Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)

Artificial Immune System for Solving Constrained Global Optimization Problems

J. Y. Wu
Department of Industrial Engineering and Management, Diwan College of Management,
87-1, Nanshi Li, Madou Town, Tainan County 72153, Taiwan
juiyu.wu@msa.hinet.net

***Abstract*- The artificial immune system (AIS) is a computational intelligence approach based on information regarding a biological immune system. This study combines the metaphor of clonal selection and idiotypic network theories to design an AIS method. Although contradicting each other, these two theories are useful in developing a function optimization tool. The AIS approach comprises idiotypic network selection, somatic hypermuation, receptor editing and bone marrow operators. The idiotypic network selection operator controls the number of good solutions. The somatic hypermuation and receptor editing operators explore a search space of solutions to an optimization problem. The bone marrow operator generates diverse solutions to maintain the population of solutions. The performance of the proposed AIS method is measured by using it to solve a set of constrained global optimization (CGO) problems. The best AIS solution is compared with the known global optimum. Numerical results show that the proposed method converged to the global optimal solution to each tested CGO problem.**

I. INTRODUCTION

Many scientific and engineering problems can be formulated as constrained global optimization (CGO) problems, as follows:

$$\text{Minimize} \quad f(\mathbf{x}) \tag{1}$$

$$\text{subject to} \quad g_m(\mathbf{x}) \le 0, \quad m = 1,2,\ldots,M \tag{2}$$

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

$$x_n^l \le x_n \le x_n^u, \quad n = 1,2,\ldots,N \tag{4}$$

where $g_0(\mathbf{x})$ represents an objective function; $g_m(\mathbf{x})$ is a set of *m* nonlinear inequality constraints; $h_k(\mathbf{x})$ is a set of *k* nonlinear equality constraints; $\mathbf{x}$ represents a vector of decision variables with real values, and each decision variable x_n is constrained by its lower and upper boundaries $[x_n^l, x_n^u]$. All methods that solve the CGO problems can be classified as local or global optimizations. Local optimization techniques have many limitations. For example, these techniques guarantee to locate a global minimum in the neighborhood of a staring point, but strongly rely upon choosing a starting point. These techniques fail to find a global minimum solution to a CGO problem that has multiple local optima. Gradient-based optimization methods that need information on the gradient of an objective function, are inefficient when the objective function is non-differentiable. These methods guarantee to produce local optimal solutions to a CGO problem. Many global optimization methods have been developed to overcome these disadvantages. These methods can be divided into two categories–deterministic and stochastic optimizations [1]. Deterministic global optimization methods often involve a sophisticated optimization process and usually make some assumptions regarding the problem to be solved [2]. Although these approaches produce a global solution to a CGO problem, they can be computationally tedious and difficult for general practitioners to use.

The simulated annealing (SA) algorithm and genetic algorithms (GAs) are popular stochastic global optimization approaches. In 1953, Metropolis *et al.* [3] first simulated the physical annealing process using a Monte Carlo method. Their approach was preferred until Kirkpatrick *et al.* [4] used an SA algorithm to solve combinational optimization problems, such as traveling salesman problems. Various SA algorithms have been developed for application to CGO problems [5, 6]. Although SA algorithms have been proven statistically to converge to a global optimum, the randomness of Monte Carlo causes that SA algorithms cannot guarantee to reach a global optimum without unlimited resource [7]. Holland first introduced GAs in 1970s. GAs have been widely used to solve optimization problems. GAs are described in detail in the literature [8]. Unfortunately, GAs have two disadvantages–the lack of a local search ability and premature convergence [9].

In the past decade, a field of computational intelligence called artificial immune systems (AIS) has emerged, and is now of interest to many researchers. AIS approaches mimic the process by which the immune system (IS) learns, memorizes, identifies and destroys foreign materials such as viruses, pathogens and bacteria (called antigens, **Ags**). AIS methods are commonly based on population-based or network-based immune algorithms [10]. They have been successfully used in various function optimization problems. For instance, de Castro and Von Zuben [11] presented an AIS approach called CLONALG to solve mutimodal optimization problems; de Castro and Timmis [12] developed an AIS method called opt-aiNET to be applied to mutimodal optimization problems, and Coello Coello and Cruz Cortés [13] designed a constraint-handling technique based on AIS to solve CGO problems.

This study presents an AIS method to overcome the difficulties of local optimization techniques, global optimization methods, SA algorithms and GAs. The proposed AIS approach is naturally an unconstrained optimization technique. Therefore, a penalty function method that can transform a CGO problem into an unconstrained optimization problem is employed. The performance, in terms of effectiveness, efficiency and ease of use, of the proposed AIS approach is evaluated by solving a set of CGO problems.

