

# ACSAMO: An Adaptive Multiobjective Optimization Algorithm using the Clonal Selection Principle

X.L. WANG<sup>1</sup> and M. Mahfouf<sup>2\*</sup>

<sup>1</sup>Department of Electrical Engineering and Information Engineering, The Lanzhou University of Technology, Lanzhou, 730050, China,

Email: [Xiaolan.WANG@sheffield.ac.uk](mailto:Xiaolan.WANG@sheffield.ac.uk)

<sup>2</sup>Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, United Kingdom

Phone: +44 (0)114 222 5607, Fax: +44 (0)114 222 5624

\*Corresponding Author Email: [M.Mahfouf@sheffield.ac.uk](mailto:M.Mahfouf@sheffield.ac.uk)

**Abstract** In this paper, an adaptive multi-objective optimization algorithm based on the clonal selection principle is proposed. This algorithm uses the adaptive mutation guided by non-dominated sorting and a weighted aggregation function. In the iteration process, the antigens to be recognized are dynamically adjusted, and the size of the Pareto memory set is adaptively changed according to the concentration metric based on the crowding distance. As a result, the diversity of the Pareto solutions is improved while keeping a good convergence performance in relation to the “true” Pareto front. Direct comparisons with other evolutionary multi-objective optimization techniques such as SPEA, NSGA-II show that the proposed algorithm is better in terms of convergence and diversity along the Pareto front respectively.

**KeyWords** Artificial Immune System, Clonal Selection Algorithm, Evolutionary Multiobjective Optimization, Adaptive, Pareto.

## 1. Introduction

Many real-world optimization problems involve multiple objectives that need to be considered simultaneously. As these objectives are usually conflicting, it is not possible to find a single solution that is optimal with respect to all objectives. Instead, there exist a number of so called “Pareto-optimal” solutions which are characterized by the fact that an improvement in anyone objective can only be obtained at the expense of degradation in at least one other objective. Such set of solutions is called *Pareto* optimal set, and the corresponding vector in the objective space is called a Pareto front.

Evolutionary algorithms (EAs) seem to be particularly suited to explore the design space for the Pareto-optimal solutions. Some researchers suggested that multiobjective search and optimization might be an interesting area where EAs do better than other stochastic search strategies. The first real application of EAs in finding multiple trade-off solutions in one single simulation run was suggested and worked out in 1985 by Schaffer<sup>[1]</sup>. His Vector-Evaluated GA made a simple modification to a single-objective GA and demonstrated that GA can be used to capture multiple trade-off solutions for a few iterations. In 1989, E.Goldberg<sup>[2]</sup> suggested a sketch of multi-objective evolutionary algorithm (MOEA) using the concept of domination. Taking the clue from his book, a number of researchers have developed different implementation of MOEAs. The algorithm “Strength Pareto Evolutionary Algorithm (SPEA)” introduced by Zitzler and Thiele<sup>[3]</sup> was conceived as a way of integrating different MOEAs. Deb et al.<sup>[4]</sup> proposed a revised version of the NSGA, called NSGA-II. This algorithm is more efficient than the previous mentioned algorithm, uses elitism and a crowded comparison operator that keeps diversity without specifying and additional parameters.

The natural immune system (IS) is a complex adaptive pattern-recognition system that defends the body from foreign pathogens. It is able to categorize all cells within the body as either belonging to its own kind (self-cell) or those that have a foreign origin (nonself cell). Rather than rely on any central control, it has a distributed task force that has the intelligence to take action from a local and also global perspective using its network of chemical messengers for communication <sup>[6],[7],[8]</sup>. Artificial immune systems (AIS) emerged in the 1990s<sup>[9]</sup> as a new computational paradigm in AI. That can be defined as computational systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving <sup>[10]</sup>. The immune system is a system with high complexity and the current AIS works adopted only a few immune mechanisms. Specifically, three immunological principles are primarily used in AIS methods. These include immune network theory, the mechanisms of negative selection, and the clonal selection principles.

