to SMPSO and HEIA on WFG6 and WFG7, to SPEA2 on WFG5 and WFG9, and to MOEA/D on WFG2. Compared to ZDT problems, WFG problems seem more difficult, but in the Final rank row, we can notice that the AUDHEIA results on all of the WFG problems are the best. HEIA and SMPSO achieve the second and third ranks, respectively. Beyond that, the fourth, fifth, sixth and seventh ranks are SPEA2, MOEA/D, AbYSS, and NSGA-II, respectively.

b) Comparison of the SP index: As in Table 10, six algorithms are used to compare the SP performance with AUDHEIA in TABLE X. These algorithms are NSGA-II [9], SPEA2 [10], AbYSS [35], MOEA/D [28], SMPSO [34] and HEIA [30]. All of the parameters are derived from the reference papers. We run each algorithm 30 times. The average values of the results are listed in Table 10. As seen, the Mean and Std SP values of AUDHEIA are the smallest on most of the WFG series test problems, except for WFG2 and WFG4. Furthermore, the SP performance of AUDHEIA is similar to that of AbYSS on WFG4, WFG7 and WFG9. Again, notice that the SP performances of AbYSS are superior to those of HEIA on WFG2, WFG4, WFG5, WFG7 and WFG9. From the Rank Sum, the SP performance of AUDHEIA is the best out of all of the algorithms. ABYSS and HEIA are ranked second and third, respectively. However, NSGAII has the worst result. Beyond that, the SP performances of the other algorithms are between those of HEIA and NSGAII.

To further explore the SP performances of various algorithms, box-plots are plotted in Fig. 11. The results show that AUD-HEIA has the smallest SP values on WFG1, WFG3, and WFG5-WFG9. In addition, the box distances are also the lowest.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 9
Performance IGD comparison of results on the WFG test problems.

Problems		Algorithms						
		NSGA-II[30]	SPEA2[30]	AbYSS[30]	MOEA/D[30]	SMPSO[30]	HEIA[30]	AUDHEIA
WFG1	Mean	6.29E-01	1.22E-00	1.29E-00	1.60E-02	6.20E-02	6.15E-03	**5.81E-03**
	Std	2.18E-01	2.07E-01	2.41E-01	7.81E-03	1.51E-01	1.76E-04	**1.36E-04**
	Rank	5−	6−	7≈	3−	4−	2−	**1**
WFG2	Mean	1.04E-01	9.33E-02	1.77E-01	4.44E-02	**4.81E-03**	7.82E-02	1.26E-02
	Std	7.35E-02	6.53E-02	7.91E-02	2.61E-02	**2.73E-04**	7.54E-02	6.89E-03
	Rank	6−	5−	7−	3≈	**1+**	4−	2
WFG3	Mean	7.40E-03	6.07E-03	6.08E-03	6.73E-03	**5.52E-03**	5.79E-03	5.61E-03
	Std	3.39E-04	4.01E-04	9.31E-04	6.41E-06	**6.52E-04**	4.17E-04	3.76E-04
	Rank	7−	4−	5≈	6−	**1+**	3≈	2
WFG4	Mean	6.90E-03	6.26E-03	**5.11E-03**	8.07E-03	6.74E-03	5.46E-03	5.25E-03
	Std	3.91E-04	1.76E-04	**1.33E-04**	1.33E-03	3.68E-04	1.64E-04	1.01E-04
	Rank	6+	4−	**1+**	7−	5−	3−	2
WFG5	Mean	6.54E-02	6.49E-02	6.47E-02	6.54E-02	6.53E-02	6.50E-02	**6.35E-02**
	Std	3.07E-03	3.19E-03	3.26E-03	1.68E-04	8.85E-04	3.19E-03	**3.19E-03**
	Rank	7≈	3≈	2+	6≈	5≈	4≈	**1**
WFG6	Mean	8.64E-03	1.27E-02	1.35E-02	7.59E-03	7.30E-03	7.07E-03	**6.75E-03**
	Std	9.77E-04	6.86E-03	8.99E-03	4.01E-03	1.28E-03	1.17E-03	**1.03E-03**
	Rank	5−	6−	7−	4−	3≈	**2-**	**1**
WFG7	Mean	8.72E-03	7.18E-03	7.71E-03	8.82E-03	**5.78E-03**	6.20E-03	5.93E-03
	Std	4.79E-04	5.52E-04	2.65E-03	6.33E-04	**8.73E-05**	1.18E-04	9.65E-05
	Rank	6−	4−	5−	7−	**1**≈	3≈	2
WFG8	Mean	2.26E-02	5.13E-02	6.79E-02	2.04E-02	2.70E-02	7.89E-03	**6.63E-03**
	Std	3.28E-03	7.98E-03	7.25E-02	2.29E-03	3.65E-03	4.25E-04	**3.96E-04**
	Rank	4−	6−	7−	3−	5−	2−	**1**
WFG9	Mean	7.96E-03	6.45E-03	7.15E-03	7.48E-03	7.86E-03	6.47E-03	**6.21E-03**
	Std	3.89E-04	2.91E-04	2.31E-03	3.73E-04	7.83E-04	1.68E-04	**1.02E-04**
	Rank	7−	2−	4−	5−	6−	3≈	**1**
Rank Sum		53	40	45	44	31	26	13
Final Rank		7	4	6	5	3	2	1
better/worse/similar		0/9/0	1/8/0	2/5/2	0/8/1	2/5/2	0/7/2	/

