



Cairo University  
**Egyptian Informatics Journal**

www.elsevier.com/locate/eij  
www.sciencedirect.com



ORIGINAL ARTICLE

# Artificial immune system based neural networks for solving multi-objective programming problems

Wael F. Abd El-Wahed <sup>a</sup>, Elsayed M. Zaki <sup>b</sup>, Adel M. El-Refaey <sup>b,\*</sup>

<sup>a</sup> Faculty of Computers & Information, Menoufia University, Egypt

<sup>b</sup> Department of Basic Engineering Science, Faculty of Engineering, Shebin El-Kom, Menoufia University, Egypt

Received 6 June 2009; accepted 27 September 2010

Available online 29 October 2010

## KEYWORDS

Artificial immune system;  
Neural networks;  
Nonlinear programming;  
Multi-objective programming;  
Clonal selection

**Abstract** In this paper, a hybrid artificial intelligent approach based on the clonal selection principle of artificial immune system (AIS) and neural networks is proposed to solve multi-objective programming problems. Due to the sensitivity to the initial values of initial population of antibodies (Ab's), neural networks is used to initialize the boundary of the antibodies for AIS to guarantee that all the initial population of Ab's is feasible. The proposed approach uses dominance principle and feasibility to identify solutions that deserve to be cloned, and uses two types of mutation: uniform mutation is applied to the clones produced and non-uniform mutation is applied to the “not so good” antibodies. A secondary (or external) population that stores the nondominated solutions found along the search process is used. Such secondary population constitutes the elitist mechanism of our approach and it allows it to move towards the Pareto front.

© 2010 Faculty of Computers and Information, Cairo University. Production and hosting by Elsevier B.V. All rights reserved.

## 1. Introduction

The field of immunological computation (IC) or artificial immune system (AIS) has been evolving steadily [10] since 1985. In recent years, several researchers have developed computational models of the immune system that attempt to capture some of its most remarkable features such as its self-organizing capability [4].

The multi-objective immune system algorithm (MISA) can be considered as the first real proposal of MOAIS in the literature [2]. In the first proposal of the algorithm, authors attempted to follow the clonal selection principle very closely, then the algorithm performances have been improved in a successive version [4] sacrificing some of the biological metaphor. The population is encoded by binary strings and it is initialized randomly. The algorithm does not use explicitly a scalar index

\* Corresponding author.

E-mail address: adel\_elrefaey@yahoo.com (A.M. El-Refaey).

1110-8665 © 2010 Faculty of Computers and Information, Cairo University. Production and hosting by Elsevier B.V. All rights reserved.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

doi:10.1016/j.eij.2010.10.002



Production and hosting by Elsevier

to define the avidity of a solution but some rules are defined for choosing the set of antibodies to be cloned. The ranking scheme uses the following criteria: (1) first feasible and non-dominated individuals, then (2) infeasible nondominated individuals, finally (3) infeasible and dominated. The memory set (called secondary population) is updated by the nondominated feasible individuals. Because of this repository being limited in size, an adaptive grid is implemented to enforce a uniform distribution of nondominated solutions [4,10].

The number of clones for the pool of best antibodies depends on the antibody–antibody affinity. These best antibodies are selected for a uniform mutation, with mutation probability proportional to antibody–antigen affinity, according to the ranking scheme, while the remaining population undergoes a non-uniform mutation. Again, the ranking scheme is used as criterion to reduce the population to its original cardinality [11].

Neural network (NN) is a well known as one of powerful computing tools to solve optimization problems. Due to massive computing unit neurons and parallel mechanism of neural network approach it can solve the large-scale problem efficiently and optimal solution can be obtained [13,16].

The other hand neural network (NN) approach is attended as a new method for solving optimization problems, this method has a great charm because NN can solve large scale and complex optimization problems in real time, and also is benefit to search the global solution.

A general methodology for solving multi-objective nonlinear programming (MONP) problems. In order to operationalize the concept of Pareto-optimal solution, we should relate it to a familiar concept. The most common strategy is to characterize Pareto-optimal solutions in terms of optimal solutions of appropriate nonlinear programming problems (NLPP). Among weighted aggregation (WA) technique we can characterize multi-objective programming problems (MOPs) into NLPPs [13,18].

We run neural networks based on weighted aggregation method, with weights to determine the end points of the Pareto front and the point that all objective functions has equal weight. From these three point we deduce the range (the upper and lower values) of each decision variable. This range used as the input to AIS this modification makes AIS faster, and give more accurate Pareto-optimal solutions.

## 2. Artificial immune systems

The natural immune system (NIS) has an amazing pattern matching ability, used to distinguish between foreign cells entering the body (referred to as non-self, or antigen (Ag)) and the cells belonging to the body (referred to as self). As the NIS encounters antigen, the adaptive nature of the NIS is exhibited, with the NIS memorizing the structure of these antigen for faster future response the antigen [4].

