

# Research Article

# Solving Unconstrained Global Optimization Problems via Hybrid Swarm Intelligence Approaches

# Jui-Yu Wu

Department of Business Administration, Lunghwa University of Science and Technology, No. 300, Section 1, Wanshou Road, Guishan, Taoyuan County 33306, Taiwan

Correspondence should be addressed to Jui-Yu Wu; jywu@mail.lhu.edu.tw

Received 7 September 2012; Revised 3 December 2012; Accepted 4 December 2012

Academic Editor: Baozhen Yao

Copyright © 2013 Jui-Yu Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Stochastic global optimization (SGO) algorithms such as the particle swarm optimization (PSO) approach have become popular for solving unconstrained global optimization (UGO) problems. The PSO approach, which belongs to the swarm intelligence domain, does not require gradient information, enabling it to overcome this limitation of traditional nonlinear programming methods. Unfortunately, PSO algorithm implementation and performance depend on several parameters, such as cognitive parameter, social parameter, and constriction coefficient. These parameters are tuned by using trial and error. To reduce the parametrization of a PSO method, this work presents two efficient hybrid SGO approaches, namely, a real-coded genetic algorithm-based PSO (RGA-PSO) method and an artificial immune algorithm-based PSO (AIA-PSO) method. The specific parameters of the internal PSO algorithm are optimized using the external RGA and AIA approaches, and then the internal PSO algorithm is applied to solve UGO problems. The performances of the proposed RGA-PSO and AIA-PSO algorithms are then evaluated using a set of benchmark UGO problems. Numerical results indicate that, besides their ability to converge to a global minimum for each test UGO problem, the proposed RGA-PSO and AIA-PSO algorithms. Thus, the RGA-PSO and AIA-PSO approaches can be considered alternative SGO approaches for solving standard-dimensional UGO problems.

# 1. Introduction

An unconstrained global optimization (UGO) problem can generally be formulated as follows:

Minimize 
$$f(\mathbf{x})$$
,  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T \in \mathfrak{R}^N$ , (1)

where  $f(\mathbf{x})$  is an objective function and  $\mathbf{x}$  represents a decision variable vector. Additionally,  $\mathbf{x} \in S, S \subseteq \mathfrak{R}^N$  denotes search space (*S*), which is *N* dimensional and bounded by parametric constraints as follows:

$$x_n^l \le x_n \le x_n^u, \quad n = 1, 2, \dots, N,$$
 (2)

where  $x_n^l$  and  $x_n^u$  are the lower and upper boundaries of the decision variables  $x_n$ , respectively.

Many conventional nonlinear programming (NLP) techniques, such as the golden search, quadratic approximation, Nelder-Mead, steepest descent, Newton, and conjugate gradient methods, have been used to solve UGO problems [1]. Unfortunately, such NLP methods have difficulty in solving UGO problems when an objective function of an UGO problem is nondifferential. Many stochastic global optimization (SGO) approaches developed to overcome this limitation of the traditional NLP methods include genetic algorithms (GAs), particle swarm optimization (PSO), ant colony optimization (ACO), and artificial immune algorithms (AIAs). For instance, Hamzaçebi [2] developed an enhanced GA incorporating a local random search algorithm for eight continuous functions. Furthermore, Chen [3] presented a two-layer PSO method to solve nine UGO problems. Zhao [4] presented a perturbed PSO approach for 12 UGO problems. Meanwhile, Toksari [5] developed an ACO algorithm for solving UGO problems. Finally, Kelsey and Timmis [6] presented an AIA method based on the clonal selection principle for solving 12 UGO problems.

This work focuses on a PSO algorithm, based on it is being effective, robust and easy to use in the SGO methods. Research on the PSO method has considered many critical issues such as parameter selection, integration of the PSO algorithm with the approaches of self-adaptation, and integration with other intelligent optimizing methods [7]. This work surveys two issues: first is a PSO approach that integrates with other intelligent optimizing methods and second is parameter selection for use in a PSO approach.