II. IMMUNE SYSTEM

Human immunity comprises innate and adaptive immunities. Innate immunity provides immediate defense of the host, destroying foreign **Ags** using the macrophages and natural killer cells. The immune response has no immunological memory, since the response can not be altered by repeated exposure to specific **Ags**. After innate immunity is achieved, adaptive immunity, which has an immunological memory for specific **Ags**, is activated. The immune response consists of antigen-specific reactions of T and B cells (lymphocytes), which operate as in cell-mediate and homoral immunities, respectively [14]. This study focuses on B cell response. B cells that develop in bone marrow produce antigen-specific antibodies (**Abs**) to fight with **Ags**. The receptors located on a B cell surface are called **Abs**. Each B cell can only manufacture one form of **Abs**. In this study, an **Ab** is considered to be a B cell, although **Abs** are only receptors of a B cell.

A. *Ag and Ab*

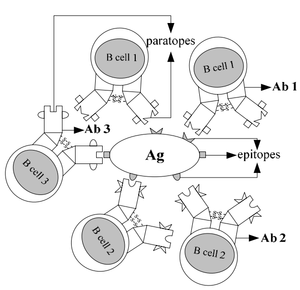

Fig. 1. **Ag** and **Abs**

Figure 1 shows an **Ag** and **Abs**. The **Ag** has multiple epitopes (antigenic determinants), which can be recognized by various **Abs** with paratopes (recognizers), on its surface. An **Ab** and an **Ag** have high **Ab-Ag** affinity when the paratope of **Ab** and the epitope of **Ag** have complementary shapes. Figure 2 presents an **Ab** structure.

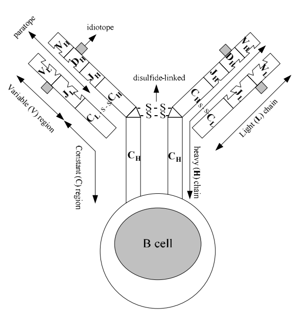

Fig. 2. **Ab** structure

The **Ab** is composed of two identical heavy (**H**) and two identical light (**L**) polypeptide chains. These chains consist of variable (V) and constant (C) regions. The V-region can recognize foreign **Ags**, while the C-region is responsible for a variety of effector functions, including complement fixation. Furthermore, **H** chain comprises gene segments variable $\mathbf{V_H}$, diversity $\mathbf{D_H}$, joining $\mathbf{J_H}$ and C-region gene $\mathbf{C_H}$; **L** chain consists of gene segments variable $\mathbf{V_L}$, joining $\mathbf{J_L}$ and C-region gene $\mathbf{C_L}$ [15]. In humans, bone marrow can synthesize diverse **Abs** by recombining gene segments ($\mathbf{V_H}$ $\mathbf{D_H}$ $\mathbf{J_H}$ and $\mathbf{V_L}$ $\mathbf{J_L}$). Moreover, the **Ab** also has two immunogenic idiotopes (antigenic determinates) that can be identified by other **Abs** in IS [16].

B. *Immune Theories*

Jenre [16] presented an idiotypic network theory based on **Ab-Ab** recognition to represent the dynamics of a set of identical B cells. Perelson [17] subsequently reviewed Jenre's theory and developed a general network model, as follows:

$$RIP = B - D + R \tag{5}$$

where
RIP = rate of increase of population of B cells
B = influx from bone marrow
D = death of un-stimulated B cells
R = reproduction of stimulated B cells

The first two terms in Eq. (5) represent immune network metadynamics, which are the continuous manufacture and recruitment of diverse B cells. The last term includes **Ab-Ag** and **Ab-Ab** recognition information. Although Langman and Cohn [18] asserted that the conceptual foundations of the idiotypic network are formal absurdities, the general model defined in Eq. (5) is useful for developing a computer tool.

Clonal selection theory is used to capture the basic concepts that are involved in an adaptive immune response to an antigenic stimulus. The clonal selection algorithm based on clonal selection theory has two features–somatic hypermutation and receptor editing [11]. Hypermutation is the local exploration of the **Ab-Ag** affinity landscape. Receptor editing provides the ability to escape from the local **Ab-Ag** affinity landscape. Therefore, hypermuation and receptor editing have complementary roles in **Ab-Ag** affinity maturation process [19].

Exactly how these two immune theories differ is discussed. Idiotypic network theory emphasizes the **Abs** of the IS are interconnected. Therefore, two levels of interaction exist − **Ag-Ab** and **Ab-Ab** recognitions. Clonal selection theory considers that the **Abs** of the IS are not connected to each other. Thus, **Abs** interact directly with **Ags**.