"+", "−",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

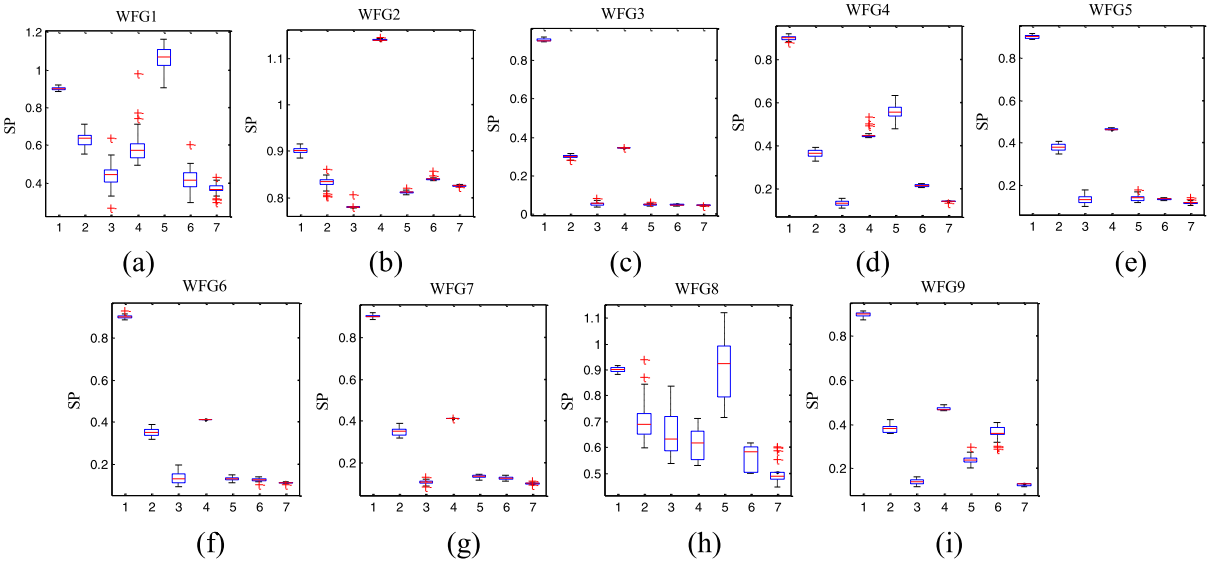

Fig. 11. Box-plots of the SP results obtained by AUDHEIA on (a) WFG1, (b) WFG2, (c)WFG3, (d) WFG4, (e) WFG5, (f) WFG5, (g) WFG5, (h) WFG5, (i) WFG5. (1, 2, 3, 4, 5, 6, and 7 in the horizontal axis stand for NSGAII, SPEA2, AbYSS, MOEA/D, SMPSO, HEIA and AUDHEI, respectively.).

Table 10
Performance SP comparison of results on the WFG test problems.

Problems		Algorithms						
		NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA	AUDHEIA
WFG1	Mean	9.01E-01	6.30E-01	4.41E-01	5.97E-01	1.06E-00	4.15E-01	**3.67E-01**
	Std	7.74E-03	3.60E-02	7.25E-02	1.01E-01	7.06E-02	6.78E-02	**1.13E-02**
	Rank	6−	5−	3−	4−	7−	2−	**1**
WFG2	Mean	9.00E-01	8.30E-01	**7.80E-01**	1.14E-00	8.11E-01	8.41E-01	8.25E-01
	Std	7.73E-03	1.46E-02	**7.08E-03**	1.47E-03	2.91E-03	3.96E-03	1.13E-03
	Rank	6−	4−	**1-**	7≈	2+	5−	3
WFG3	Mean	9.01E-01	2.99E-01	5.34E-02	3.43E-01	5.04E-02	4.78E-02	**4.57E-02**
	Std	6.78E-03	8.87E-03	9.50E-03	9.44E-05	4.09E-03	2.87E-03	**1.45E-03**
	Rank	7−	5−	4≈	6−	3+	2−	**1**
WFG4	Mean	9.00E-01	3.60E-01	**1.29E-01**	4.50E-01	5.54E-01	2.12E-01	1.38E-01
	Std	9.09E-03	1.77E-02	**1.12E-02**	2.08E-02	3.77E-02	5.54E-03	2.82E-03
	Rank	7+	4−	**1**≈	5−	6−	3−	2
WFG5	Mean	8.98E-01	3.77E-01	1.31E-01	4.64E-01	1.40E-01	1.35E-01	**1.16E-01**
	Std	8.49E-03	1.75E-02	1.86E-02	4.37E-03	1.32E-02	3.63E-03	**6.93E-03**
	Rank	7−	5+	2+	6−	4≈	3≈	**1**
WFG6	Mean	9.01E-01	3.50E-01	1.32E-01	4.10E-01	1.28E-01	1.22E-01	**1.09E-01**
	Std	8.66E-03	2.01E-02	2.69E-02	9.92E-05	1.01E-02	7.57E-03	**2.29E-03**
	Rank	7−	5−	4−	6−	3≈	2−	**1**
WFG7	Mean	9.01E-01	3.48E-01	1.05E-01	4.10E-01	1.33E-01	1.26E-01	**1.02E-01**
	Std	6.49E-03	1.95E-02	9.48E-03	8.97E-05	8.71E-03	7.41E-03	**4.18E-03**
	Rank	7−	5−	**2≈**	6−	4≈	3−	**1**
WFG8	Mean	8.99E-01	7.07E-01	6.58E-01	6.17E-01	9.08E-01	5.66E-01	**4.97E-01**
	Std	8.95E-03	7.91E-02	9.44E-02	5.65E-02	1.13E-01	4.31E-02	**3.85E-02**
	Rank	6−	5−	4−	3−	7−	2−	**1**
WFG9	Mean	8.97E-01	3.81E-01	1.39E-01	4.69E-01	2.40E-01	3.58E-01	**1.25E-01**
	Std	9.41E-03	1.70E-02	1.16E-02	5.88E-03	1.82E-02	3.37E-02	**4.45E-03**
	Rank	7−	5≈	2≈	6−	3−	4−	**1**
Rank Sum		60	43	23	49	39	26	12
Final Rank		7	5	2	6	4	3	1
better/worse/similar		0/9/0	1/8/0	2/5/2	0/8/1	2/5/2	0/9/0	/