The clonal selection principle is used to explain the basic features of an adaptive immune response to an antigenic stimulus. DeCastro <sup>[11]</sup> proposed an artificial immune network for data analysis: aiNet. In aiNet, a minimum spanning tree algorithm is used to identify clusters. A later version of aiNet was developed to solve multimodel function optimization problems. This algorithm (CLONAL) <sup>[12]</sup> is based on the clonal selection and affinity maturation principle and focused on the multimodel function optimization. In this paper we extend the clonal selection principle to multiobjective optimization problems and we proposed the Adaptive Clonal Selection Algorithm for Multiobjective Optimization problem (ACSAMO). By direct comparison with the well-known multiobjective optimization algorithm, our approach performs better in terms of convergence and diversity along the Pareto front.

The remaining parts of the paper are organized as follows: Section 2 presents the adaptive clonal selection principle for multiobjective optimization. Section 3 will show the simulation experiment results using the proposed algorithm to solve the multiobjective optimization benchmark function ZDT1-ZDT4. This section also introduces some performance metrics and the comparative result between our approach and other commonly recognized effective multiobjective optimization algorithms. Finally, discussions and conclusions remarks will be given in Section 4.

## 2. A Clonal Selection Algorithm for Multiobjective Optimization

### 2.1 Multiobjective Optimization

The aim of multiobjective optimization is to find a vector of decision variables that satisfy the constraints and to optimize a vector function whose elements represent the objective functions <sup>[5]</sup>. These functions form a mathematical description of performance criteria that are usually in conflict with each other. Hence, the term “optimize” means finding such a solution that would give the values of all the objective functions acceptable to the designer. Formally, the general multiobjective optimization problem can be stated as follows:

**Definition 1:** Find the vector:

$$\mathbf{x}^* = [x^*_1, x^*_2, \dots, x^*_n]^T$$

Which will satisfy J inequality constraints, K equality constraints and make the M vector functions to be minimized or maximized. That is:

$$\text{Minimize / Maximize: } f_m(\mathbf{x}) \quad m=1,2,\dots,M \quad (1)$$

$$\text{Subject to: } g_j(\mathbf{x}) \geq 0 \quad j=1,2,\dots,J \quad (2)$$

$$h_k(\mathbf{x}) = 0 \quad k=1,2,\dots,K \quad (3)$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i=1,2,\dots,n$$

Most multiobjective optimization algorithms use the concept of domination. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not. We use the operator  $\triangleleft$  between two solutions  $i$  and  $j$  as  $i \triangleleft j$  to denote that solution  $i$  is better than solution  $j$  on a particular objective. Hence, the following definition covers mixed problems with minimization as well as the maximization of some objective function.

**Definition 2:** a solution  $x^{(1)}$  is said to dominate the other solution  $x^{(2)}$ , if both condition 1 and conditions 2 are true:

**Conditions 1:** The solution is  $x^{(1)}$  no worse than  $x^{(2)}$  in all objectives.

**Conditions 2:** The solution  $x^{(1)}$  is strictly better than  $x^{(2)}$  in at least one objective.

Another important concept is that of Pareto optimality. On many occasions, the global Pareto-optimal set is simply referred to as the Pareto-optimal set. We define a global Pareto-optimal set as follows:

**Definition 3:** The non-dominated set of the entire feasible search space is the global Pareto-optimal set.

## 2.2 The artificial immune system and the clonal selection principle

Any molecule that can be recognized by the adaptive immune system is known as an antigen (Ag). When an animal is exposed to antigens (Ag's), some subpopulation of its bone-marrow-derived cells responds by producing antibodies (Ab's). Ab's are molecules which are attached primarily to the surface of B cells whose aim is to recognize and bind to Ag's. Each B cell secretes a single type of Ab, which is relatively specific for the Ag. By binding to these Ab's and with a second signal from accessory cells, such as the T-helper cell, the Ag stimulates the B cell to proliferate and mature into terminal Ab secreting cells, called plasma cells. The process of cell division generates a clone, i.e., a cell or set of cells that are the progenies of a single cell. The clonal selection principle establishes the idea that only those cells that recognize the antigens (Ag's) are selected to proliferate. The selected cells are subject to an affinity maturation process, which improves their affinity to the selective Ag's. The main immune aspects taken into account to develop the algorithm are: 1) maintenance of a specific memory set; 2) selection and clone of the most stimulated Ab's; and, 3) affinity maturation. Our ACSAMO is composed basically of two populations: a set of antigens Ag's and a set of antibodies Ab's. The set of Ag's maintain the best solution found along the evolutionary process. The set of Ab's can be decomposed into two subsets, one we call the memory set that is used to store the non-dominated solutions vectors found so far, the other set is the basic Ab's population for evolution. In our algorithm no encoding is performed, each cell is a real-valued vector in an Euclidean shape-space. The following terminology is adopted:

- **Fitness:** fitness of an individual in relation to an objective function to be optimized.
- **Affinity:** the absolute distance between the current individuals and the best individual.

- **Clone and mutation:** clone is the offspring cells that are the identical copies of their parent cell. The offspring will further suffer a somatic mutation so that they become variations of their parent.

## 2.3 Adaptive clonal selection algorithm for multiobjective optimization

### 2.3.1 The fitness and affinity

The fitness is defined as the follows vector of multiobjective function as follows:

$$\mathbf{F} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T \quad (4)$$

For a two objective optimization:

$$\mathbf{F} = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$$

Where:  $m=1,2,\dots,M$ ,  $M$  is the number of objective functions to be optimized.

$\mathbf{x} = [x_1, x_2, \dots, x_j, \dots, x_d]^T$  is the vector of decision variables. The vectors consist of the antibodies population to be evolved.

$j=1,2,\dots,d$ ,  $d$  is the numbers of decision variables.

We assume that  $x_c$  represents the best previous population and  $x_g$  denotes the best solution among all solutions in the population. That are the antigens we need to approach. The affinity between antigens and antibodies is then defined in a decision variable space as follows:

$$\mathbf{AF} = [af_1, af_{22}, \dots, af_N]^T \quad (5)$$

$i=1,2,\dots,N$ ,  $N$  is the size of basic population.

$$\text{Where: } af_i = |x_i - x_c| + |x_i - x_g| \quad (6)$$

Hence the antigens to be recognized by antibodies are dynamically adjusted in the evolutionary process.

In order to calculate  $x_c$ ,  $x_g$  and evaluate the performance of individuals, an appropriate evaluation function can be defined. We use a weighted aggregation approach to construct the evaluation function WF (weighted function) for multiobjective optimization:

$$WF = \sum_{m=1}^M w_m f_m; \quad \sum_{m=1}^M w_m = 1 \quad (7)$$

To approximate the Pareto front instead of a certain Pareto solution, the weights  $w_m$  for each objective are changed systematically and normalized as follows:

$$w_m = \frac{\lambda_m}{\sum_{m=1}^M \lambda_i} \quad \lambda_m = U(0,1)$$

The function  $U(0,1)$  generates a uniformly distributed random number within the interval  $[0,1]$ . In this way, we can obtain a uniformly distributed random weight combination, which is generated at every

iteration cycle. The idea here is to use dynamic weights <sup>[13]</sup> instead of fixed weights to obtain the Pareto solutions. This dynamically weighted aggregation approach was introduced for the selection of the best  $x_c$  and  $x_g$ . We will show that this approach works very well with multiobjective problems.

### 2.3.2 Clone and Mutation

As already stated, a clone is the identical copy of a parent. In our algorithm, we need to decide the size of such a clone. Here, we used the same number of clones generated for each Ab's. The total number of cloning becomes:

$$N_c = \sum_{i=1}^N \text{round}(\beta * N) \quad (8)$$

Where,  $\beta$  is the multiplying factor,  $N$  is the total number of Ab's and  $\text{round}(\cdot)$  is the operator that rounds its argument toward the closest integer. Each term of this sum corresponds to the clone size of each selected Ab.

The affinity proportional mutation is performed according to the following expression:

$$x_i^m = x_i + \gamma * \alpha * N(0,1) \quad (9)$$

Where:

$$\alpha = \frac{\sum_{j=1}^d af_{i(j)}}{d} \quad (10)$$

$x_i^m$  is the mutation cell of  $x_i$ .  $N(0,1)$  is a Gaussian random variable of zero mean and standard deviation  $\sigma=1$ .  $\gamma$  is the factor of mutation intension. This expression indicates that the worst individuals suffered a mutation much bigger than the best one.