In NIS research, four models of the NIS can be found:

- *The classical view* of the immune system is that the immune system distinguishes between self and non-self, using lymphocytes produced in the lymphoid organs. These lymphocytes “learn” to bind to antigen.
- *Clonal selection theory*, where an active B-cell produces antibodies through a cloning process. The produced clones are also mutated.

- *Danger theory*, where the immune system has the ability to distinguish between dangerous and non-dangerous antigen.
- *Network theory*, where it is assumed that B-cells form a network. When a B-cell responds to an antigen, that B-cell becomes activated and stimulates all other B-cells to which it is connected in the network [12].

An artificial immune system (AIS) models some of the aspects of a NIS, and is mainly applied to solve pattern recognition problems, to perform classification tasks, and to cluster data. One of the main application areas of AISs is in anomaly detection, such as fraud detection, and computer virus detection [4,10].

### 2.1. Clonal selection theory

Any molecule that can be recognized by the adaptive immune system is known as an Ag. When an animal is exposed to an Ag, some subpopulation of its bone-marrow-derived cells (B-lymphocytes) responds by producing Ab's. Ab's are molecules 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 (divide) and mature into terminal (nondividing) Ab secreting cells, called plasma cells. The process of cell division (mitosis) generates a clone, i.e., a cell or set of cells that are the progenies of a single cell. B-cells, in addition to proliferating and differentiating into plasma cells, can differentiate into long-lived B-memory cells. Memory cells circulate through the blood, lymph, and tissues and, when exposed to a second antigenic stimulus, commence to differentiate into plasma cells capable of producing high-affinity Ab's, preselected for the specific Ag that had stimulated the primary response. Fig. 1 depicts the clonal selection principle.

The main features of the clonal selection theory [2,10] that will be explored in this paper are:

- (1) Proliferation and differentiation on stimulation of cells with Ag's.
- (2) Generation of new random genetic changes, expressed subsequently as diverse Ab patterns, by a form of accelerated somatic mutation (a process called affinity maturation).
- (3) Estimation of newly differentiated lymphocytes carrying low-affinity antigenic receptors.

## 3. Multi-objective programming (MOP) problem

This section provides the necessary mathematical background for MOP [8,17]. Consider a multi-objective programming problem with  $k$  objectives ( $f_i(x)$ ), ( $j = 1, 2, \dots, k$ ) and  $n$  decision variables ( $x_i = 1, 2, \dots, n$ ):

*MOP:*

$$\text{Min } F(x) = (f_1(x), \dots, f_k(x))$$

Subject to

$$S = \{x \in R^n | g(x) \geq 0, h(x) = 0\} \quad (1)$$

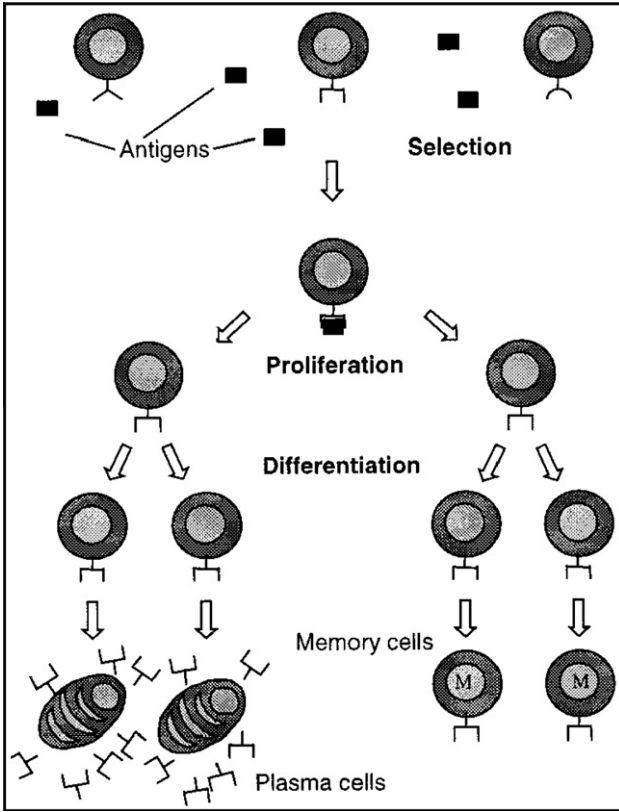


Figure 1 Clonal selection principle.

where  $x \in R^n, f: R^n \rightarrow R^k$ , is  $k$ -dimensional vector valued continuous functions of  $n$  variables,  $g = [g_1, \dots, g_m]^T: R^n \rightarrow R^m$ , is  $m$ -dimensional vector valued continuous functions of  $n$  variables and  $h = [h_1, \dots, h_p]^T: R^n \rightarrow R^p$ , is  $p$ -dimensional vector valued continuous functions of  $n$  variables.