"+", "−",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

Thus, according to the above analysis, we can draw the conclusion that AUDHEIA exhibits a better distribution and more stable performance.

4.4.3. Comparisons on the DTLZ test problems

a) Comparison of the IGD index: Based on the above experimental results, AUDHEIA has excellent performance for the ZDT and WFG problems. However, these problems have two objectives. To further research AUDHEIA, the DTLZ problems [15], which are three-objective problems, are used to detect its performance. As shown in Fig. 12, because DTLZ2 has many local PFs and DTLZ7 is discontinuous, they are difficult to solve. To compare the performances of the different algorithms, DTLZ2 and DTLZ7 are adopted to draw the results of the algorithms. A first observation is that MOEA/D is unable to approximate the PFtrue of DTLZ2 and DTLZ7 completely. In addition to that, the PF_{true} can be found by SMPSO, while the distribution of solutions is uneven. Compared with the above algorithms, AUDHEIA is well close to PF_{true} and is uniformly distributed.

The comparison results for all of the DTLZ test problems are shown in Table 11. The parameters and results are all derived from [30]. For the problems of DTLZ1, DTLZ3 and DTLZ6, AUDHEIA obtains the best performance. SPEA2 exhibits the optimal performance on DTLZ2 and DTLZ7. MOEA/D and AbYSS achieve the best results on DTLZ4 and DTLZ5, respectively. In addition, from the Wilcoxon's rank sum test, it can be seen that AUDHEIA exhibits properties similar to those of HEIA on DTLZ1 and DTLZ3, those of MOEA/D on DTLZ2, those of NSGA-II on DTLZ4 and DTLZ6, and those of AbYSS on DTLZ5. The Final rank row shows that AUDHEIA has the first rank and that SMPSO gets the second rank. Meanwhile, HEIA and SPEA2 obtain the third and fourth ranks, respectively. AbYSS, MOEA/D and NSGA-II obtain the fifth through seventh ranks.

b) Comparison of the SP index: In Table 12, the same experiments are used to test the SP performance of AUDHEIA. All the parameters are derived from the reference papers, and we run each algorithm 30 times. The average values of the results are listed in Table 12. A first observation is that the Mean and Std SP values of AUDHEIA are the smallest on the DTLZ2, DTLZ3, DTLZ6 and DTLZ7 test problems. In addition, the SP performance of AUDHEIA is similar to that of AbYSS on DTLZ5. Furthermore, MOEA/D and SPEA2 have the optimal SP performances on DTLZ1 and DTLZ4, respectively. From the