### 2.3.3 The Selection Operator

After cloning and mutation, the new offspring combines with the parent population to form a temporary population. Based on non-dominated sorting and crowding distance, the algorithm generates the next generation of individuals and non-dominated solution memory set as follows:

- 1) Non-dominated sorting:
  - a) Identify the non-dominated solutions in the temporary population and store them in a matrix Pfront (Pareto Front), set front count  $c=1$ ;
  - b) Remove the non-dominated individuals from the temporary population;
  - c)  $c=c+1$ , identify the non-dominated solutions in the remaining population and store them in a matrix Frontk (front k);
  - d) Repeat b) and c) until all individuals are separated into different fronts.
- 2) Select the individuals for the next iteration:

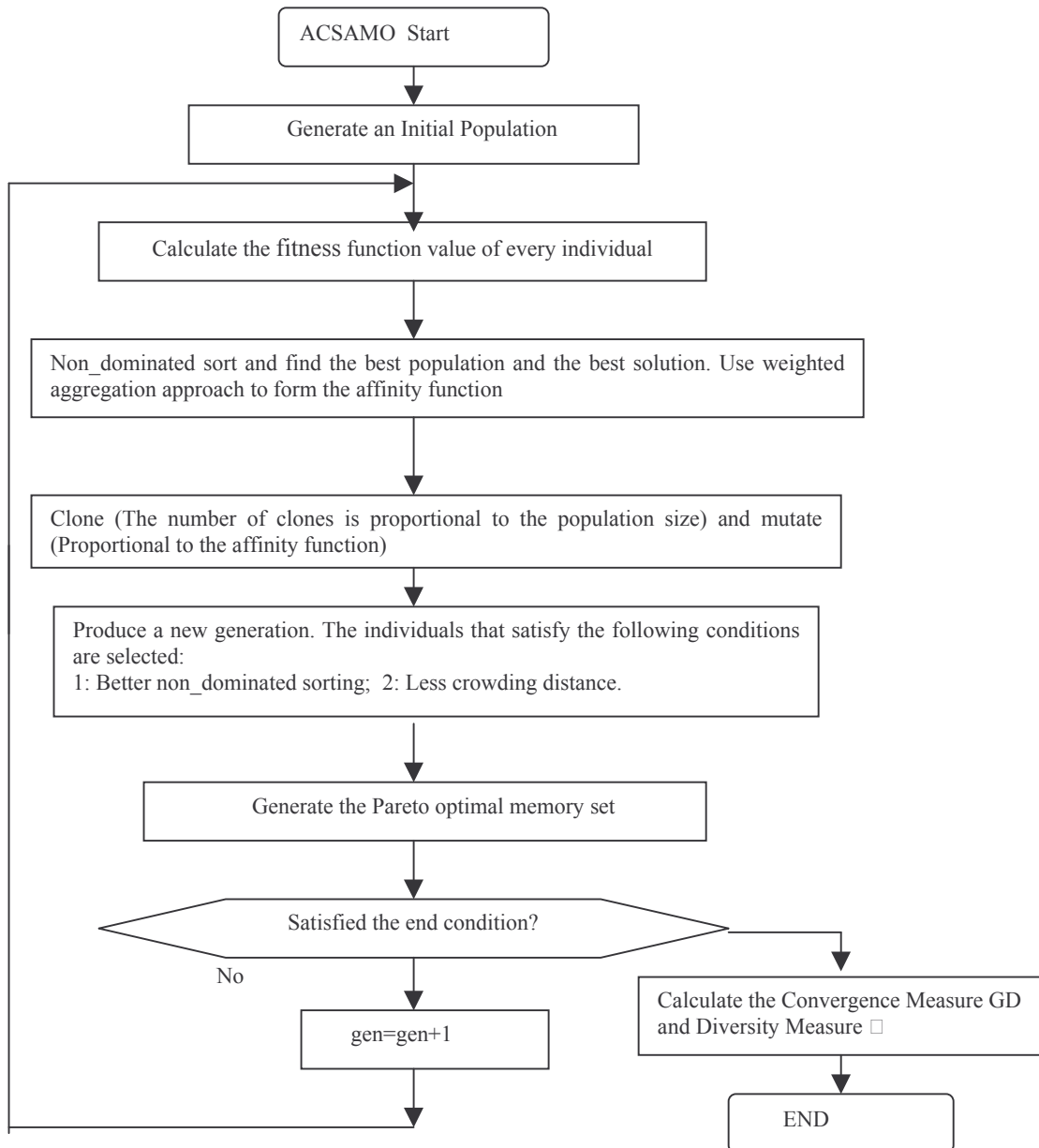
If Pfront size  $>N$ , then calculate the crowding distance among individuals in this Pareto Front and sort