III. PENALTY FUNCTION METHODS

Penalty function methods, which are constraint handling techniques, are usually used in evolutionary algorithms, such as GAs, to solve CGO problems. Exterior and interior penalty functions are popular. Exterior penalty functions use an infeasible solution as a starting point, and the direction of convergence is from the infeasible region to the feasible region. Interior penalty functions start from a feasible solution, and move from the feasible region to the constrained boundaries. Exterior penalty functions are favored over interior penalty functions, because they do not require a feasible starting point and are easily implemented. Many exterior penalty functions have been performed, such as static, dynamic, adaptive and death penalty functions [20]. This study used an adaptive penalty function, as follows:

$$\text{Minimize} \quad f_{\text{pesudo}} = f(\mathbf{x}) + \rho_g \left\{ \sum_{m=1}^{M} \{\max[0, g_m(\mathbf{x})]\}^2 \right\} \qquad (6)$$

where

f_{pesudo} = pseudo-objective function obtained using an original objective function plus a penalty term

ρ_g = penalty parameter in current generation g

The parameter ρ_g can be adaptively modified by:

$$\rho_{g+1} = \begin{cases} \rho_g / c_1 & \text{if } g \ge \kappa \text{ and } \mathbf{x}^l \in S \text{ for all } g - \kappa + 1 \le l \le g \\ c_2 \rho_g & \text{if } g \ge \kappa \text{ and } \mathbf{x}^l \notin S \text{ for all } g - \kappa + 1 \le l \le g \\ \rho_g & \text{otherwise} \end{cases} \qquad (7)$$

where

ρ_{g+1} = penalty parameter in generation $g+1$

c_1, $c_2 > 1$ and $c_1 \ne c_2$ (to avoid cycle)

κ = number of past generations

$\mathbf{x}^l$ = best solution in generation l

S = feasible solution

This study defined a solution with a constraint violation of $vio = \sum_{m=1}^{M} \max[0, g_m(\mathbf{x})] \le 1\times10^{-6}$ as feasible. These settings can yield an accuracy of at least four decimal places for the violation of each constraint of the AIS solution to a testing CGO problem, as follows:

$\rho_0 = 1\times10^{9}$ (initial penalty parameter), $\kappa = 50$, $c_1 = 1.5$ and $c_2 = 10$.

Moreover, the parameter ρ_{g+1} was limited within the interval $[1\times10^{7}, 1\times10^{11}]$.

IV. METHOD

An AIS approach based on the metaphor of the clonal selection and idiotypic network theories was developed and applied to solve CGO problems. Although these two theories conflict with each other (as described in Section II), they are useful in designing a function optimization tool. Figure 3 shows the pseudo-code of the proposed AIS method, which is described below.

Step 1: Initialization

Many parameters must be predetermined, such as repertoire (population) size rs and the threshold degree of **Ab-Ab** recognition p_{rt}. Section V considers the manipulation of these parameters. Real numbers are used to represent **Ab** and **Ag**, since real-coded method yields a more precise solution than that obtained using the binary-coded method to solve optimization problems with continuous decision variables x_n. Figure 4 shows the **Ag** and **Ab** the representation. The epitope of **Ag** in Fig. 4 describes the known parameters in a CGO problem; the paratope of **Ab** represents the decision variables x_n of the CGO problem, and the idiotope of **Ab** is responsible for **Ab-Ab** recognition. An available **Ab** repertoire (population) is randomly created based on rs from [x_n^l, x_n^u].

```
begin
  g ← 0
  Step 1: initialization
      (a) creation of rs and p_rt
      (b) generation of available Ab repertoire
      while g < g_max do
          Step 2: evaluation of Ab-Ag affinity
              Ab* ← max( affinity_j ), j = 1,2,…,rs
          Step 3: clonal selection
              for each Ab_j, j = 1,2,…,rs do
                  if p_rj ≥ p_rt then
                      promotion (clone)
                  else
                      suppression
                  endIf
              endFor
          Step 4: affinity maturation
              for each promoted Ab_j do
                  if rand() ≤ 0.5 do
                      somatic hypermutation
                  else
                      receptor editing
                  endIf
              endFor
          Step 5: introduction of diverse Abs
          Step 6: update of Ab repertoire
          g ← g + 1
      endWhile
end
```

Fig. 3. The pseudo-code of the proposed AIS method

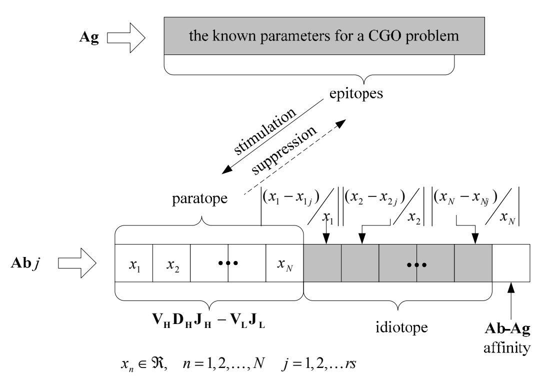

Fig. 4. **Ag** and **Ab** representation

Step 2: Evaluation of Ab-Ag affinity

A pseudo-objective function is obtained using the adaptive penalty function, as defined in Eq. (6). Equation (6) is transformed to continue the **Ab-Ag** affinity metaphor, as follows:

$$\text{Maximize } affinity_j = -1 \times \left\{ f(\mathbf{x}) + \rho_g \sum_{m=1}^{M} \{\max[0, g_m(\mathbf{x})]\}^2 \right\}, \quad (8)$$

$$j = 1,2,\ldots,rs$$

Equation (8) is then used to measure **Ab-Ag** affinity. After the **Ab-Ag** affinities of **Abs** in the current **Ab** repertoire are evaluated, the **Ab** with the highest **Ab-Ag** affinity is selected to undergo clonal selection in Step 3. The best **Ab** ($\mathbf{Ab}^*$) is defined by x_n^* ($n = 1,2,\ldots,N$).