The  $k$  objectives are conflicted with each other. Therefore, the target of MOP is to achieve a set of efficient solutions that are called Pareto set. The related concepts of *Pareto-optimal solution*, and *weak Pareto-optimal solution* [17].

**Definition 1** (*Pareto-optimal solution*).  $x^*$  is said to be Pareto-optimal solution of MOP If there is no other feasible  $x$  such that, for all  $j, j = 1, 2, \dots, k$  with strict inequality for at least one  $j$ .

**Definition 2** (*Weak Pareto-optimal solution*).  $x^* \in S$  is said to be a weak Pareto-optimal solution if and only if there is no other  $x \in S$  such that  $f_j(x^*) < f_j(x)$  for all  $j, j = 1, 2, \dots, k$ .

### 3.1. Weighted method for MOP problem

The weighted method [8] for the multi-objective optimization problem is formulated as:

$$\begin{aligned}
 P(w): \quad & \text{Min} \quad \sum_{i=1}^k w_i f_i(x) \\
 \text{s.t.} \quad & x \in S, \quad w \in W, \\
 & W = \left\{ w \in R^k \mid w_j \geq 0, \sum_{j=1}^k w_j = 1 \right\}
 \end{aligned} \tag{2}$$

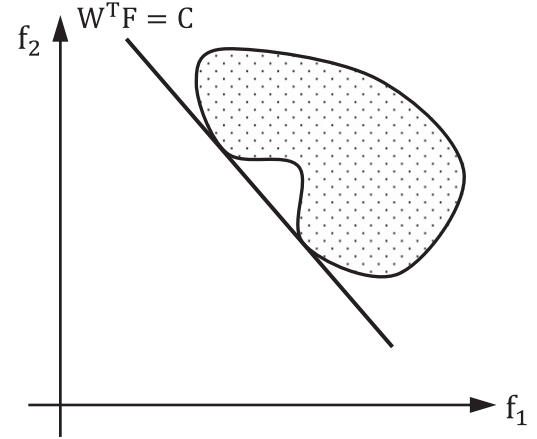


Figure 2 Geometrical representation of weighted method.

Multi-objective optimization runs are conducted with different weighting vector ( $W$ ) in order to locate a set of points on the Pareto front. This method is the simplest and the most straight forward way of obtaining the Pareto-optimal front. However, this method is associated with some major drawbacks. Depending on the scaling of the different objectives and the shape of the Pareto front, it is hard to select the weighting. Another problem occurs when the solution space is non-convex. In that case not all the Pareto-optimal solutions can be obtained by solving the problem  $P(w)$ . But in our study we concentrate on the Convex MOP problems to avoid this weakness of  $P(w)$  [6].

The solution of the single objective  $P(w)$  Eq. (2) (in a minimizing case) is shown in Fig. 2, where the shaded area denotes the feasible region.

**Theorem 1.** If  $x^* \in S$  is an optimal solution of the weighting problem  $P(w)$  where either  $W > 0$ , or  $x^*$  is a unique optimal solution, then  $x^*$  is a Pareto-optimal solution of the MOP [8].

**Theorem 2.** Let the multi-objective optimization problem be convex. If  $x^* \in S$  is an efficient solution of the MOP, then  $x^*$  is an optimal solution of the weighting problem  $P(w)$  for some  $W = (w_1, w_2, \dots, w_k) \geq 0$  [8].

### 4. The multi-objective optimization neural network

To formulate the optimization problem in terms of a neural network, the key step is to construct an appropriate energy function  $E(z)$  such that the lowest energy state corresponds to the intended optimal solution  $z^*$ . Based on the energy function, we construct a gradient system of differential equations which corresponds to a *neural network* [9,18].

The (MOP) is transformed via the weighted approach into single nonlinear programming problem:

NLPP:

$$\begin{aligned}
 \text{Min} \quad & F(x) = \sum_{j=1}^k w_j f_j(x) \\
 \text{s.t.} \quad & g_i(x) \geq 0, \quad i = 1, 2, \dots, m \\
 & h_l = a_l^T x - b_l, \quad l = 1, 2, \dots, p
 \end{aligned} \tag{3}$$

$$\text{where } W = \left\{ w \in R^k \mid w_i \geq 0, \sum_{i=1}^k w_i = 1 \right\}$$

where  $f_j(x), j = 1, 2, \dots, k$  and  $g_i(x) \in C^2$  (the Set of all continuous function which possess the second derivatives) ( $i = 1, 2, \dots, m$ ) are convex functions, and the vectors  $\{a_i\}_{i=1,2,\dots,p}$  are linearly independent. We suppose throughout that NLPP has an optimal solution, and is strongly consistent, i.e., there exists  $x^0 \in R^n$  such that  $g_i(x^0) > 0$ , ( $i = 1, 2, \dots, m$ ),  $h_j(x^0) = 0, (j = 1, 2, \dots, p)$  [7].

