

[Egyptian Informatics Journal (2010) 11, 59](http://dx.doi.org/10.1016/j.eij.2010.10.002)–[65](http://dx.doi.org/10.1016/j.eij.2010.10.002)

Cairo University

Egyptian Informatics Journal

[www.elsevier.com/locate/eij](http://www.elsevier.com/locate/eij) [www.sciencedirect.com](http://www.sciencedirect.com/science/journal/11108665)

ORIGINAL ARTICLE

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

Waiel F. Abd El-Wahed [a](#_bookmark0), Elsayed M. Zaki [b](#_bookmark1), Adel M. El-Refaey [b](#_bookmark1),[\*](#_bookmark2)

a *Faculty of Computers & Information, Menoufia University, Egypt*

b *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

Abstract In this paper, a hybrid artificial intelligent approach based on the clonal selection prin- ciple 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

KEYWORDS

Artificial immune system; Neural networks; Nonlinear programming; Multi-objective program- ming;

Clonal selection

Elsevier B.V. All rights reserved.

1. Introduction

The field of immunological computation (IC) or artificial im-

\* Corresponding author.

E-mail address: [adel\_elrefaey@yahoo.com](mailto: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](http://dx.doi.org/10.1016/j.eij.2010.10.002)

Production and hosting by Elsevier

mune system (AIS) has been evolving steadily [[10]](#_bookmark15) since 1985. In recent years, several researchers have developed computa- tional models of the immune system that attempt to capture some of its most remarkable features such as its self-organizing capability [[4]](#_bookmark16).

The multi-objective immune system algorithm (MISA) can be considered as the first real proposal of MOAIS in the liter- ature [[2]](#_bookmark17). In the first proposal of the algorithm, authors at- tempted to follow the clonal selection principle very closely, then the algorithm performances have been improved in a suc- cessive version [[4]](#_bookmark16) 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

60 W.F. Abd El-Wahed et al.

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 indi- viduals, 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 dis- tribution of nondominated solutions [[4,10]](#_bookmark16).

The number of clones for the pool of best antibodies de- pends 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]](#_bookmark15).

Neural network (NN) is a well known as one of powerful computing tools to solve optimization problems. Due to mas- sive computing unit neurons and parallel mechanism of neural network approach it can solve the large-scale problem effi- ciently and optimal solution can be obtained [[13,16]](#_bookmark15).

The other hand neural network (NN) approach is attended as a new method for solving optimization problems, this meth- od 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 famil- iar 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]](#_bookmark15).

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.

1. 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]](#_bookmark16).

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 lym-

phocytes produced in the lymphoid organs. These lympho- cytes ‘‘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 net- work. 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]](#_bookmark18).

An artificial immune system (AIS) models some of the as- pects of a NIS, and is mainly applied to solve pattern recogni- tion 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 detec- tion [[4,10]](#_bookmark16).

* 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 mole- cules 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 prolif- erate (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](#_bookmark4) depicts the clonal selection principle.

The main features of the clonal selection theory [[2,10]](#_bookmark17) 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 accel- erated somatic mutation (a process called affinity maturation).
    3. Estimation of newly differentiated lymphocytes carrying low-affinity antigenic receptors.

1. Multi-objective programming (MOP) problem

This section provides the necessary mathematical background

problem with *k*\_objectives (*fi*(*x*)); (*j* = 1; 2; ... ; *k*) and *n* decision variables (*xi* = 1; 2; ... ; *n*): for MOP [[8,17]](#_bookmark15). Consider a multi-objective programming

*MOP:*

Min *F*(*x*)= (*f*1(*x*); ... ; *fk*(*x*))

Subject to

*S* = {*x* ∈ *Rn*|*g*(*x*) P 0; *h*(*x*)= 0} (1)

Artificial immune system based neural networks for solving multi-objective programming problems 61



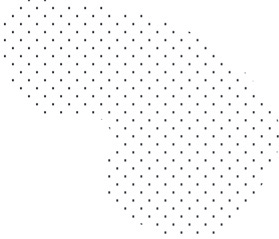


Figure 2 Geometrical representation of weighted method.

Figure 1 Clonal selection principle.

where *x* ∈ *Rn*, *f* : *Rn* → *Rk*, is *k*-dimensional vector valued con- tinuous functions of *n* variables, *g* = [*g* ; .. . ; *g* ]*T* : *Rn* → *Rm*, ables and *h* = [*h*1; .. . ; *hp*]*T* : *Rn* → *Rp*, is *p*-dimensional vector is *m*-dimensional vector valued continuous functions of *n* vari-

1 *m*

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]](#_bookmark18).