Step 3: Clonal selection

Various selection operators can be applied to create an intermediate **Ab** repertoire, such as roulette wheel selection, elitist selection, rank-based selection and tournament selection [21]. However, this study presents an operator called idiotypic network selection, based on the idiotypic network theory [16], to control the number of antigen-specific **Abs** and reproduce them. The operator is defined by:

$$p_{r_j} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{e^{d_{nj}}} \quad (9)$$

$$d_{nj} = \left| \frac{x_n^* - x_{nj}}{x_n^*} \right|, \; j = 1,2,\ldots,rs, \;\; n = 1,2,\ldots,N \quad (10)$$

where

p_{r_j} = probability that **Ab** *j* recognizes $\mathbf{Ab}^*$

x_{nj} = decision variables x_n of **Ab** *j*

The $\mathbf{Ab}^*$ selected in Step 2 is the foreign **Ag**. The $\mathbf{Ab}^*$ is recognized by other **Ab** *j* in the current **Ab** repertoire. A large value of p_{r_j} means that **Ab** *j* can effectively recognize $\mathbf{Ab}^*$. The **Ab** *j* with p_{r_j} that is equal to or larger than the threshold degree p_{rt} is reproduced to create an intermediate **Ab** repertoire. The **Ab** *j* with p_{r_j} that is smaller than the p_{rt} is suppressed.

Step 4: Affinity maturation

The intermediate **Ab** repertoire is divided into two subsets. A uniform random number is generated for each **Ab** in the intermediate **Ab** repertoire. These **Abs** undergo somatic hypermutation when their random numbers equal or are smaller than 0.5. They undergo receptor editing when their random numbers exceed 0.5. Somatic hypermutation and receptor editing are described below.

1) Somatic hypermutation

This study employs multi-non-uniform mutation [22] as the somatic hypermutation operator, which can be stated as follows:

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x_n^u - x_{\text{current},n})A(g), & if\ U(0,1) < 0.5 \\ x_{\text{current},n} - (x_{\text{current},n} - x_n^l)A(g), & if\ U(0,1) \ge 0.5 \end{cases} \quad (11)$$

where

$$A(g) = \left[U_1(0,1)\left(1 - \frac{g}{g_{\max}}\right) \right]^2 = \text{perturbation factor}$$

$x_{\text{current},n}$ = current value of decision variable x_n

$x_{\text{trial},n}$ = trial value of decision variable x_n

g = current generation

$g_{\max}$ = maximum generation number

$U(0,1)$ and $U_1(0,1)$ = uniform random number

This operator has two tasks, uniform search and local fine-tuning.

2) Receptor Editing

Standard Cauchy distribution $C(0,1)$, in which the local parameter is zero and scale parameter is one, is used to develop a receptor editing operator. The probability density function of $C(0,1)$ is as follows:

$$f_{C(0,1)}(s) = \frac{1}{\pi}\frac{1}{1+s^2}, \quad -\infty \le s \le \infty \quad (12)$$

Cauchy random variables generated from $C(0,1)$ are employed to perform receptor editing, since they can offer a large jump in **Ab-Ag** affinity landscape to increase the probability of escape from the local **Ab-Ag** affinity landscape. The proposed Cauchy receptor editing can be expressed as follows:

$$\mathbf{x}_{\text{trial}} = \mathbf{x}_{\text{current}} + [U_2(0,1)]^2 \times \boldsymbol{\sigma} \quad (13)$$

where

$\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_N]^T$, vector of Cauchy random variables

$U_2(0,1)$ = uniform random number in the interval [0, 1]

This operator functions in local fine-tuning and large perturbation.

Step 5: Introduction of diverse Abs

The paratope of an **Ab** can be created by recombining gene segments $\mathbf{V_H}$ $\mathbf{D_H}$ $\mathbf{J_H}$ and $\mathbf{V_L}$ $\mathbf{J_L}$. Based on this metaphor, this study presents a bone marrow operator to synthesize diverse **Abs** to recruit the **Abs** that were suppressed in Step 3. Figure 5 shows the bone marrow operator. This operator randomly chooses two **Abs** from the intermediate **Ab** repertoire and a recombination point from the gene segments of the paratope of the selected **Abs**. As shown in Fig. 5, the selected gene segments (gene x_2 of **Ab** 1 and gene x_2 of the **Ab** 2) are reproduced to create a library of gene segments, and the selected gene segments in the paratope are then deleted. The new **Ab 1** is formed by inserting into the gene segment, which is gene x_2 of the **Ab** 2 in the library plus a random variable generated from standard normal distribution $N(0,1)$, at the recombination point. Accordingly, the new **Ab 2** is created. Receptor editing, as described in Step 4, can be applied to these **Abs** to increase the diversity of the created **Abs**. Finally, the gene segments of the idiotope, corresponding to paratope at the recombination point, are also altered.