According to the result in Rao [15], the dual nonlinear programming problem (DNPP) can be formulated as follows:

*DNLPP:*

$$\begin{aligned} \text{Max}_{x, \lambda, \mu} \quad & L(x, \lambda, \mu), \quad x \in R^n \\ \text{s.t.} \quad & \nabla L(x, \lambda, \mu) = 0 \\ & \lambda \geq 0, \quad \mu \text{ unrestricted in sign,} \end{aligned} \quad (4)$$

where

$$\begin{aligned} \lambda &= (\lambda_1, \lambda_2, \dots, \lambda_m)^T, \quad \mu = (\mu_1, \mu_2, \dots, \mu_p)^T \\ L(x, \lambda, \mu) &= F(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{j=1}^p \mu_j h_j(x) \equiv L(z) \\ \nabla L(x, \lambda, \mu) &= \nabla F(x) - \sum_{i=1}^m \lambda_i \nabla g_i(x) - \sum_{j=1}^p \mu_j \nabla h_j(x) \end{aligned}$$

**Theorem 3.** Suppose that  $x^*$  is an optimal solution of convex nonlinear programming problem (CNLPP), then there exists  $(\lambda^*, \mu^*)^T \in R^{m+p}$ , such that  $(x^*, \lambda^*, \mu^*)^T$  is an optimal solution of dual problem for a specific value of  $W$  [15].

To entertain general inequality constraints, we can easily generalize Lemma 1 in Leung et al. [9] and obtain the following theorems:

**Theorem 4.** Suppose  $g(x) \in R^m, g(x) \in C^1$  (the set of continuously differentiable function). Then

$$g(x) \geq 0 \iff \frac{1}{2} g(x)^T (g(x) - |g(x)|) = 0 \quad (5)$$

and  $\frac{1}{2} g(x)^T (g(x) - |g(x)|) \in C^1$ , where  $|x| = (|x_1|, |x_2|, \dots, |x_n|)^T$  [7].

By Theorem 4, general inequality constraints can easily be transformed into equality constraints, and after the transformation, differentiability of the function is the same as that of  $g(x)$ . It is easy to prove that  $\frac{1}{2} g(x)^T (g(x) - |g(x)|)$  is a convex function when  $g(x)$  is a concave function.

**Theorem 5.**  $x^*$  and  $(x^*, \lambda^*, \mu^*)^T$  are optimal solutions of NLPP and DNLPP, respectively, for a specific value of  $W$ , if and only if

$$\begin{aligned} F(x^*) &= L(x^*, \lambda^*, \mu^*), \quad \nabla L(x^*, \lambda^*, \mu^*) = 0 \\ g(x^*) &\geq 0, \quad h(x^*) = 0, \quad \lambda^* \geq 0, \mu^* \text{ unrestricted in sign} \\ \lambda^{*T} g(x^*) &= 0 \end{aligned} \quad (6)$$

and the set of vectors  $\nabla h_i(x^*)$  are linearly independent.

**Proof.** By the duality theorem of the nonlinear programming [15] and Kuhn Tucker optimality condition of convex nonlin-

ear programming [15], Theorem 3 can be immediately obtained.

By Theorems 4 and 5, the energy function of Convex NLP can be constructed as follows:

$$\begin{aligned} E(z) &= E(x, \lambda, \mu) = \frac{1}{2} [\lambda^T g(x)]^2 + \frac{1}{2} \|\nabla_x L(x, \lambda, \mu)\|^2 \\ &\quad + \frac{1}{2} g(x)^T (g(x) - |g(x)|) + \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2} \lambda^T (\lambda - |\lambda|) \end{aligned} \quad (7)$$

where  $z = (x^T, \lambda^T, \mu^T)^T \in R^{n+m+p}$ . Every term of the right-hand side of Eq. (7) being zero corresponds to every equality or inequality being satisfied in Eq. (6) and  $E(z) \geq 0$ . Thus  $E(z)$  is differentiable function by Leung et al. [9].  $\square$

**Theorem 6.**  $z^* = (x^*, \lambda^*, \mu^*)^T$  is zero point of  $E(z) \iff z^*$  is an optimal solution of NLPP and DNLPP (i.e.,  $x^*$  and  $(x^*, \lambda^*, \mu^*)^T$  are optimal solutions of NLPP and DNLPP for a specific value of  $W$ , respectively).