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

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

* 1. *Weighted method for MOP problem*

The weighted method [[8]](#_bookmark15) for the multi-objective optimization problem is formulated as:

*k*

*P*(*w*) : Min *wifi*(*x*)

X

Multi-objective optimization runs are conducted with dif- ferent 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 draw- backs. 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]](#_bookmark15).

The solution of the single objective *P(w)* Eq. [(2)](#_bookmark5) (in a min- imizing case) is shown in [Fig. 2](#_bookmark3), where the shaded area denotes the feasible region.

Theorem 1. *If x*\* ∈ *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].](#_bookmark15)

*convex. If x*\* ∈ *S is an efficient solution of the MOP, then x*\* Theorem 2. *Let the multi-objective optimization problem be W* = (*w*1; *w*2; ... ; *wk*) P 0 [[8]](#_bookmark15). *is an optimal solution of the weighting problem P(w) for some*

1. 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 func-

tion, we construct a gradient system of differential equations

which corresponds to a *neural network* [[9,18]](#_bookmark15).

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

*NLPP:*

X*k*

*i*=1

s.t. *x* ∈ *s*; *w* ∈ *W*;

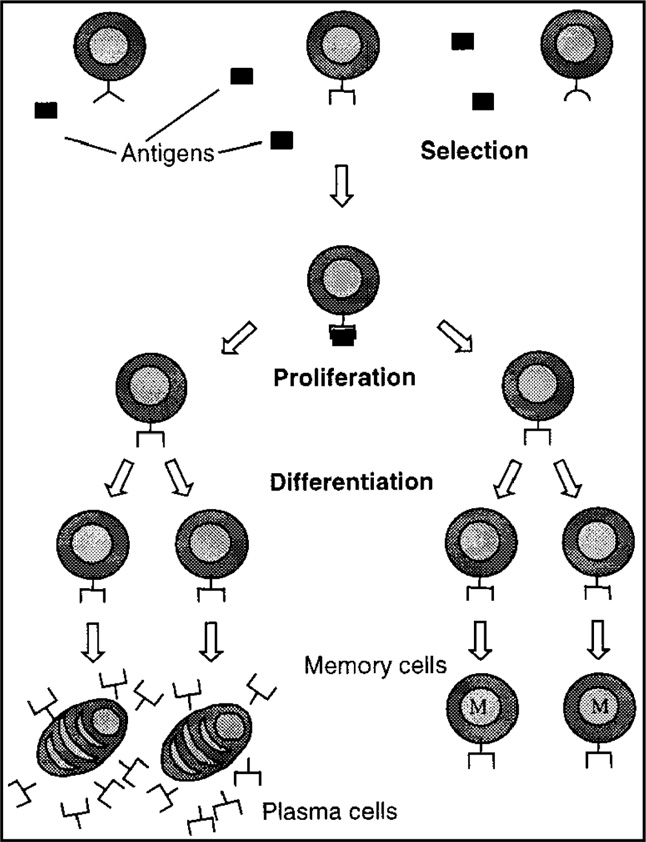
(2)

Min *F*(*x*)= *wjfj*(*x*)

*j*=1

(3)

*W* = (*w* ∈ *Rk*|*wj* P 0;



X

*k j*=1

*wj* = 1)

s.t. *gi* (*x*) P 0; *i* = 1; 2; .. . ; *m*

*hl* = *aTx* — *bj*; *l* = 1; 2; .. . ; *p*

*l*

where *W* = (*w* ∈ *Rk*|*wi* P 0;

*k*

X

*i*=1

*wi* = 1)

ear programming [[15]](#_bookmark19), Theorem 3 can be immediately obtained.

where *f* (*x*); *j* = 1; 2; .. . ; *k* and *g* (*x*)∈ *C*2 (the Set of all contin-

By Theorems 4 and 5, the energy function of Convex NLP

*j i* can be constructed as follows:

uous function which possess the second derivatives) (*i* = 1; 2; .. . ; *m*) are convex functions, and the vectors {*al*}*l*=1;2;...;*p* are linearly independent. We suppose throughout that NLPP

*E*(*z*)= *E*(*x*; k; l)= 1 k*Tg*(*x*) 2

1

+ 2 ∇*xL*(*x*;



k; l)2

has an optimal solution, and is strongly consistent, i.e., there 1 *T*

2

1 2 1 *T*

exists *x*0 ∈ *Rn*such that *gi*(*x*0) > 0, (*i*=1,2,.. .,m), *hj*(*x*0)=

0,(*j*=1,2,.. .,p) [[7]](#_bookmark15).