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

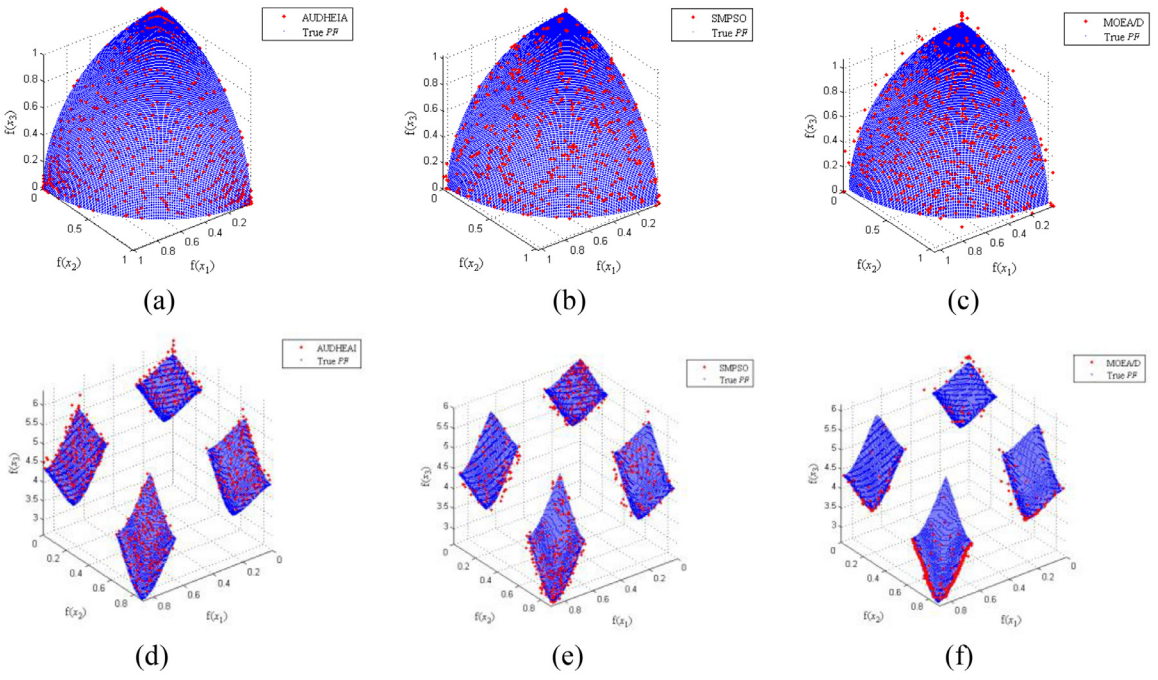

Fig. 12. Nondominated solution sets found by all the algorithms on DTLZ2 and DTLZ7. (a) DTLZ2–AUDHEIA. (b) DTLZ2–SMPSO. (c) DTLZ2–MOEA/D. (d) DTLZ7–AUDHEIA. (e) DTLZ7–SMPSO. (f) DTLZ7–MOEA/D.

Table 11
Performance IGD comparison of results on the DTLZ test problems.

Problems		Algorithms						
		NSGA-II[30]	SPEA2[30]	AbYSS[30]	MOEA/D[30]	SMPSO[30]	HEIA[30]	AUDHEIA
DTLZ1	Mean	2.18E-02	3.32E-02	2.53E-02	4.23E-02	1.28E-02	1.16E-02	**1.01E-02**
	Std	1.98E-02	3.48E-02	5.12E-02	1.17E-01	3.82E-04	3.98E-04	**1.75E-04**
	Rank	4–	6–	5–	7–	3–	2–	**1**
DTLZ2	Mean	3.07E-02	**2.38E-02**	3.02E-02	2.82E-02	3.12E-02	3.08E-02	2.68E-02
	Std	7.43E-04	**2.76E-04**	7.26E-04	1.76E-04	7.21E-04	7.57E-04	4.06E-04
	Rank	5≈	**1+**	4–	3–	7≈	6≈	2
DTLZ3	Mean	5.04E-02	3.83E-01	3.67E-02	1.45E-01	3.10E-02	3.04E-02	**2.85E-02**
	Std	1.95E-02	2.08E-01	1.63E-02	5.13E-01	5.73E-04	6.47E-04	**5.21E-04**
	Rank	5–	7–	4–	6–	3≈	2–	**1**
DTLZ4	Mean	3.08E-02	2.09E-02	2.81E-02	**1.87E-02**	2.60E-02	3.24E-02	2.92E-02
	Std	2.39E-03	1.52E-03	2.07E-03	**6.04E-04**	6.46E-03	2.33E-03	2.02E-03
	Rank	6–	2+	4≈	**1+**	3+	7–	5
DTLZ5	Mean	1.07E-03	8.73E-04	**7.96E-04**	1.85E-03	8.16E-04	8.44E-04	8.12E-04
	Std	4.20E-05	3.41E-05	**3.11E-05**	2.06E-05	2.97E-05	3.29E-05	3.01E-05
	Rank	6–	5–	**1+**	7–	3≈	4–	2
DTLZ6	Mean	1.41E-01	2.75E-01	2.04E-02	1.72E-03	7.87E-04	7.96E-04	**7.85E-04**
	Std	3.91E-02	2.09E-02	1.92E-02	7.27E-06	4.16E-05	4.17E-05	**4.17E-05**
	Rank	6–	7–	5–	4–	2≈	3–	**1**
DTLZ7	Mean	3.05E-02	**2.39E-02**	1.20E-01	7.99E-02	3.35E-02	3.21E-02	2.98E-02
	Std	8.03E-04	**3.98E-04**	1.37E-01	8.61E-02	1.53E-03	1.12E-03	9.56E-04
	Rank	3≈	**1+**	7–	6–	5–	4–	2
Rank Sum		35	29	30	34	26	28	14
Final Rank		7	4	5	6	2	3	1
better/worse/similar		0/7/0	3/4/0	1/5/1	1/6/0	1/4/2	0/7/0	/

"+", "–",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Table 12
Performance SP comparison of results on the DTLZ test problems.