the individuals according ascending distance, Discard the abundant individuals with smaller crowding distance to form the next new population NEXTp. If Pfront size  $<N$ , then fill the NEXTp with the solutions of the second non-dominated front, followed by the third non-dominated front, and so on, until the NEXTp size  $\geq N$ . After then, if NEXTp size  $>N$ , then sort the individuals in the last Pareto Front and delete some with smaller crowding distance.

3) Form the Pareto optimal memory set:

Select the solutions in Pareto front 1 to form memory set. It is assumed that the maximizing size MN of memory is N and let the size of memory to change along the generation. If the size of Pareto front 1 exceeds the N, then, crowding distance sorting is used in this set and MN-N individuals with smaller distance are discard.

In the light of the above considerations, the proposed algorithm can be summarized by the chart of figure 1.



**Fig.1.** Flow chart for the multi-objective optimization algorithm using clonal selection principle

### 3. The application of ACSAMO to solve multiobjective optimization problems

To compare the performance of the ACSAMO to other recently developed algorithms, such as SPEA, NSGA-II, two performance metrics, namely the Generational Distance (GD) and the Spread  $\Delta$ , which were described in [6] were used. GD measures the closeness of the obtained Pareto solution set Q from a known set of the Pareto-optimal set  $P^*$ , which is defined as follows:

$$GD = \frac{\left( \sum_{i=1}^{|Q|} d_i^m \right)^{\frac{1}{m}}}{|Q|} \quad (11)$$

For a two-objective problem ( $m=2$ ),  $d_i$  is the Euclidean distance between the solution  $i \in Q$  and the nearest member of  $P^*$ . A set of  $|P^*|=500$  uniformly distributed Pareto-optimal solutions is used to calculate the closeness metric GD in our simulation experiment.

The Spread metric  $\Delta$  measures the diversity of the solutions along the Pareto-optimal front in the final population and is defined as follows:

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q| \bar{d}} \quad (12)$$

Where  $d_i$  is the distance between the neighboring solutions in the Pareto solution set Q.  $\bar{d}$  is the mean value of all  $d_i$ .  $d_m^e$  is the distance between the extreme solutions of  $P^*$  and Q along the  $m$ th objective. It is worth noting that for an ideal distribution of the solutions (uniform distribution),  $\Delta=0$ .

In order to demonstrate the effectiveness of the proposed ACSAMO algorithm, we used a set of commonly recognized test functions ZDT1-ZDT4 [5]. The multiobjective optimization test function ZDT1-ZDT4 have two objective function which are to be minimized:

$$\begin{aligned} \text{Minimize: } & f_1(\mathbf{x}) \\ \text{Minimize: } & f_2(\mathbf{x}) = g(\mathbf{x}) * h(f_1(\mathbf{x}), g(\mathbf{x})) \end{aligned} \quad (13)$$

The functions  $f_1(\mathbf{x})$ , and  $f_2(\mathbf{x}) = g(\mathbf{x}) * h(f_1(\mathbf{x}), g(\mathbf{x}))$  were defined as follows:

$$\begin{aligned} ZDT_1 : \\ f_1(x) &= x_1 \\ g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ g(f_1, g) &= 1 - \sqrt{\frac{f_1}{g}} \end{aligned} \quad (14)$$

ZDT<sub>2</sub> :

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ g(f_1, g) &= 1 - \left( \frac{f_1}{g} \right)^2 \end{aligned} \quad (15)$$

ZDT<sub>3</sub> :