According to the result in Rao [[15]](#_bookmark19), the dual nonlinear pro-

+ 2 *g*(*x*) (*g*(*x*)— |*g*(*x*)|) + 2 *Ax* — *b *

+ 2 k (k — |k|)

(7)

gramming problem (DNPP) can formulated as follows:

*DNLPP:*

Max*x*;k;l *L*(*x*; k; l); *x* ∈ *Rn*

s.t. ∇*L*(*x*; k; l)= 0

k P 0; l unrestricted in sign;

where

k = (k1; k2; ... ; k*m*)*T*; l = (l ; l ; .. . ; l )*T*

*m p*

1 2 *p*

*L*(*x*; k; l)= *F*(*x*)— X k*igi* (*x*)— X l*jhj*(*x*)Ξ *L*(*z*)

(4)

where *z* = (*xT*; k*T*; l*T*)*T* ∈ *Rn*+*m*+*p*. Every term of the right-hand

inequality being satisfied in Eq. [(6)](#_bookmark9) and *E*(*z*) P 0. Thus *E*(*z*) side of Eq. [(7)](#_bookmark7) being zero corresponds to every equality or is differentiable function by Leung et al. [[9]](#_bookmark15). h

Theorem 6. *z*\* = (*x*\*; k\*; l\*)*T is zero point of E*(*z*) →⇒ *z*\* *is an optimal solution of NLPP and DNLPP (i.e., x*\* *and* (*x*\*; k\*; l\*)*T are optimal solutions of NLPP and DNLPP for a specific value of W, respectively).*

Proof. By Theorem 4, *z*\* = (*x*\*; k\*; l\*)*T* is zero point of

*E*(*z*) →⇒ *z*\* satisfies Eq. [(6)](#_bookmark9). By Theorem 5, *z*\* satisfies Eq.

*i*=1

*m*

*j*=1

*p*

[(6)](#_bookmark9) →⇒ *x*\* and (*x*\*; k\*; l\*)*T* are Pareto-optimal solutions of

∇*L*(*x*; k; l)= ∇*F*(*x*)— X k ∇*g* (*x*)— X l ∇*h* (*x*)

*i*=1

*i*

*i*

*j*=1

*j*

*j*

NLPP and DNLPP for a specific value of *W*, respectively.

Therefore, Theorem 6 is proved. h

Theorem 3. *Suppose that x*\* *is an optimal solution of convex* (k\*; l\*)*T* ∈ *Rm*+*p, such that* (*x*\*; k\*; l\*)*T, is an optimal solution nonlinear programming problem (CNLPP), then there exists of dual problem for a specific value of W* [[15]](#_bookmark19).

To entertain general inequality constraints, we can easily generalize Lemma 1 in Leung et al. [[9]](#_bookmark15) and obtain the following theorems:

Theorem 4. *Suppose g*(*x*)∈ *Rm*; *g*(*x*)∈ *C*1 *(the set of continu- ously differentiable function). Then*

Employing the unified idea in Leung et al. [[9]](#_bookmark15),we can use the

gradient system to construct the following multi-objective neu- ral network for solving a convex multi-objective optimization problems:

*dz*

*dt* = —∇*E*(*z*) (8)

Using the formula of computing gradients in Leung et al.

[[9]](#_bookmark15), the multi-objective neural network as in Eq. [(8)](#_bookmark8) can be written in details as follows:

*dx T T T*

*g*(*x*) P 0 →⇒

1 *g*(*x*)*T*(*g*(*x*)— |*g*(*x*)|) = 0 (5)

2

*dt* = —∇*xE*(*z*)= —k *g*(*x*).∇*g*(*x*) k — ∇*g*(*x*) (*g*(*x*)— |*g*(*x*)|)

— ∇2 *L*(*z*)∇*xL*(*z*)— *AT*(*Ax* — *b*)

*xx*

*and* 1 *g*(*x*)*T*(*g*(*x*)— |*g*(*x*)|) ∈ *C*1; *where* |*x*|= (|*x*1|; |*x*2|; ... ;