Regarding the first issue, the conventional PSO algorithm lacks evolution operators of GAs, such as crossover and mutation operations. Therefore, PSO has premature convergence, that is, a rapid loss of diversity during optimization [4]. To overcome this limitation, many hybrid SGO methods have been developed to create diverse candidate solutions to enhance the performance of a PSO approach. Hybrid algorithms have some advantages; for instance, hybrid algorithms outperform individual algorithms in solving certain problems and thus can solve general problems more efficiently [8]. Kao and Zahara [9] presented a hybrid GA and PSO algorithm to solve 17 multimodal test functions. Their study used the operations of GA and PSO methods to generate candidate solutions to improve solution quality and convergence rates. Furthermore, Shelokar et al. [10] presented a hybrid PSO and ACO algorithm to solve multimodal continuous optimization problems. Their study used an ACO algorithm to update the particle positions to enhance a PSO algorithm performance. Chen et al. [11] presented a hybrid PSO and external optimization based on the Bak-Sneppen model to solve unimodal and multimodal benchmark problems. Furthermore, Thangaraj et al. [12] surveyed many algorithms that combine the PSO algorithm with other search techniques and compared the performances obtained using hybrid differential evolution PSO (DE-PSO), adaptive mutation PSO (AMPSO), and hybrid GA and PSO (GA-PSO) approaches to solve nine conventional benchmark problems.

Regarding the second issue, a PSO algorithm has numerous parameters that must be set, such as cognitive parameter, social parameter, inertia weight, and constriction coefficient. Traditionally, the optimal parameter settings of a PSO algorithm are tuned based on trial and error. The abilities of a PSO algorithm to explore and exploit are constrained to optimum parameter settings [13, 14]. Therefore, Jiang et al. [15] used a stochastic process theory to analyze the parameter settings (e.g., cognitive parameter, social parameter, and inertia weight) of a standard PSO algorithm.

This work focuses on the second issue related to the application of a PSO method. Fortunately, the optimization of parameter settings for a PSO algorithm can be viewed as an UGO problem. Moreover, real-coded GA (RGA) and AIA are efficient SGO approaches for solving UGO problems. Based on the advantage of a hybrid algorithm [8], this work develops two hybrid SGO approaches. The first approach is a hybrid RGA and PSO (RGA-PSO) algorithm, while the second one is a hybrid AIA and PSO (AIA-PSO) algorithm. The proposed RGA-PSO and AIA-PSO algorithms are considered as a means of solving the two optimization problems simultaneously. The first UGO problem (optimization of cognitive parameter, social parameter, and constriction coefficient) is

optimized using external RGA and AIA approaches, respectively. The second UGO problem is then solved using the internal PSO algorithm. The performances of the proposed RGA-PSO and AIA-PSO algorithms are evaluated using a set of benchmark UGO problems and compared with those of many hybrid algorithms [9, 10, 12].

The rest of this paper is organized as follows. Section 2 describes RGA, PSO, and AIA approaches. Section 3 then presents the proposed RGA-PSO and AIA-PSO methods. Next, Section 4 compares the experimental results of the proposed RGA-PSO and AIA-PSO approaches with those of many hybrid methods. Conclusions are finally drawn in Section 5.

### 2. Related Works

The SGO approaches such as RGA, PSO, and AIA [16] are described as follows.

2.1. Real-Coded Genetic Algorithm. GAs are based on the concepts of natural selection and use three genetic operations, that is, selection, crossover, and mutation, to explore and exploit the solution space. In solving continuous function optimization problems, RGA method outperforms binary-coded GA approach [17]. Therefore, this work describes operators of a RGA method [18].

2.1.1. Selection Operation. A selection operation picks up strong individuals from a current population based on their fitness function values and then reproduces these individuals into a crossover pool. Many selection operations developed include the roulette wheel, the ranking, and the tournament methods [17, 18]. This work employs the normalized geometric ranking method as follows:

$$p_j = q'(1-q)^{\operatorname{rank}-1}, \quad j = 1, 2, \dots, \operatorname{ps}_{\operatorname{RGA}},$$
 (3)

where  $p_j$  = probability of selecting individual j, q = probability of choosing the best individual (here q = 0.35),  $q' = q/(1 - (1 - q)^{ps_{RGA}})$ , and rank = individual ranking based on fitness value, where 1 represents the best, rank = 1, 2, ..., ps<sub>RGA</sub>, and ps<sub>RGA</sub> = population size of the RGA method.