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ g(f_1, g) &= 1 - \sqrt{\frac{f_1}{g} - \frac{f_1}{g} * \sin(10\pi f_1)} \end{aligned} \quad (16)$$

ZDT<sub>4</sub> :

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + 10(n-1) \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \\ g(f_1, g) &= 1 - \sqrt{\frac{f_1}{g}} \end{aligned} \quad (17)$$

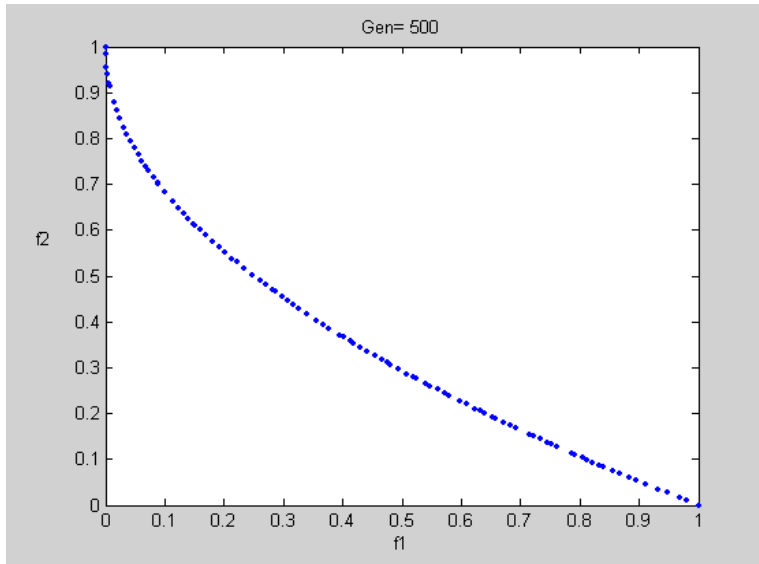
The function ZDT1 has a convex Pareto front, while ZDT2 has a concave Pareto front. Discreteness in the Pareto front for ZDT3 causes difficulties in finding a diverse set of solutions. ZDT4 is a multi-modal problem. There exist  $21^9$  local Pareto-optimal solutions with 10 decision variables, thereby making a total of 100 distinct Pareto-optimal fronts in the objective space, of which only one is the global. This features cause difficulties for many algorithms to converge to the true Pareto-optimal front. For all test functions, the proposed ACSAMO algorithm performed very well and converged to the Pareto-optimal with a high accuracy while maintaining a good diversity among the Pareto solutions.

The algorithm used the following parameters settings:

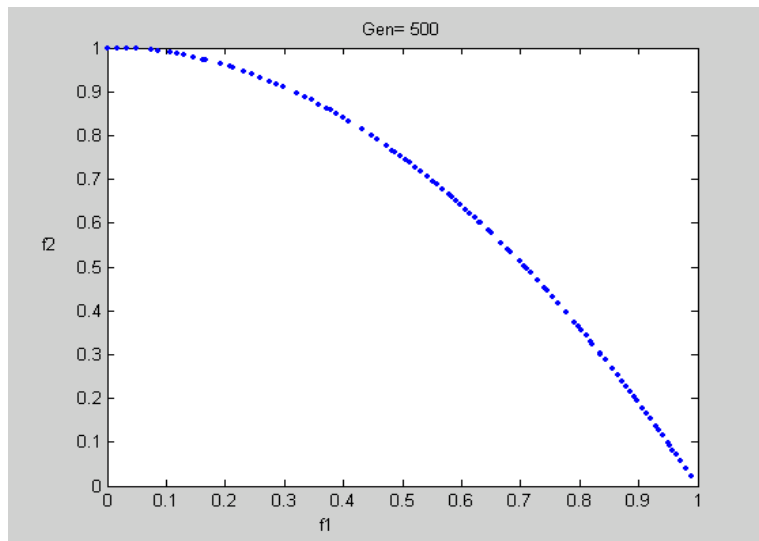
- For ZDT1-ZDT4, N=100, Gen=500(the number of iteration), d=10,  $\beta=2\%$ ,  $\gamma=10$ .
- For ZDT1, ZDT2 and ZDT4,  $p_m=0.1$ .
- For ZDT3,  $p_m=0.6$ .