*T* 2

*d*

 = —∇k*E*(*z*)= —k*Tg*(*x*).*g*(*x*)+ ∇*g*(*x*)∇*xL*(*z*)— (k — |k|)

k

|*xn*|)

[[7]](#_bookmark15). *dt*

By Theorem 4, general inequality constraints can easily be

*d*l = —∇ *E*(*z*)= *A*∇ *L*(*z*)

transformed into equality constraints, and after the transfor- *dt* l *x*

mation, differentiability of the function is the same as that of

(8’)

*g*(*x*). It is easy to prove that 1 *g*(*x*)*T*(*g*(*x*)— |*g*(*x*)|) is a convex

2

function when *g*(*x*) is a concave function.

Theorem 5. *x*\**and* (*x*\*; k\*; l\*)*T are optimal solutions of NLPP and DNLPP, respectively, for a specific value of W, if and only*

*if*

*F*(*x*\*)= *L*(*x*\*; k\*; l\*); ∇*L*(*x*\*; k\*; l\*)= 0

*g*(*x*\*) P 0; *h*(*x*\*)= 0; k\* P 0; l\* unrestricted in sign

k\**Tg*(*x*\*)= 0 (6)

*and the set of vectors* ∇*hl*(*x*\*) *are linearly independent.*

Proof. By the duality theorem of the nonlinear programming

[[15]](#_bookmark19) and Kuhn Tucker optimality condition of convex nonlin-

where ∇2 *L*(*z*) is the Hessian matrix of the function *L*(*z*) with respect to *x*. Suppose ∇*E*(*z*) is Lipschitz continuous, then the initial value problem of differential equations in Eq. [(8)](#_bookmark8) has a

differential [Eq. (8) or](#_bookmark8) [(8)](#_bookmark8)' are continuous, these equations can unique solution because the function in the right hand side of easily be achieved by hardware implementation of the net-

*xx*

work. Therefore, it is a feasible neural network [[2,7]](#_bookmark17).

1. The proposed approach

The algorithm run in two stages, the first one run neural net- works 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 selec- tion principle*, [[1,4]](#_bookmark20) modeling the fact that only the highest affinity antibodies will proliferate. Antibodies, in our case, are repre- sented 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 iden- tify solutions that deserve to be cloned. Additionally, our ap- proach uses mutation [[4]](#_bookmark16) (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 second- ary (or external) population that stores the nondominated solu- tions 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]](#_bookmark21).

* 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]](#_bookmark15)

[*Step 1*] InitializationLet *t* = 0. Randomly choose initial vec- tor *x*(*t*)∈ *Rn*, k(*t*)∈ *Rm*, l(*t*)∈ *Rp*, D*t* > 0 (for exam- ple, D*t* = 0.0001) and error e = 10—9.

[*Step 2*] Transform the of MOP into NLPP Eq. [(3)](#_bookmark6)

[*Step 3*] Computation of gradient:

*u*(*t*)= ∇*xE*(*z*)= k*Tg*(*x*) · *g*(*x*)*T*k + ∇*g*(*x*)*T*[*g*(*x*)— |*g*(*x*)|]

+ ∇2 *L*(*z*)∇*xL*(*z*)+ *AT*(*Ax* — *b*)

*xx*

*v*(*t*)= ∇k*E*(*z*)= k*Tg*(*x*) · *g*(*x*)— ∇*g*(*x*)∇*xL*(*z*)+ [k — |k|]

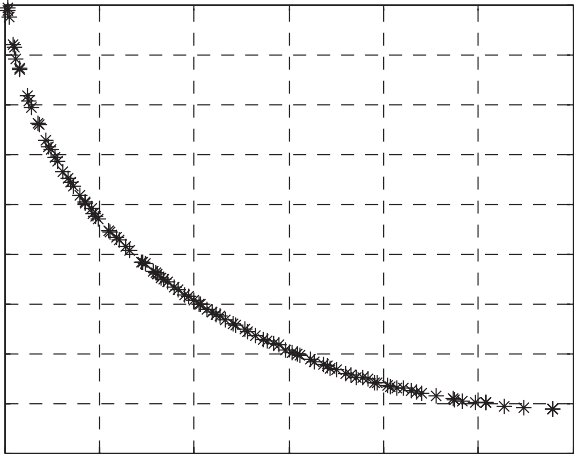
*w*(*t*)= ∇l*E*(*z*)= —*A*∇*xL*(*z*)

[*Step 4*] States updating:

*x*(*t* + D*t*)= *x*(*t*)— D*t*.*u*(*t*) k(*t* + D*t*)

= k(*t*)— D*t* · *v*(*t*)l(*t* + D*t*)= l(*t*)— D*t* · *w*(*t*)

BNH problem

45

40

35

30

25

f2

20

15

10

5

0

0 20 40 60 80 100 120

f1

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

1. Experiments

In order to validate our approach, five benchmark functions which reported in the standard evolutionary multi-objective optimization literature [[5,14]](#_bookmark15).