**Proof.** By Theorem 4,  $z^* = (x^*, \lambda^*, \mu^*)^T$  is zero point of  $E(z) \iff z^*$  satisfies Eq. (6). By Theorem 5,  $z^*$  satisfies Eq. (6)  $\iff x^*$  and  $(x^*, \lambda^*, \mu^*)^T$  are Pareto-optimal solutions of NLPP and DNLPP for a specific value of  $W$ , respectively.

Therefore, Theorem 6 is proved.  $\square$

Employing the unified idea in Leung et al. [9], we can use the gradient system to construct the following multi-objective neural network for solving a convex multi-objective optimization problems:

$$\frac{dz}{dt} = -\nabla E(z) \quad (8)$$

Using the formula of computing gradients in Leung et al. [9], the multi-objective neural network as in Eq. (8) can be written in details as follows:

$$\begin{aligned} \frac{dx}{dt} &= -\nabla_x E(z) = -\lambda^T g(x) \cdot \nabla g(x)^T \lambda - \nabla g(x)^T (g(x) - |g(x)|) \\ &\quad - \nabla_{xx}^2 L(z) \nabla_x L(z) - A^T (Ax - b) \\ \frac{d\lambda}{dt} &= -\nabla_\lambda E(z) = -\lambda^T g(x) \cdot g(x) + \nabla g(x) \nabla_x L(z) - (\lambda - |\lambda|) \\ \frac{d\mu}{dt} &= -\nabla_\mu E(z) = A \nabla_x L(z) \end{aligned} \quad (8')$$

where  $\nabla_{xx}^2 L(z)$  is the Hessian matrix of the function  $L(z)$  with respect to  $x$ . Suppose  $\nabla E(z)$  is Lipschitz continuous, then the initial value problem of differential equations in Eq. (8) has a unique solution because the function in the right hand side of differential Eq. (8) or (8') are continuous, these equations can easily be achieved by hardware implementation of the network. Therefore, it is a feasible neural network [2,7].

## 5. The proposed approach

The algorithm run in two stages, the first one run neural networks with a random initial input based on the weighted method with three points which is the end points and the midpoint of weights. The second stage uses the output of neural networks

as the input to AIS which has taken ideas from the *clonal selection principle*, [1,4] modeling the fact that only the highest affinity antibodies will proliferate. Antibodies, in our case, are represented by binary strings that encode the decision variables of the problem to be solved. However, we do not use a population of antigens, but only Pareto dominance and feasibility to identify solutions that deserve to be cloned. Additionally, our approach uses mutation [4] (uniform mutation is applied to the clones and non-uniform mutation is applied to the “not so good” antibodies, as we will see later on). We also use a secondary (or external) population that stores the nondominated solutions found along the search process. Such secondary population is the elitist mechanism most commonly adopted in multi-objective optimization, and it allows us to move towards the Pareto front [3].

### 5.1. The algorithm

The proposed algorithm for solving multi-objective immune system algorithm (MISA) based on NN is as follow:

*Neural network simulation algorithm* [13]

[Step 1] Initialization Let  $t = 0$ . Randomly choose initial vector  $x(t) \in R^n, \lambda(t) \in R^m, \mu(t) \in R^p, \Delta t > 0$  (for example,  $\Delta t = 0.0001$ ) and error  $\varepsilon = 10^{-9}$ .

[Step 2] Transform the of MOP into NLPP Eq. (3)

[Step 3] Computation of gradient:

$$u(t) = \nabla_x E(z) = \lambda^T g(x) \cdot g(x)^T \lambda + \nabla g(x)^T [g(x) - |g(x)|] \\ + \nabla_{xx}^2 L(z) \nabla_x L(z) + A^T (Ax - b)$$

$$v(t) = \nabla_\lambda E(z) = \lambda^T g(x) \cdot g(x) - \nabla g(x) \nabla_x L(z) + [\lambda - |\lambda|]$$

$$w(t) = \nabla_\mu E(z) = -A \nabla_x L(z)$$

[Step 4] States updating:

$$x(t + \Delta t) = x(t) - \Delta t \cdot u(t) \quad \lambda(t + \Delta t) \\ = \lambda(t) - \Delta t \cdot v(t) \quad \mu(t + \Delta t) = \mu(t) - \Delta t \cdot w(t)$$

[Step 5] Calculate:

$$s = \sum_{i=1}^n u_i^2(t), r = \sum_{j=1}^m v_j^2(t), q = \sum_{j=1}^p w_j^2(t)$$

[Step 6] Stopping rule:

if  $s < \varepsilon, r < \varepsilon$  and  $q < \varepsilon$ , then output  $x(t + \Delta t), \lambda(t + \Delta t), \mu(t + \Delta t)$  into the input file of MISA; otherwise let  $t = t + \Delta t$  and go to Step 3.

*MISA simulation algorithm*

[Step 1] Initialization based on NN output.