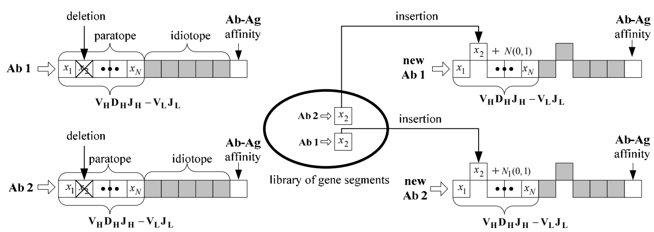

Fig. 5. Illustration of bone marrow operator

Step 6: Update of Ab repertoire

A new **Ab** repertoire is generated based on Eq. (5), consisting of the **Abs** that had been created from Steps 4 and 5. The **Ab-Ag** affinities of the **Abs** in the generated **Ab** repertoire are measured. This study presents a strategy for updating the **Ab** repertoire. If the **Ab-Ag** affinity of **Ab** *j* in the new **Ab** repertoire exceeds that in the current **Ab** repertoire, then the strong **Ab** in the new **Ab** repertoire replaces the weak **Ab** in the current **Ab** repertoire. If the **Ab-Ag** affinity of **Ab** *j* in the new **Ab** repertoire is equal to or worse than that in the current **Ab** repertoire, then the **Ab** *j* in the current **Ab** repertoire survives. This strategy not only maintains the strong **Abs**, but also effectively eliminates non-functional **Abs**.

Steps 2-6 are repeated until the termination criterion $g_{\max}$ is met.

V. RESULTS

The proposed AIS method, as described in Section IV, was applied to a set of CGO problems taken from other studies [13, 23, 24]. To measure the AIS sensitivity relative to rs, the parameter p_{rt} was fixed, and the values $rs=\{10,50,100,150\}$ were set. To evaluate the AIS sensitivity relative to p_{rt}, the parameter rs was fixed, and the values $p_{rt}=\{0.9,0.95\}$ were used. The proposed AIS approach was coded in MATLAB and was run on a Pentium 4 2.4 (GHz) PC. For each test problem, the proposed method was run 50 times independently, under a set of parameter settings (rs and p_{rt}). The termination conditions were for test problem 1 $g_{\max}$ = 3500, for test problem 3 $g_{\max}$ = 5000 and for test problems 2 and 4 $g_{\max}$ = 3000. Numerical results were summarized, including the best, median, worst and mean CPU times (MCT).

A. *Test problem 1 (TP1)*

TP1 has ten decision variables, eight inequality constraints and 20 boundary conditions, as follows:

$$\text{Minimize}\quad f(\mathbf{x})=x_1^2+x_2^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+4(x_4-5)^2+(x_5-3)^2+2(x_6-1)^2+5x_7^2+7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45$$

$$\text{Subject to}\quad g_1(\mathbf{x})\equiv -105+4x_1+5x_2-3x_7+9x_8\le 0,$$
$$g_2(\mathbf{x})\equiv 10x_1-8x_2-17x_7+2x_8\le 0,$$
$$g_3(\mathbf{x})\equiv -8x_1+2x_2+5x_9-2x_{10}-12\le 0,$$
$$g_4(\mathbf{x})\equiv 3(x_1-2)^2+4(x_2-3)^2+2x_3^2-7x_4-120\le 0,$$
$$g_5(\mathbf{x})\equiv 5x_1^2+8x_2+(x_3-6)^2-2x_4-40\le 0,$$
$$g_6(\mathbf{x})\equiv x_1^2+2(x_2-2)^2-2x_1x_2+14x_5-6x_6\le 0,$$
$$g_7(\mathbf{x})\equiv 0.5(x_1-8)^2+2(x_2-4)^2+3x_5^2-x_6-30\le 0,$$
$$g_8(\mathbf{x})\equiv -3x_1+6x_2+12(x_9-8)^2-7x_{10}\le 0,$$
$$-10\le x_n\le 10,\quad n=1,2,\ldots,10.$$

The global solution to TP1 is as follows.

$\mathbf{x}^*$ = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927),

$f(\mathbf{x}^*)$ = 24.306.

Table I shows the numerical results obtained when the AIS method has fixed p_{rt} = 0.9, and $rs=\{10,50,100,150\}$ are used for TP1. This table indicates that the best and worst results obtained with rs = 100 were better than those obtained with $rs=\{10,50,150\}$, and increasing rs increased the CPU computational time.

TABLE I

Numerical results obtained by fixing p_{rt} = 0.9 and using various rs s for TP1

	p_{rt} = 0.9			
	rs = 10	rs = 50	rs = 100	rs = 150
Best	24.934	24.446	24.377	24.382
Median	26.381	24.877	24.663	24.627
Worst	29.233	27.268	24.988	25.496
MCT	8.47 sec.	16.80 sec.	26.70 sec.	36.26 sec.