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]](#_bookmark22), M\_OU by Cvetkovic, M\_LOC by Kita, M\_3OU by Viennet [[5]](#_bookmark15).

Problem 1 (*BNH*).

Min *f*1(*x*)= 4*x*2 + 4*x*2

1 2

*f* (*x*)= (*x* — 5)2 + (*x* — 5)2

2 1 2

s.t. *g* (*x*)= (*x*1 — 5)2 + *x*2 6 25

1

2

[*Step 5*] Calculate:

*n*

*i*

X

*j*

*p*

*j*

*s* =

X

*i*=1

*u*2(*t*), *r* =

*m j*=1

*v*2(*t*), *q* =

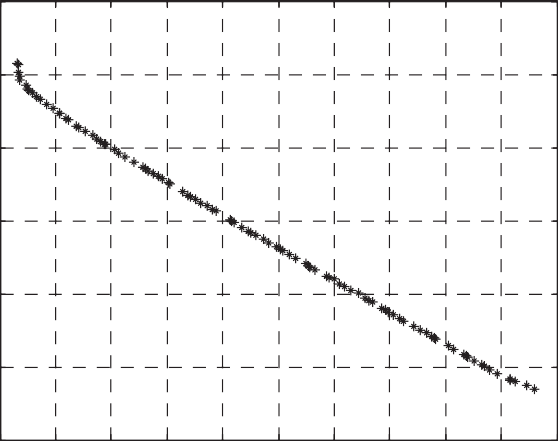
X*j*=1

*w*2(*t*)

*g* (*x*)= (*x*1 — 8)2 + (*x*2 + 3)2 P 7.7

SRN problem

2

50

[*Step 6*] Stopping rule:

if *s* < e, *r* < e and *q* < e, then output *x*(*t* + D*t*), k(*t* + D*t*), l(*t* + D*t*) into the input file of MISA; otherwise let *t* = *t* + D*t* and go to Step 3.

0

-50

*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*] Appling a uniform mutation to the clones.

[*Step 6*] Appling 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-

-100

-150

f2

-200

-250

0 20 40 60 80 100 120 140 160 180 200

f1

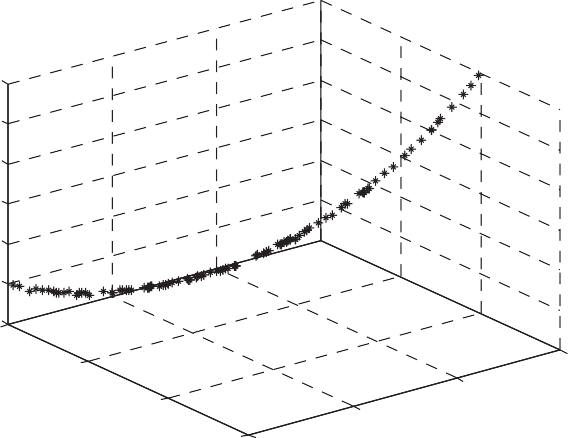
determined) number of times.