The Pareto solutions found by ACSAMO after a single run are shown in Fig.2- Fig.5 respectively. It can be seen that the algorithm possesses very good convergence properties while maintaining a good diversity among the Pareto solutions. To compare the ACSAMO to other recently developed algorithms, such as SPEA, NSGA-II, the algorithm was run 10 times independently. The average performance metric value and corresponding variance  $\sigma^2$  are summarized in Table 1 and 2 respectively. In these Tables, the compared results for SPEA, NSGA-II were obtained from [5]. It can be seen that the proposed algorithm performed

very well as far as convergence and diversity are concerned. From Tables 1 and 2, and the Pareto Fronts shown in Fig.1-5, ACSAMO has achieved a better convergence and diversity overall.



**Fig.2.** Pareto optimal solutions of ACSAMO on ZDT1



**Fig.3.** Pareto optimal solutions of ACSAMO on ZDT2

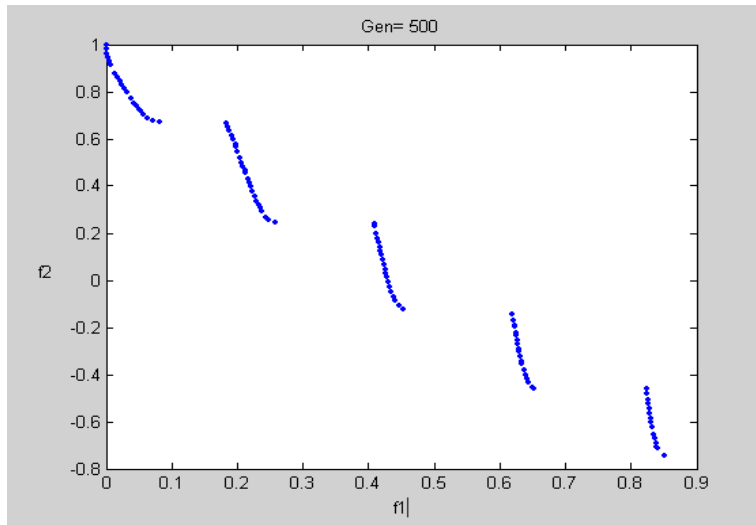


Fig.4. Pareto optimal solutions of ACSAMO on ZDT3

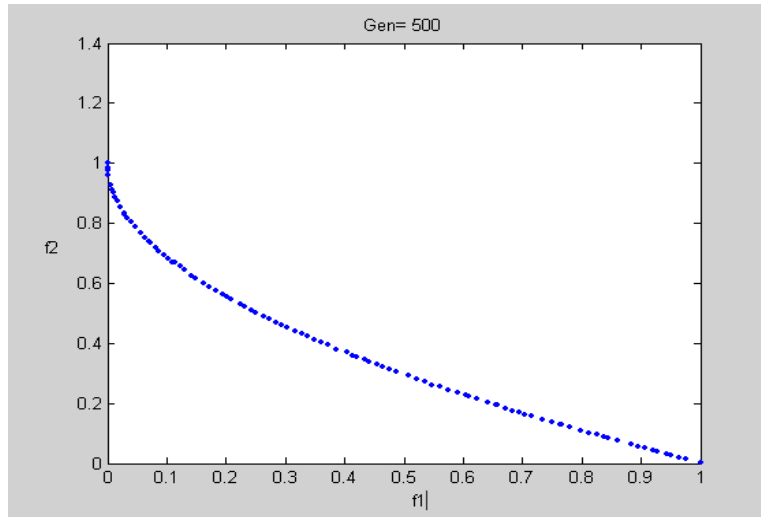


Fig.5. Pareto optimal solutions of ACSAMO on ZDT4

Table1: The mean and variable values of the convergence measure GD

Algorithm	ZDT1		ZDT2		ZDT3		ZDT4	
	<i>GD</i>	$\sigma^2$	<i>GD</i>	$\sigma^2$	<i>GD</i>	$\sigma^2$	<i>GD</i>	$\sigma^2$
<b>SPEA</b>	1.25e-3	0	3.04e-3	2.00e-5	4.42e-2	1.90e-5	9.514	11.32
<b>NSGA II</b>	8.94e-4	0	8.24e-4	0	4.34e-2	4.20e-5	2.92e-2	4.67e-2
<b>ACSAMO</b>	2.32e-4	3.36e-5	9.70e-5	9.30e-6	6.28e-4	2.65e-5	3.00e-4	4.75e-5