Problems		Algorithms						
		NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA	AUDHEIA
DTLZ1	Mean	0.8515	0.8082	0.8639	**0.4981**	0.7095	0.6885	0.5987
	Std	0.0240	0.2435	0.0260	**0.0308**	0.0384	0.0280	0.0071
	Rank	6–	5–	7≈	**1+**	4–	3–	2
DTLZ2	Mean	0.9079	0.6149	0.7952	0.9285	0.6210	0.5583	**0.3301**
	Std	0.0342	0.0380	0.0433	0.0260	0.0330	0.0323	**0.0323**
	Rank	6–	3–	5–	7–	4≈	2–	**1**
DTLZ3	Mean	0.8164	1.3626	0.7887	0.9227	0.6320	0.6070	**0.5780**
	Std	0.0254	0.0938	0.0472	0.1248	0.0286	0.0119	**0.0054**
	Rank	5–	7–	4–	6–	3–	2–	**1**
DTLZ4	Mean	0.7241	**0.5864**	0.6926	0.8012	0.6376	0.7533	0.6768
	Std	0.0082	**0.0151**	0.0216	0.0228	0.0308	0.0075	0.0052
	Rank	5–	**1+**	4–	**7-**	2+	6–	3+
DTLZ5	Mean	0.4536	0.3734	**0.1104**	0.7554	0.1398	0.1410	0.1184
	Std	0.0788	0.0334	**0.0071**	0.0132	0.0364	0.0019	0.0029
	Rank	6–	5–	**1≈**	7–	**3-**	4≈	2
DTLZ6	Mean	0.5146	0.7616	1.0958	0.7653	0.1042	0.1255	**0.0928**
	Std	0.0177	0.0325	0.1894	0.0019	0.0222	0.0047	**0.0028**
	Rank	4–	5–	7–	6≈	**2-**	3–	**1**
DTLZ7	Mean	1.0353	0.8668	0.7721	1.0749	0.7390	0.6988	**0.2943**
	Std	0.0663	0.0324	0.0645	0.0604	0.0956	0.0994	**0.0223**
	Rank	6–	5–	4≈	7+	3–	2–	**1**
Rank Sum		38	31	32	41	21	22	11
Final Rank		6	4	5	7	2	3	1
better/worse/similar		0/7/0	1/6/0	0/6/1	1/6/0	1/6/0	0/7/0	/

"+", "−",and "≈" indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by AUDHEIA using Wilcoxon's rank sum test with a significant level $\rho = 0.05$, respectively.

Table 13
Final rank of all the algorithms on the ZDT, WFG, and DTLZ problems.

Problems	Algorithms						
	NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA	AUDHEIA
Rank Sum on ZDTs	21	23	19	25	10	14	7
Rank Sum on WFGs	44	32	37	36	25	26	15
Rank Sum on DTLZs	28	25	26	28	19	28	21
Total Rank sum on all the Problems	93	80	82	89	54	68	43
Final Rank on all the Problems	7	4	5	6	2	3	1

Table 14
Final comparisons of all the algorithms on the ZDT, WFG, and DTLZ problems.

Problems	Algorithms					
	NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA
ZDTs	0/5/0	0/5/0	0/4/1	1/4/0	1/2/2	0/4/1
WFGs	0/9/0	1/8/0	2/5/2	0/8/1	2/5/2	0/7/2
DTLZs	2/4/1	3/4/0	3/3/1	2/3/2	2/1/4	0/7/0
better/worse/similar	2/18/1	4/17/0	5/12/4	3/11/3	5/8/8	0/18/3
Final result (AUDHEIA vs other algorithms)	*better*	*better*	*better*	*better*	*better*	*better*

Rank Sum, the SP performance of AUDHEIA is obviously better than that of other algorithms. HEIA and SMPSO are similar and are ranked second and third, respectively. However, the result of MOEA/D is the worst.

To further explore the SP performances of various algorithms, box-plots are plotted in Fig. 13. The results show that AUDHEIA has the smallest SP values, except on DTLZ1, DTLZ4 and DTLZ5. In addition, its box distances are also the lowest. In addition, outliers seldom occur. Finally, we can draw the conclusion that AUDHEIA exhibits a better distribution and more stable performance. Tables 13 and 14 list the IGD rank sums of NSGA-II, SPEA2, AbYSS, MOEA/D, SMPSO, HEIA and AUDHEIA on all of the ZDT, WFG, and DTLZ test algorithms. When all of the test problems are considered, it is easy to see from the final rank that AUDHEIA performs better than other problems. In addition, we have summed up the comparison results of AUDHEIA with the other algorithms in Table 13. The second-to-last line reveals the comparative results between AUDHEIA