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

1

0.5

M UC problem

4

3.8

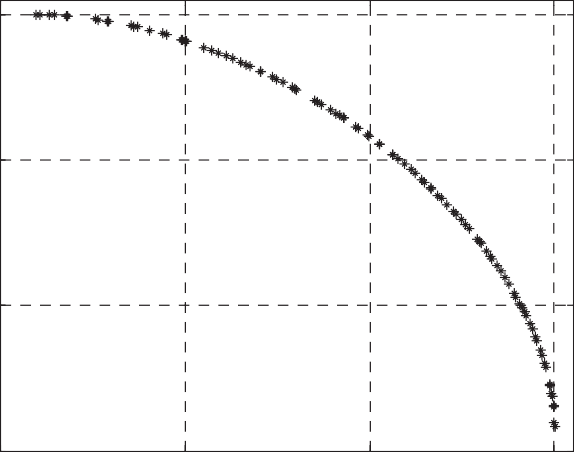
3.6

3.4

f3

3.2

M 3OU problem

0

-0.5

-0.5 0 0.5 1

3

f2

2.8

6

5.5

f2

8

7

5 6

4.5 5 f1

f1

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

M LOC problem

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

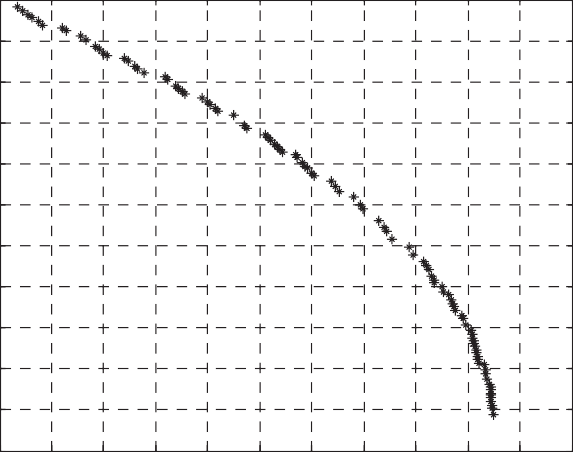
Problem 3 (*M\_UC*).

Max *f*1(*x*)= sin ÿ*x*2 + *x*2 — 1

1

2

*f*2 (*x*)= sin ÿ*x*2 + *x*2 + 1

8.4

8.3

8.2

8.1

8

f2

7.9

7.8

7.7

7.6

7.5

7.4

-3 -2 -1 0 1 2 3 4 5 6 7 8

f1

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

1 2

s.t. 0 6 *x*1 6 3p/4

0 6 *x*2 6 3p/4

The output of NN indicates that the range of variables will 0 6 *x*1 6 2 and 0 6 *x*2 6 2. Pareto front using MISA based NN shown in [Fig. 5](#_bookmark12).

Problem 4 (*M\_LOC*).

Max *f*1(*x*)= —*x*2 + *x*2

1

*f*2(*x*)= *x*1/2 + *x*2 + 1

s.t. *g*1(*x*)= *x*1/6 + *x*2 6 13/2 *g*2(*x*)= *x*1/2 + *x*2 6 15/2 *g*3(*x*)= 5*x*1 + *x*2 6 30

The output of NN indicates that the range of variables will 0 6 *x*1 6 5 and 3 6 *x*2 6 7. Pareto front using MISA based NN shown in [Fig. 6](#_bookmark14).

The output of NN indicates that the range of variables will 0 6 *x*1 6 5 and 0 6 *x*2 6 3. Pareto front using MISA based NN shown in [Fig. 3](#_bookmark10).

Problem 5 (*M\_3OU*).

Min *f*1(*x*)= *x*2 + (*x*2

1

*f*2(*x*)= *x*2 + (*x*2

1

— 1)2

+ 1)2 + 1

*f* (*x*)= (*x* — 1)2 + *x*2 + 2

3 1 2

Problem 2 (*SRN*).

s.t — 2 6 *x*1 6 2 — 2 6 *x*2 6 2

Min *f* (*x*)= (*x* — 2)2 + (*x* — 1)2 + 2

1 1 2

The output of NN indicates that the range of variables will

*f*2(*x*)= 9*x*1 — (*x*2

— 1)2

0 6 *x*1 6 2 and —1 6 *x*2 6 2. Pareto front using MISA based

NN shown in [Fig. 7](#_bookmark13).

s.t. *g*1(*x*)= *x*2 + *x*2 6 225

1 2

*g*2(*x*)= —*x*1 + 3*x*2 P 10

The output of NN indicates that the range of variables will

—12 6 *x*1 6 12 and —10 6 *x*2 6 10. Pareto front using MISA based NN shown in [Fig. 4](#_bookmark11).

1. Conclusion

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

works. The approach is able to produce results similar or bet- ter than those generated by other evolutionary algorithms after determining the max and min values with NN and use it to ini- tialize 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. Affin- ity in this case, is defined in terms of nondominance and feasi- bility. 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 interna- tional 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. Evolution- ary 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. Klu- wer 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, Universite´Libre De Bruxelles Faculte´Des Sciences Applique´es; 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 mak- ing: 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.

Further Reading

1. Nunes de Castro L, Von Zuben FJ. Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 2002;6(3):239–51.