**Table2: The mean and variable values of the diversity measure  $\Delta$** 

Algorithm	ZDT1		ZDT2		ZDT3		ZDT4	
	$\Delta$	$\sigma_{\Delta}^2$	$\Delta$	$\sigma_{\Delta}^2$	$\Delta$	$\sigma_{\Delta}^2$	$\Delta$	$\sigma_{\Delta}^2$
<b>SPEA</b>	0.730	9.07e-3	0.678	4.48e-3	0.666	6.66e-4	0.732	1.13e-2
<b>NSGA <math>\square</math></b>	0.463	4.16e-2	0.435	4.46e-2	0.576	5.08e-3	0.655	1.98e-1
<b>ACSAMO</b>	0.3063	0.0206	0.3262	0.012	0.6264	0.0133	0.2958	0.0257

#### 4. Discussions and Conclusions

In this paper, we proposed an algorithm to solve multiobjective optimization using the clonal selection principle. Our approach combined the clonal selection principle and the non-dominated sorting technique. We use the crowding distance as the metric for the solution concentration, we also as let the antigens to be recognized and the size of the Pareto optimal memory set dynamically adjusted with the iterations. In addition, a weighted aggregation function has been introduced to form an affinity function that guides the mutation process more effectively. For comparison purposes, two performance metrics, namely the Generational Distance GD and the Spread  $\Delta$ , were used. Results of experiments using a series of well-known benchmark functions indicated that the proposed algorithm converged to the Pareto optimal solutions with very good accuracy while maintaining a good diversity along the Pareto optimal front when dealing with unconstrained test functions.

#### 5. Acknowledgement

The first author gratefully Acknowledge the support from the Gansu Science and Technology Council via project ZS021-A25-017-G and the China Scholarship Council.

#### 6. References

- [1] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms and their applications. Proceedings of the First International Conference on Genetic Algorithms; 1985. pp. 93–100.
- [2] Goldberg DE. Genetic algorithms in search, optimization, and machine learning, reading. Massachusetts: Addison Wesley; 1989.
- [3] Zitzler E, Theiele L. Multiobjective evolutionary algorithm: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3(4):257–71.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal and T.Meyarivan. A fast and Elitist multiobjective Genetic Algorithm: NSGA- $\square$ . IEEE Transactions on Evolutionary Computation, 6(2):182-197, April 2002.
- [5] K. Deb, Multi-objective optimization using evolutionary algorithms. London: Wiley; 2002.

- [6] Dasgupta, D., et al, Artificial Immune System (AIS) Research in the Last Five Years, CEC-03, 2003.
- [7] Leandro Nunes de Castro, Fernando José Von Zuben. Technical Report: Artificial Immune Systems: Part I – Basic Theory and Application. December, 1999.
- [8] C. Berek and M. Ziegner,, The maturation of the immune response, *Immun. Today*, vol. 14, no. 8, pp. 400–402, 1993.
- [9] Dasgupta, D.(ed.) Artificial Immune Systems and Their Applications, Springer-Verlag,1999.
- [10] J. Timmis<sup>1</sup>, T. Knight, L.N. de Castro and E. Hart, An Overview of Artificial Immune Systems.2003.
- [11] Leandro N. de Castro and Fernando J. Von Zuben. aiNet: An artificial Immune Network for data Analysis, in *Data Mining: A Heuristic Approach*,2001.
- [12] Leandro N. de Castro and Fernando J. Von Zuben. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transaction on Evolutionary Computation*. VOL. 6, NO. 3,pp239-251 JUNE 2002.
- [13] M.Mahfouf, Adaptive Weighted Particle Swarm Optimization for multi-objective Optimal Design of Alloy Steels. In *Proc.Parallel Problem Solving from Nature-PPSN*, 8<sup>th</sup> International Conference, Birmingham, September 2004,pp.762-771.