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

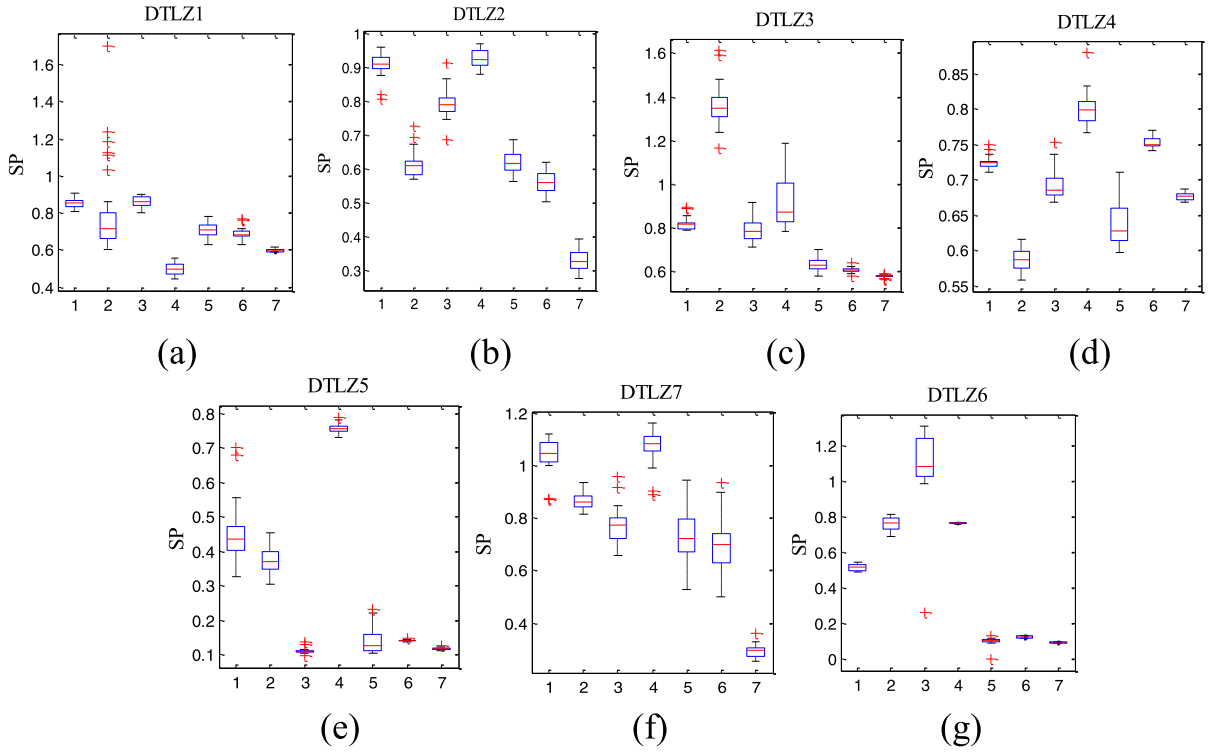

Fig. 13. Box-plots of the SP results obtained by AUDHEIA on (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4, (e) DTLZ5, (f) DTLZ6 and (b) DTLZ7. (1, 2, 3, 4, 5, 6, and 7 in the horizontal axis stand for NSGAII, SPEA2, AbYSS, MOEA/D, SMPSO, HEIA and AUDHEIA.).

Table 15
Function calculation comparison of different algorithms.

Algorithms	Objective function number of times called	Function number of times called	Maximum value	Minimum value	Average values
AUDHEIA	**2600**	**2010**	**2900**	**1450**	**1860**
NSGAII-DLS[33]	3010	2240	3500	2030	2100
NSGAII[9]	20,000+	20,000+	20,000+	20,000+	5150
HMOEA/D[28]	13,200	9600	10,500	7050	6900
NSGAII-els[5]	13,350	8550	17,250	13,200	3000

and the others on all the test problems. Furthermore, the last line illustrates the final results of the comparative analyses. From the above results, the performance of AUDHEIA is better than that of the other compared algorithms. Based on the above analysis, it is concluded that AUDHEIA can be used to solve different test problems. In addition, when solving complex problems, such as WFG problems, its advantages are more obvious.

4.4.4. Comparisons of convergence speed

To verify the convergence speed of AUDHEIA, the number of times that a function is called before the specified performance index is reached is adopted in this paper. According to the experiments in reference [24], the optimization is stopped when an IGD value of 0.01 is reached, and then the number of times that the function was called is recorded. The results of this experiment are shown in Table 15.

The average values of the numbers of times that the function is called after a ten-time continuous experimental study are listed in Table 15. Five indexes are used to evaluate the convergence speed. They are the objective function's number of times called, the functional number of times called, and the Maximum, Minimum, and Average values of times called. In addition, four algorithms (NSGAII-DLS, NSGAII [9], HMOEA/D [38], and NSGAII-els [23]) are used for comparison with AUDHEIA. The parameters and results are all derived from [27]. From the simulation results, it can be seen that the numbers of times the objective function and function of AUDHEIA are called appear to be the lowest. In addition, the results show that NSGAII-DLS is similar to AUDHEIA with regard to the function's number of times being called. HMOEA/D and NSGA-els also have similar results. However, their objective function's and function's number of times called are significantly higher than those of AUDHEIA. Beyond that, the results of NSGAII are above 20,000, which is the worst performance. We can also see that AUDHEIA has the lowest maximum, minimum, and average values of the function's computation time. According to the above analysis, we can conclude that AUDHEIA can quickly converge to the $\mathbf{PF}_{\text{true}}$.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

5. Discussion

The goal of this paper is to increase the distribution of the individuals during iterations. Then, an adaptive uniform distribution selection framework of AUDHEIA is proposed in this paper. According to the above analysis and results, the following observations can be made.

5.1. High convergence precision

Convergence accuracy is the key issue for the evolutionary algorithm, and better diversity can improve it. In this paper, a uniform distribution selection mechanism is proposed. Then, better diversity can be obtained, along with the improvement of the distribution. Furthermore, the limit optimization variation strategy of the best individual can avoid falling into local optimum, and then the global optimal solution can be better found. In Tables 13 and 14, the results show that AUDHEIA can search for more accurate solutions to obtain higher convergence accuracy.

5.2. Better distribution

The distribution is another important issue for the evolutionary algorithm. In this paper, the distribution enhancement module is activated during the iteration process when the distribution is not satisfied in the corresponding interval. In addition, the distribution of individuals change unceasingly during the iteration process, and the threshold is adjusted adaptively. In Tables 10, 12, and 13, the experimental results indicate that AUDHEIA exhibits better distribution and more stable performance.

5.3. Faster convergence speed

Convergence speed is also an important issue for AUDHEIA. In this paper, the limit optimization variation strategy of the best individual can avoid falling into local optima, and improve the search speed. In addition, the nondominated solutions can quickly locate the Pareto front when individuals are evenly spread in iterations. From Table 15, the results indicate that AUDHEIA can quickly converge to the PF_{true}.