Table II lists the numerical results obtained using the AIS method with fixed rs = 100 and $p_{rt}=\{0.9,0.95\}$. This table indicates that the best, median and worst results obtained using p_{rt} = 0.9 were superior to those obtained using p_{rt} = 0.95.

TABLE II

Numerical results obtained by fixing rs = 100 and using various p_{rt} s for TP1

	rs = 100	
	p_{rt} = 0.9	p_{rt} = 0.95
Best	24.377	24.487
Median	24.663	24.914
Worst	24.988	26.418
MCT	26.70 sec.	27.99 sec.

The best solution obtained using the AIS approach was

$\mathbf{x}^*_{\text{AIS}}$ = (2.18791147, 2.33643103, 8.75582941, 5.11528280, 0.96298318, 1.38959267, 1.34627148, 9.84486741, 8.30123925, 8.34348600), $f(\mathbf{x}^*_{\text{AIS}})$ = 24.377.

Each constraint was met as follows.

$g_m(\mathbf{x}^*_{\mathrm{AIS}}) =$ (−0.001207, −0.009214, −0.011205, −0.610655, −0.009739, −0.066272, −6.182473, −47.334679).

B. Test problem 2 (TP2)

TP2 involves five decision variables, six inequality constraints and 10 boundary conditions, as follows:

$$\begin{aligned}
\text{Minimize}\quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141\\
\text{Subject to}\quad & g_1(\mathbf{x}) \equiv -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0,\\
& g_2(\mathbf{x}) \equiv -6.665593 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \le 0,\\
& g_3(\mathbf{x}) \equiv 9.48751 - 0.0071371x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 \le 0,\\
& g_4(\mathbf{x}) \equiv -29.48751 + 0.0071371x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 0,\\
& g_5(\mathbf{x}) \equiv 10.669039 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 \le 0,\\
& g_6(\mathbf{x}) \equiv -15.699039 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 0,\\
& 78 \le x_1 \le 102,\\
& 33 \le x_2 \le 45,\\
& 27 \le x_n \le 45, \quad n = 3,4,5.
\end{aligned}$$

The global solution to TP2 is

$\mathbf{x}^* = (78.0, 33.0, 29.995256, 45.0, 36.775812)$,

$f(\mathbf{x}^*) = -30665.539$.

The optimal settings of the parameters in the AIS method were rs = 100 and p_{rt} = 0.9, and Table III summarizes the best numerical results for TP2.

TABLE III
The best numerical results for TP 2

Parameter settings	Best	Median	Worst	MCT
rs = 100, p_{rt} = 0.9	−30665.543	−30665.524	−30665.500	16.93 sec.

The best solution obtained by the proposed AIS method was

$\mathbf{x}^*_{\mathrm{AIS}} =$ (78.00000054, 33.00001145, 29.99523373, 44.99989602, 36.77585061), $f(\mathbf{x}^*_{\mathrm{AIS}}) = -30665.543$.

Each constraint had an accuracy of five decimal places, as follows.

$g_m(\mathbf{x}^*_{\mathrm{AIS}}) =$ (−92.000004, 0.000004, −8.840512, −11.159488, 0.000009, −5.000009).

C. Test problem 3 (TP3)

TP3 has seven decision variables, four inequality constraints and 14 boundary conditions, as follows:

$$\begin{aligned}
\text{Minimize}\quad & f(\mathbf{x}) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6\\
& \qquad + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7\\
\text{Subject to}\quad & g_1(\mathbf{x}) \equiv -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0,\\
& g_2(\mathbf{x}) \equiv -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0,\\
& g_3(\mathbf{x}) \equiv -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0,\\
& g_4(\mathbf{x}) \equiv 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0,\\
& -10 \le x_n \le 10, \quad n = 1,2,\ldots,7.
\end{aligned}$$

The global solution to TP3 is

$\mathbf{x}^* =$ (2.330499, 1.951372, −0.4775414, 4.365726, −0.6244870, 1.038131, 1.594227), $f(\mathbf{x}^*) = 680.630$.

The optimal settings of the parameters of the AIS method were rs = 100 and p_{rt} = 0.9, and Table IV lists the best numerical results for TP3. The best solution obtained using the proposed AIS approach was

$\mathbf{x}^*_{\mathrm{AIS}} =$ (2.32369042, 1.95040940, −0.48948969, 4.37120767, −0.63059961, 1.03715874, 1.58827062), $f(\mathbf{x}^*_{\mathrm{AIS}}) = 680.632$.

Every constraint was satisfied as follows.

$g_m(\mathbf{x}^*_{\mathrm{AIS}}) =$ (−0.000129, −252.485130, −145.002999, −0.000180).

TABLE IV
The best numerical results for TP 3

Parameter settings	Best	Median	Worst	MCT
rs = 100, p_{rt} = 0.9	680.632	680.640	680.651	31.58 sec.