*2.1.2. Crossover Operation.* While exploring the solution space by creating new offspring, the crossover operation randomly chooses two parents from the crossover pool and then uses these two parents to create two new offspring. This operation is repeated until the  $ps_{RGA}/2$  is satisfied. The whole arithmetic crossover is easily performed as follows:

$$\mathbf{v}_{1}' = \beta \times \mathbf{v}_{1} + (1 - \beta) \times \mathbf{v}_{2},$$
  
$$\mathbf{v}_{2}' = (1 - \beta) \times \mathbf{v}_{1} + \beta \times \mathbf{v}_{2},$$
  
(4)

where  $\mathbf{v}_1$  and  $\mathbf{v}_2$  = parents (decision variable vectors),  $\mathbf{v}'_1$  and  $\mathbf{v}'_2$  = offspring (decision variable vectors), and  $\beta$  = uniform random number in the interval [0, 1.5].

2.1.3. Mutation Operation. Mutation operation can improve the diversity of individuals (candidate solutions). Multi-nonuniform mutation is described as follows:

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x_n^u - x_{\text{current},n}) \operatorname{pert}(g_{\text{RGA}}) \\ & \text{if } U_1(0,1) < 0.5, \\ x_{\text{current},n} - (x_{\text{current},n} - x_n^l) \operatorname{pert}(g_{\text{RGA}}) \\ & \text{if } U_1(0,1) \ge 0.5, \end{cases}$$
(5)

where pert( $g_{\text{RGA}}$ ) =  $[U_2(0, 1)(1 - g_{\text{RGA}}/g_{\text{max,RGA}})]^2$ , perturbed factor,  $U_1(0, 1)$  and  $U_2(0, 1)$  = uniform random variable in the interval [0, 1],  $g_{\text{max,RGA}}$  = maximum generation of the RGA method,  $g_{\text{RGA}}$  = current generation of the RGA method,  $x_{\text{current},n}$  = current decision variable  $x_n$ , and  $x_{\text{trial},n}$ = trial decision variable (candidate solution)  $x_n$ .

2.2. Particle Swarm Optimization. Kennedy and Eberhart [19] first presented a standard PSO algorithm, which is inspired by the social behavior of bird flocks or fish schools. Like GAs, a PSO method is a population-based algorithm. A population of candidate solutions is called a particle swarm. The particle velocities can be updated by (6) as follows:

$$v_{j,n} (g_{PSO} + 1) = v_{j,n} (g_{PSO}) + c_1 r_{j,1} (g_{PSO})$$

$$\times \left[ p_{j,n}^{lb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right]$$

$$+ c_2 r_{j,2} (g_{PSO}) \left[ p_{j,n}^{gb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right]$$

$$j = 1, 2, \dots, p_{SPSO}, \ n = 1, 2, \dots, N$$
(6)

where  $v_{j,n}(g_{PSO} + 1) = \text{particle velocity of decision variable } x_n$ of particle *j* at generation  $g_{PSO} + 1$ ,  $v_{j,n}(g_{PSO}) = \text{particle veloc-}$ ity of decision variable  $x_n$  of particle *j* at generation  $g_{PSO}$ ,  $c_1 = \text{cognitive parameter}$ ,  $c_2 = \text{social parameter}$ ,  $x_{j,n}(g_{PSO}) = \text{parti-}$ cle position of decision variable  $x_n$  of particle *j* at generation  $g_{PSO}$ ,  $r_{j,1}(g_{PSO})$ ,  $r_{j,2}(g_{PSO}) = \text{independent uniform numbers in the interval [0, 1] at generation <math>g_{PSO}$ ,  $p_{j,n}^{\text{gb}}(g_{PSO}) = \text{best}$ global solution at generation  $g_{PSO}$ , and  $p_{PSO} = \text{population}$ size of the PSO algorithm.