6. Conclusion

This paper presents an adaptive hybrid evolutionary immune multi-objective algorithm based on a uniform distribution selection mechanism. The experimental results show that the proposed AUDHEIA has a good approximation of PF_{true}, and the distribution of individuals is better than that of the other algorithms discussed in this paper. In addition, it has a faster convergence speed. Compared with the other algorithms, the advantages of AUDHEIA are as shown below:

1) A distribution measurement module is proposed to measure the distribution of the individu*als.* A standard threshold is used to judging the distribution level. Because the distribution of individuals changes unceasingly during the iteration process, the threshold is adjusted adaptively. When the number of clusters in the interval is less than the threshold, the distribution enhancement module is activated. Otherwise, the intervals meet the distribution. Thus the distribution of AUDHEIA is better than that of the others algorithms discussed in this paper.
2) The individuals are mapped to the hyperplane that corresponds to the objective space, and the individuals are clustered on this plane. In addition, the hyperplane is equally divided, and excellent individuals with the same number are selected from every interval. After that, the selected individuals from different clusters are stored in an elitism archive. Thus, the diversity of the population can be improved.
3) Two variation strategies are used to supplement the insufficient individuals. The first strategy can effectively improve the local search ability. The other one can avoid falling into local optima and can thus improve the search speed. In addition, the individual distribution is more uniform during the iterative process. When very few better individuals find the Pareto front, the rest of the individuals can be more quickly and evenly distributed on the Pareto front. Then, the search speed can be improved.

We also experimented on NSGAII with this adaptive uniform distribution method, which also achieved good application results. Therefore, it can be verified that this method can be applied to other evolutionary algorithms to solve multi-objective optimization problems. Although the performance of AUDHEIA is very promising, there are still several issues worth studying for further improvements. Further study will be conducted in the future. First, when the sparse degree of PF_{true} is different or empty, the mechanism should be further improved. Second, a new method should be designed for dynamic multi-objective optimization problems. Third, further studies on MOPs with more than three objectives [1,8] or in noisy environments [37] are necessary. Moreover, we can also apply it to real-world applications [41].

Declaration of Competing Interest

The authors declare that there is no conflict of interest regarding the publication of this work.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants 61603012, 61533002 and 61603009, the Beijing Municipal Education Commission Foundation under grant KM201710005025, the Beijing Postdoctoral Research Foundation under grant 2017ZZ-028, the China Postdoctoral Science Foundation funded project as well as Beijing Chaoyang District Postdoctoral Research Foundationunder grant 2017ZZ-01-07, the Beijing Natural Science Foundation 4182007.