D. Test problem 4 (TP4)

TP4 involves 13 decision variables, nine inequality constraints and 26 boundary conditions, as follows:

$$\begin{aligned}
\text{Minimize}\quad & f(\mathbf{x}) = 5\sum_{n=1}^{4} x_n - 5\sum_{n=1}^{4} x_n^2 - \sum_{n=5}^{13} x_n\\
\text{Subject to}\quad & g_1(\mathbf{x}) \equiv 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0,\\
& g_2(\mathbf{x}) \equiv 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0,\\
& g_3(\mathbf{x}) \equiv 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0,\\
& g_4(\mathbf{x}) \equiv -8x_1 + x_{10} \le 0,\\
& g_5(\mathbf{x}) \equiv -8x_2 + x_{11} \le 0,\\
& g_6(\mathbf{x}) \equiv -8x_3 + x_{12} \le 0,\\
& g_7(\mathbf{x}) \equiv -2x_4 - x_5 + x_{10} \le 0,\\
& g_8(\mathbf{x}) \equiv -2x_6 - x_7 + x_{11} \le 0,\\
& g_9(\mathbf{x}) \equiv -2x_8 - x_9 + x_{12} \le 0,\\
& 0 \le x_n \le 1, \quad n = 1,2,\ldots,9,\\
& 0 \le x_n \le 100, \quad n = 10,11,12,\\
& 0 \le x_{13} \le 1.
\end{aligned}$$

The global solution to TP4 is

$\mathbf{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$, $f(\mathbf{x}^*) = -15$.

The optimal settings of the parameters in the AIS method were rs = 100 and p_{rt} = 0.9, and Table V summarizes the best numerical results for TP4. The best solution obtained by the proposed AIS approach was

$\mathbf{x}^*_{\mathrm{AIS}} =$ (0.99996543, 0.99995683, 0.99994412, 0.99999796, 0.99991012, 0.99987109, 0.99995193, 0.99984601, 0.99989468, 2.99971215, 2.99943694, 2.99917334, 0.99996526), $f(\mathbf{x}^*_{\mathrm{AIS}}) = -14.997$.

Each constraint was met as follows:

$g_m(\mathbf{x}^*_{\mathrm{AIS}}) =$ (−0.001006, −0.001295, −0.001588, −5.000011, −5.000218, −5.000380, −0.000194, −0.000257, −0.000413).

TABLE V
The best numerical results for TP 4

Parameter settings	Best	Median	Worst	MCT
rs = 100 p_{rt} = 0.9	−14.997	−14.993	−14.988	24.67 sec.

E. Comparison

Table VI compares the numerical results of the proposed AIS method with those obtained using published GAs. In Table VI, the '−' indicates that information is not available; GA-1 is a GA with penalty function methods, as used by Michalewicz and Schoenauer [23], and GA-2 represents a GA with a penalty function method, but without any penalty parameter, as employed by Deb [24]. GA-1 was run ten times and GA-2 was executed 50 times, to obtain the best, median and worst for each tested CGO problems. Table VI indicates that the proposed AIS approach converged to a global optimum solution to each tested CGO problem, and that the best, median and worst results obtained using the AIS method were superior to those obtained by GA-1 for TPs 1 and 3, and that the worst results obtained using the AIS approach were better than those obtained by GA-2 for TPs 1, 2 and 4.

Coello Coello and Cruz Cortés [13] presented a constraint-handling technique based on AIS, which was embedded in a GA, to solve TP2−TP4. They reported the best, mean and worst solutions. Table VII compares the results of the proposed AIS method with those of the AIS approach (here called AIS-1) that was presented by Coello Coello and Cruz Cortés. The table shows that the best, mean and worst results obtained using the proposed AIS were better than those obtained using the AIS-1 for TP2 and TP3, and that the mean and worst results obtained by the proposed AIS were superior to those obtained by AIS-1 for TP4.

According to the No Free Lunch theorem [25], if algorithm A outperforms algorithm B on average for one class of problems, then the former must be worse than the latter on average over the remaining problems. Therefore, no unique stochastic global optimization approach is likely available to perform the best for all CGO problems.

F. Summary of Results

The performance of the proposed AIS approach is summarized as follows:

1. Effectiveness: The method yields the global solution to each tested CGO problem and each constraint is satisfied.
2. Efficiency: The method takes an acceptable CPU computational time, as shown in Tables I−V.
3. Ease of use: The method is easy to implement. This study recommends the following settings rs = 100, p_{rt} = 0.9, $\rho_0 = 1\times10^9$, $\kappa = 50$, $c_1 = 1.5$ and $c_2 = 10$.

VI. CONCLUSION

This study presented a simply artificial immune system (AIS) based on the metaphor of biological immune system, and used it to solve four constrained global optimization (CGO) problems. Numerical results show that the proposed AIS method was effective and efficient in solving each tested problem. It can be employed as a stochastic global optimization tool and may be applied to the CGO problems that cannot be easily solved by local optimization techniques or complex deterministic global methods.