[Step 2] Sorting population according to dominance.

[Step 3] Choose the “best” antibodies to be cloned.

[Step 4] Cloning “best” antibodies.

[Step 5] Applying a uniform mutation to the clones.

[Step 6] Applying a non uniform mutation to the “not so good” antibodies.

[Step 7] Returns the population size to its original value.

[Step 8] Repeat this process from Step 2 during a certain (pre-determined) number of times.

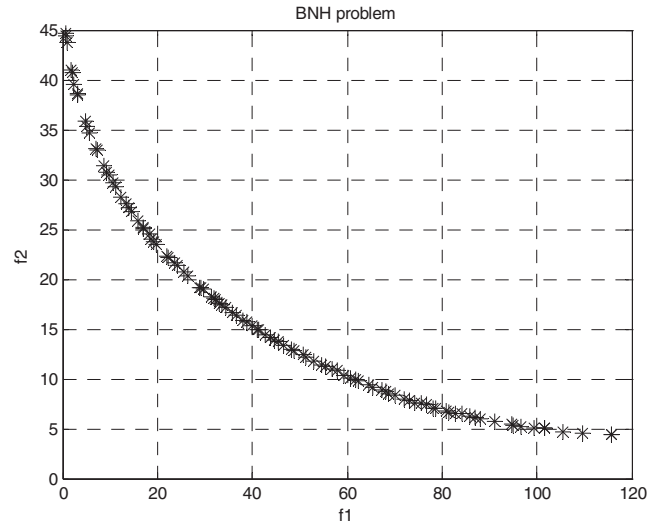


Figure 3 Pareto front of BNH problem using MISA based NN.

## 6. Experiments

In order to validate our approach, five benchmark functions which reported in the standard evolutionary multi-objective optimization literature [5,14].

The results indicated below when using the following parameters for MISA: population size = 100, number of grid subdivisions = 25, size of the external population = 100.

The problems chosen from the benchmark domains are BNH used by Binh and Korn, SRN used by Srinivas, Deb [14], M\_OU by Cvetkovic, M\_LOC by Kita, M\_3OU by Viennet [5].

### Problem 1 (BNH).

$$\begin{aligned} \text{Min} \quad & f_1(x) = 4x_1^2 + 4x_2^2 \\ & f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 \\ \text{s.t.} \quad & g_1(x) = (x_1 - 5)^2 + x_2^2 \leq 25 \\ & g_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7 \end{aligned}$$

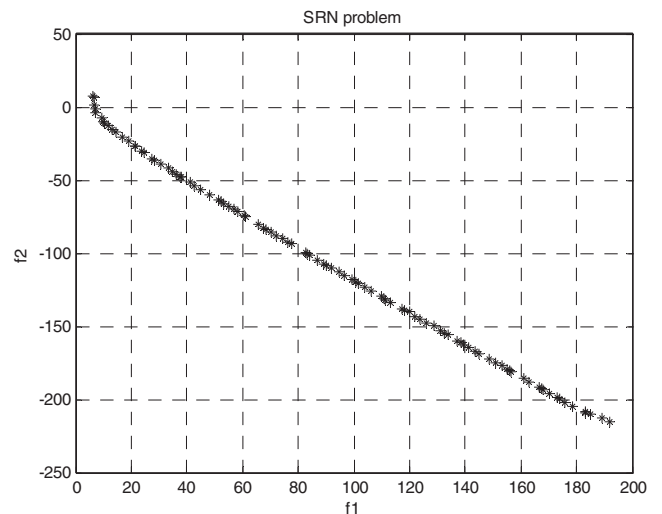
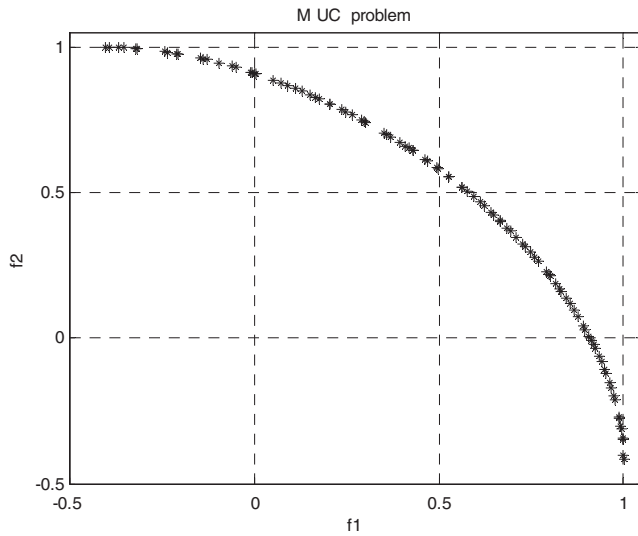
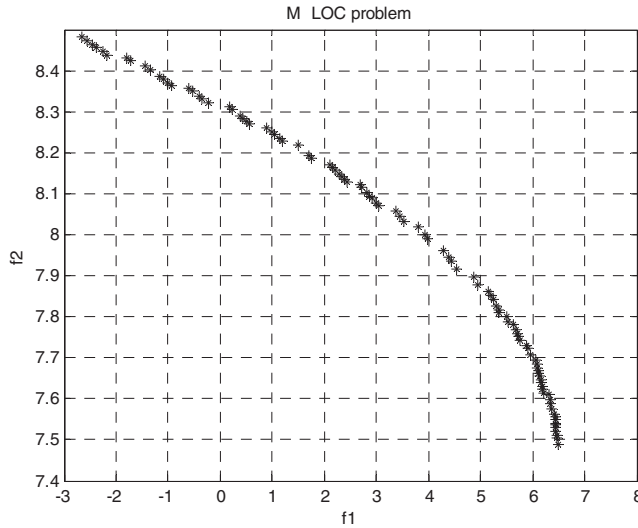