References

[1] S. Bandyopadhyay, A. Mukherjee, An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution, IEEE Trans. Evol. Comput. 19 (3) (2015) 400–413.
[2] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multi-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188.
[3] J.X. Cheng, G.G. Yen, G.X. Zhang, A grid-based adaptive multi-objective differential evolution algorithm, Inform. Sci. 367 (1) (2016) 890–908.
[4] C.A.C. Coello, N.C. Cortés, An approach to solve multiobjective optimization problems based on an artificial immune system, in: first International Conference on Artificial Immune Systems (ICARIS'2002), UK, 2002, pp. 212–221.
[5] C.A.C. Coello, N.C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genet. Program. Evol. Mach. 6 (2) (2005) 163–190.
[6] D.W. Corne, J.D. Knowles, M.J. Oates, The Pareto envelope-based selection algorithm for multi-objective optimization, in: International Conference on Parallel Problem Solving from Nature, Berlin, Germany, 2000, pp. 839–848.
[7] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[8] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[10] Z. Eckart, L. Marco, T. Lothar, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, 2001, p. 103. TIK-Report.
[11] F. Freschi, M. Repetto, VIS: an artificial immune network for multi-objective optimization, Eng. Optimiz. 38 (8) (2006) 975–996.
[12] J.Q. Gao, J. Wang, WBMOAIS: a novel artificial immune system for multiobjective optimization, Comput. Oper. Res. 37 (1) (2010) 50–61.
[13] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Psychology Press, 1987, pp. 28–31.
[14] M.G. Gong, L.C. Jiao, H.F. Du, L.F. Bo, Multi-objective immune algorithm with nondominated neighbor-based selection, Evol. Comput. 16 (2) (2008) 225–255.
[15] D. Han, W.L. Du, W. Du, Y.C. Jin, C.P. Wu, An adaptive decomposition-based evolutionary algorithm for many-objective optimization, Inform. Sci. 491 (2019) 204–222.
[16] H.G. Han, W. Lu, J.F. Qiao, An adaptive multi-objective particle swarm optimization based on multiple adaptive methods, IEEE Trans. Cybern. 47 (9) (2017) 2754–2767.
[17] Z.H. Hu, A multiobjective immune algorithm based on a multipleaffinity model, Eur. J. Oper. Res. 202 (1) (2010) 60–72.
[18] W. Hu, G.G. Yen, Adaptive multi-objective particle swarm optimization based on parallel cell coordinate system, IEEE Trans. Evol. Comput. 19 (1) (2015) 1–18.
[19] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Germany, 2005, pp. 280–295.
[20] T. Jansen, C. Zarges, Reevaluating immune-inspired hypermutations using the fixed budget perspective, IEEE Trans. Evol. Comput. 18 (5) (2014) 674–688.
[21] S.Y. Jiang, S.X. Yang, An improved multi-objective optimization evolutionary algorithm based on decomposition for complex Pareto fronts, IEEE Trans. Cybern. 46 (2) (2016) 421–437.
[22] L.C. Jiao, M.G. Gong, R.H. Shang, H.F. Du, B. Lu, Clonal selection with immune dominance and energy based multiobjective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Germany, 2005, pp. 474–489.
[23] H. Kim, M.S. Liou, Adaptive directional local search strategy for hybrid evolutionary multiobjective optimization, App. Soft Comput. 19 (2014) 290–311.
[24] J. Knowles, D. Corne, Properties of an adaptive archiving algorithm for storing nondominated vectors, IEEE Trans. Evol. Comput. 7 (2) (2003) 100–116.
[25] K.B. Lee, J.H. Kim, Multi-objective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots, IEEE Trans. Evol. Comput. 17 (6) (2013) 755–766.
[26] K. Li, S. Kwong, J.J. Cao, M.Q. Li, J.H. Zheng, R.M. Shen, Achieving balance between proximity and diversity in multi-objective evolutionary algorithm, Inform. Sci. 182 (1) (2012) 220–242.
[27] S.Y. Li, W.J. Li, J.F. Qiao, A local search NSGA2 algorithm based on density, Control Decis. 1 (2018) 60–66 (in China).
[28] H. Li, Q.F. Zhang, Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[29] Q.Z. Lin, J.Y. Chen, A novel micro-population immune multi-objective optimization algorithm, Comput. Oper. Res. 40 (6) (2013) 1590–1601.
[30] Q.Z. Lin, J.Y. Chen, Z.-.H. Zhan, W.-.N. Chen, C.A.C. Coello, Y.L. Yin, C.-.M. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multi-objective optimization problems, IEEE Trans. Evol. Comput. 20 (5) (2015) 711–729.
[31] Q.Z. Lin, Y.P. Ma, J.Y. Chen, Q.L. Zhu, C.A.C. Coello, K.C. Wong, F. Chen, An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies, Inform. Sci. 430 (2018) 46–64.
[32] J.P. Luo, Y. Yang, Q.Q. Liu, X. Li, M.R. Chen, K.Z. Gao, A new hybrid memetic multi-objective optimization algorithm for multi-objective optimization, Inform. Sci. 448 (2018) 164–186.
[33] J.N. Morse, Reducing the size of the nondominated set: pruning by clustering, Comput. Oper. Res. 7 (1–2) (1980) 55–66.
[34] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A.C. Coello, F. Luna, E. Alba, SMPSO: a new PSO-based metaheuristic for multi-objective optimization, in: 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), Nashville, USA, 2009, pp. 66–73.
[35] A.J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J.J. Durillo, A. Beham, AbYSS: adapting scatter search to multi-objective optimization, IEEE Trans. Evol. Comput. 13 (1) (2008) 439–457.
[36] R.H. Shang, L.C. Jiao, F. Liu, W.P. Ma, A novel immune clonal algorithm for MO problems, IEEE Trans. Evol. Comput. 16 (1) (2012) 35–50.
[37] V.A. Shim, K.C. Tan, J.Y. Chia, A.A. Mamun, Multi-objective optimization with estimation of distribution algorithm in a noisy environment, Evol. Comput. 21 (1) (2013) 149–177.
[38] K. Sindhya, K. Miettinen, K. Deb, A hybrid framework for evolutionary multi-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2013) 495–511.
[39] L.X. Tang, X.P. Wang, A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems, IEEE Trans. Evol. Comput. 17 (1) (2013) 20–45.
[40] J. Yoo, P. Hajela, Immune network simulations in multicriterion design, Struct. Optim. 18 (2–3) (1999) 85–94.
[41] Z.H. Zhan, X.F. Liu, Y.J. Gong, J. Zhang, H.S.H. Chung, Y. Li, Cloud computing resource scheduling and a survey of its evolutionary approaches, ACM Comput. Surv. 47 (4) (2015) 63.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032

[42] Z.H. Zhang, Immune optimization algorithm for constrained nonlinear ultiobjective optimization problems, Appl. Soft. Comput. 7 (3) (2007) 840–857.
[43] Z.H. Zhang, Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control, Appl. Soft. Comput. 8 (2) (2008) 959–971.
[44] Q.F. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[45] Q.F. Zhang, W.D. Liu, E. Tsang, B. Virginas, Expensive multi-objective optimization by MOEA/D with Gaussian process model, IEEE Trans. Evol. Comput. 14 (3) (2010) 456–474.
[46] Q.F. Zhang, A. Zhou, S.Z. Zhao, P.N. Suganthan, W.D. Liu, S. Tiwari, in: Multi-Objective Optimization Test Instances for the CEC 2009 Special Session and Competition, University of Essex, Singapore, 2009, p. 264. Special session on performance assessment of multi-objective optimization algorithms, Technical report.
[47] X.J. Zhu, T. Chen, Schr, Pareto multi-objective genetic algorithm with multi individual participation, Electron. J. 29 (1) (2001) 106–109 (in Chinese).
[48] E. Zitzler, K. Deb, L. Thiele, Comparison of multi-objective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.

Please cite this article as: J. Qiao, F. Li and S. Yang et al., An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.032