TABLE VI
Comparison of results of the proposed AIS approach and those of the published GAs

TP No.	Global Optimum	Methods	Best	Median	Worst
1	24.306	GA-1 [23]	24.690	29.258	36.060
		GA-2[24]	24.372	24.409	25.075
		the proposed AIS	24.377	24.663	24.988
2	−30665.539	GA-1 [23]	−	−	−
		GA-2 [24]	−30665.537	−30665.535	−29846.654
		the proposed AIS	−30665.543	−30665.524	−30665.500
3	680.630	GA-1 [23]	680.642	680.718	680.955
		GA-2 [24]	680.634	680.642	680.651
		the proposed AIS	680.632	680.640	680.651
4	−15.000	GA-1 [23]	−15.000	−15.000	−15.000
		GA-2 [24]	−15.000	−15.000	−13.000
		the proposed AIS	−14.997	−14.993	−14.988

TABLE VII
Comparison of results of the proposed AIS approach and those of the AIS-1 method

TP No.	Global Optimum	Methods	Best	Mean	Worst
2	−30665.539	the proposed AIS	−30665.543	−30665.522	−30665.500
		AIS-1 [13]	−30665.00	−30662.32	−30652.21
3	680.630	the proposed AIS	680.632	680.641	680.651
		AIS-1 [13]	680.750	681.666	683.258
4	−15.000	the proposed AIS	−14.997	−14.993	−14.988
		AIS-1 [13]	−14.998	−14.820	−12.993

REFERENCES

[1] P. Xu, "A hybrid global optimization: the multi-dimensional case," *J. Comput. Appl. Math.* vol. 155, pp. 423-446, 2003.
[2] C. A. Floudas, *Deterministic Global Optimization.* Boston: Kluwer, 1999.
[3] N. Metropolis, A. W. Rosenbluth et al., "Equation of state calculations by fast computing machines," *J. Chem. Phys.* vol. 12, pp. 1087-1092, 1953.
[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi Jr., "Optimization by simulated annealing," *Science,* vol. 220, pp. 671-680, 1983.
[5] H. Szu, and R. Hartley, "Fast simulated annealing," *Phys. Lett. A.* vol. 122, pp. 157-162, 1987.
[6] L. Ingber, "Very fast simulated re-annealing," *Math. Comput. Model.* vol. 12, pp. 967-973, 1989.
[7] P. Xu, "A hybrid global optimization method: the one-dimensional case," *J. Comput. Appl. Math.* vol. 147, pp. 301-314, 2002.
[8] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms," in *Chapter 4 Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies.* pp. 97-125, New York: Springer, 2005.

[9] I. Tazawa, S. Koakutsu, and H. Hirata, "An immunity based genetic algorithm and its application to the VLSI floorplan design problem," in Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, pp. 417-421, 1996.

[10] L. N. de Castro, "Immune, swarm, and evolutionary algorithms: part I: basic models," in Proceedings of the Ninth International Conference on Neural Information Processing, Singapore, vol. 3, pp. 1464-1468, 2002.

[11] L. N. de Castro, and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evolut. Comput.* vol. 6, pp. 239-251, 2002.

[12] L. N. de Castro, and J. Timmis, "An artificial immune network for multimodal function optimization," in Proceedings of the congress on Evolutionary computation, Honolulu, HI, USA, pp. 699-704, 2002.

[13] C. A. Coello Coello, and N. Cruz Cortés, "A parallel implementation of an artificial immune system to handle constraints in genetic algorithms: preliminary results," in Proceedings of the congress on Evolutionary computation, Honolulu, HI, USA, pp. 819-824, 2002.

[14] P. W. Janice, "The immune system," Seminars in Oncology Nursing, vol. 12, pp. 89-96, 1996.

[15] L. N. de Castro, and F. J. Von Zuben. (1999) Artificial Immune Systems: Part Ⅰ—Basic Theory and Applications. FEEC/Univ. Campinas, Campinas, Brazil. [Online]. Available: http://www.dca.fee.unicamp.br/~lnunes/immune.html

[16] N. K. Jerne, "Idiotypic networks and other preconceived ideas," *Immunol. Rev.* no. 79, pp. 5-24, 1984.

[17] A. S. Perelson, "Immune network theory," *Immunol. Rev.* no. 110, pp. 5-36, 1989.

[18] R. E. Langman, and M. Cohn, "The 'complete' idiotype network is an absurd immune system," *Immunol. Today,* vol. 7, pp. 100-101, 1986.

[19] A. J. T. George, and D. Gray, "Receptor editing during affinity maturation," *Immunol. Today,* vol. 20, p. 196, 1999.

[20] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Comput. Method Appl. M.* vol. 191, pp. 1245-1287, 2002.

[21] L. N. de Castro, and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. New York: Springer, 2002.

[22] C. R. Houck, J. A. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," North Carolina State Univ., Raleigh, NC, Technical Report, NSCU-IE TR 95-09, 1995.

[23] Z. Michalewicz, "Genetic algorithm, numerical optimization, and constraints," in Proceedings of the Sixth International Conference on Genetic Algorithms, San Mateo, CA, USA, pp. 151-158, 1995.

[24] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Method Appl. M.* vol. 186, pp. 311-338, 2000.

[25] D. H. Wolpert, and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evolut. Comput.* vol. 1, pp. 67-82, 1997.