Figure 4 Pareto front of SRN problem using MISA based NN.





**Figure 5** Pareto front of M\_UC problem using MISA based NN.



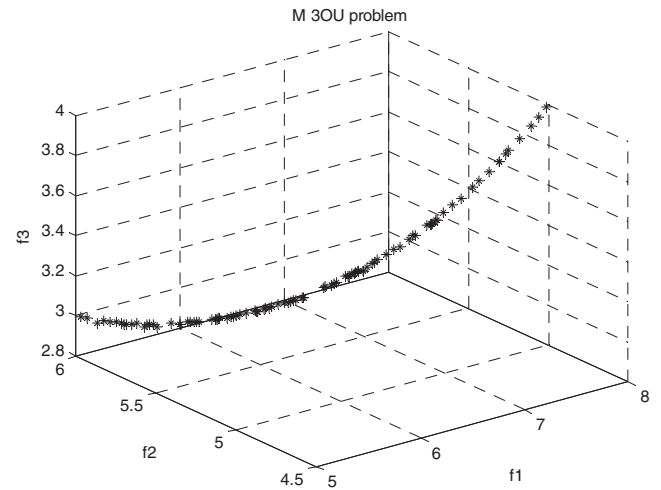
**Figure 6** Pareto front of M\_LOC problem using MISA based NN.

The output of NN indicates that the range of variables will  $0 \leq x_1 \leq 5$  and  $0 \leq x_2 \leq 3$ . Pareto front using MISA based NN shown in Fig. 3.

**Problem 2 (SRN).**

$$\begin{aligned} \text{Min} \quad & f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2 \\ & f_2(x) = 9x_1 - (x_2 - 1)^2 \\ \text{s.t.} \quad & g_1(x) = x_1^2 + x_2^2 \leq 225 \\ & g_2(x) = -x_1 + 3x_2 \geq 10 \end{aligned}$$

The output of NN indicates that the range of variables will  $-12 \leq x_1 \leq 12$  and  $-10 \leq x_2 \leq 10$ . Pareto front using MISA based NN shown in Fig. 4.



**Figure 7** Pareto front of M\_3OU problem using MISA based NN.

**Problem 3 (M\_UC).**

$$\begin{aligned} \text{Max} \quad & f_1(x) = \sin(x_1^2 + x_2^2 - 1) \\ & f_2(x) = \sin(x_1^2 + x_2^2 + 1) \\ \text{s.t.} \quad & 0 \leq x_1 \leq 3\pi/4 \\ & 0 \leq x_2 \leq 3\pi/4 \end{aligned}$$

The output of NN indicates that the range of variables will  $0 \leq x_1 \leq 2$  and  $0 \leq x_2 \leq 2$ . Pareto front using MISA based NN shown in Fig. 5.

**Problem 4 (M\_LOC).**

$$\begin{aligned} \text{Max} \quad & f_1(x) = -x_1^2 + x_2 \\ & f_2(x) = x_1/2 + x_2 + 1 \\ \text{s.t.} \quad & g_1(x) = x_1/6 + x_2 \leq 13/2 \\ & g_2(x) = x_1/2 + x_2 \leq 15/2 \\ & g_3(x) = 5x_1 + x_2 \leq 30 \end{aligned}$$

The output of NN indicates that the range of variables will  $0 \leq x_1 \leq 5$  and  $3 \leq x_2 \leq 7$ . Pareto front using MISA based NN shown in Fig. 6.

**Problem 5 (M\_3OU).**

$$\begin{aligned} \text{Min} \quad & f_1(x) = x_1^2 + (x_2 - 1)^2 \\ & f_2(x) = x_1^2 + (x_2 + 1)^2 + 1 \\ & f_3(x) = (x_1 - 1)^2 + x_2^2 + 2 \\ \text{s.t.} \quad & -2 \leq x_1 \leq 2 \quad -2 \leq x_2 \leq 2 \end{aligned}$$

The output of NN indicates that the range of variables will  $0 \leq x_1 \leq 2$  and  $-1 \leq x_2 \leq 2$ . Pareto front using MISA based NN shown in Fig. 7.

**7. Conclusion**

We have presented a hybrid multi-objective optimization algorithm based on the clonal selection principle and neural net-

works. The approach is able to produce results similar or better than those generated by other evolutionary algorithms after determining the max and min values with NN and use it to initialize population with at least feasible antibodies which help MISA to find the Pareto-optimal solution more accurate and more faster. Our approach uses an affinity measure to control the amount of mutation to be applied to the antibodies. Affinity in this case, is defined in terms of nondominance and feasibility. This affinity measure, combined with the secondary population are used to distribute nondominated solutions in a uniform way.

The approach proposed also uses a very simple mechanism to deal with constrained test functions, and our results indicate that such mechanism, despite its simplicity, is effective in practice.

All calculations are carried by *Matlab* 7.2 program for NN and *VC++* 2008 for MISA, and are run on Laptop 2 GHz/1 Gb RAM/Windows XP, the solution take small number of iterations.

## References

- [1] Coello Coello CA. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 2002;191(11/12): 1245–87.
- [2] Coello Coello CA, Cruz Cort'es N. An approach to solve multiobjective optimization problems based on an artificial immune system. In: Timmis J, Bentley PJ, editors. *First international conference on artificial immune systems (ICARIS'2002)*. UK: University of Kent at Canterbury; September 2002. p. 212–21 [ISBN 1-902671-32-5].
- [3] Coello Coello CA, Van Veldhuizen DA, Lamont GB. *Evolutionary algorithms for solving multiobjective problems*. New York: Kluwer Academic Publishers; 2002 [ISBN 0-3064-6762-3].
- [4] Coello Coello CA, Cruz Cort'es N. Solving multiobjective optimization problems using an artificial immune system. *Genet Program Evol Mach* 2005;6:163–90 [Springer Science + Business Media, Inc.].
- [5] Deepti Chafekar, Jiang Xuan, Khaled Rasheed, Constrained multi-objective optimization using steady state genetic algorithms. In: *Proceedings of the 2003 international conference on Genetic and evolutionary computation: Part I*, Chicago, IL, USA, July 12–16; 2003.
- [6] Flam SD. Solving convex programming by means of ordinary differential equations. *Math Operat Res* 1992;17(2):290–302.
- [7] Chen K-z, Leung Y. A neural network for solving nonlinear programming problem. *Neural Comput Appl* 2002;11:103–11.
- [8] Miettinen Kaisa M. *Nonlinear multiobjective-optimization*. Kluwer Academic Publishers; 2002.
- [9] Leung Y, Chen K-z, Jiao Y-c, Gao X-b, Leung KS. A new gradient-based neural network for solving linear and quadratic programming problems. *IEEE Trans Neural Netw* 2001;12(5):1047–83.
- [10] Nunes de Castro L, Von Zuben FJ. *Artificial immune systems. Part I: Basic theory and applications*, Technical Report TR-DCA 01/99, FEEC/UNICAMP, Brazil; 1999.
- [11] Nunes de Castro L, Von Zuben FJ. *aiNet: an artificial immune network for data analysis. Data mining: a heuristic approach*, Idea Group Publishing, USA; 2001. p. 231–59.
- [12] Abo-Sinna M, El-Refaey Adel M. Neural networks for solving multi-objective non-linear programming problems. *Ain Shams Sci Bull* 2005;40(4):1235–51.
- [13] Filomeno Coelho R. *Multicriteria optimization with expert rules for mechanical design*, Dissertation, Université Libre De Bruxelles Faculté Des Sciences Appliquées; 2004.
- [14] Rao SS. *Engineering optimization: theory and practice*. 3rd ed. New York: Wiley; 1996.
- [15] Rudi Y, Cangpu W. A novel neural network model for nonlinear programming. *Acta Autom Sin* 1996;22(2):293–300.
- [16] Chankong Vira, Haimes Yacov Y. *Multiobjective decision making: theory and methodology*. Elsevier Science Publishing; 1983.
- [17] Xia Y, Wang J, Hung DL. Recurrent neural networks for solving inequalities and equations. *IEEE Trans Circuits Syst* 1999;46(4):452–62.
- [18] Nunes de Castro L, Von Zuben FJ. Learning and optimization using the clonal selection principle. *IEEE Trans Evol Comput* 2002;6(3):239–51.

## Further Reading