The particle positions can be obtained using (7) as follows:

$$x_{j,n} (g_{\text{PSO}} + 1) = x_{j,n} (g_{\text{PSO}}) + v_{j,n} (g_{\text{PSO}} + 1),$$
  

$$j = 1, 2, \dots, \text{ps}_{\text{PSO}}, \ n = 1, 2, \dots, N.$$
(7)

Shi and Eberhart [20] introduced a modified PSO algorithm by incorporating an inertia weight  $(\omega_{in})$  into (8) to

control the exploration and exploitation capabilities of a PSO algorithm as follows:

$$v_{j,n} (g_{PSO} + 1) = \omega_{in} v_{j,n} (g_{PSO}) + c_1 r_{j,1} (g_{PSO})$$

$$\times \left[ p_{j,n}^{lb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right]$$

$$+ c_2 r_{j,2} (g_{PSO}) \left[ p_{j,n}^{gb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right]$$

$$j = 1, 2, \dots, p_{SPSO}, \ n = 1, 2, \dots, N.$$
(8)

A constriction coefficient ( $\chi$ ) in (9) is used to balance the exploration and exploitation tradeoff [21–23] as follows:

$$v_{j,n} (g_{PSO} + 1) = \chi \left\{ v_{j,n} (g_{PSO}) + \tau_1 (g_{PSO}) \right\}$$
$$\times \left[ p_{j,n}^{lb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right]$$
$$+ \tau_2 (g_{PSO}) \left[ p_{j,n}^{gb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right] \right\}$$
$$j = 1, 2, \dots, p_{SPSO}, \ n = 1, 2, \dots, N,$$
(9)

where

$$\chi = \frac{2U_3(0,1)}{\left|2 - \tau - \sqrt{\tau(\tau - 4)}\right|},\tag{10}$$

 $U_3(0, 1) =$  uniform random variable in the interval [0, 1],  $\tau = \tau_1 + \tau_2$ ,  $\tau_1 = c_1 r_{j,1}$ ,  $\tau_1 = c_2 r_{j,2}$ .

This work considers parameters  $\omega_{in}$  and  $\chi$  to modify the particle velocities as follows:

$$v_{j,n} (g_{PSO} + 1) = \chi \left\{ \omega_{in} v_{j,n} (g_{PSO}) + c_1 r_{j,1} (g_{PSO}) \right. \\ \times \left[ p_{j,n}^{lb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right] \\ + c_2 r_{j,2} (g_{PSO}) \\ \times \left[ p_{j,n}^{gb} (g_{PSO}) - x_{j,n} (g_{PSO}) \right] \right\} \\ j = 1, 2, \dots, p_{SPSO}, \ n = 1, 2, \dots, N,$$

where  $\omega_{in} = ((g_{max,PSO} - g_{PSO})/g_{max,PSO})$ , increased  $g_{PSO}$  value reduces the  $\omega_{in}$ , and  $g_{max,PSO} = maximum$  generation of the PSO algorithm.

According to (11), the optimal values of parameters  $c_1$ ,  $c_2$ , and  $\chi$  are difficult to obtain through trial and error. This work thus optimizes these parameter settings by using RGA and AIA approaches.

2.3. Artificial Immune Algorithm. Wu [24] presented an AIA approach based on clonal selection and immune network theories to solve constrained global optimization problems. The AIA method consists of selection, hypermutation, receptor editing, and bone marrow operations. The selection operation is performed to reproduce strong antibodies (Abs). Also, diverse Abs are created using hypermutation, receptor editing, and bone marrow operations, as described in the following subsections.

2.3.1. Ab and Ag Representation. In the human system, an antigen (Ag) has multiple epitopes (antigenic determinants), which can be recognized by many Abs with paratopes (recognizers), on its surface. In the AIA approach, an Ag represents known parameters of a solved problem. The Abs are the candidate solutions (i.e., decision variables  $x_n$ , n = 1, 2, ..., N) of the solved problem. The quality of a candidate solution is evaluated using an Ab-Ag affinity that is derived from the value of an objective function of the solved problem.

*2.3.2. Selection Operation.* The selection operation controls the number of antigen-specific **Abs**. This operation is defined according to **Ab-Ag** and **Ab-Ab** recognition information as follows:

$$p_{j,\text{rec}} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{e^{d_{j,n}}},$$
$$d_{j,n} = \left| \frac{x_n^* - x_{j,n}}{x_n^*} \right|, \quad j = 1, 2, \dots, \text{rs}_{\text{AIA}}, \ n = 1, 2, \dots, N,$$
(12)

where  $p_{j,rec}$  = probability that  $\mathbf{Ab}_j$  recognizes  $\mathbf{Ab}^*$  (the best solution),  $x_n^*$  = the best  $\mathbf{Ab}^*$  with the highest  $\mathbf{Ab}$ -Ag affinity,  $x_{j,n}$  = decision variables  $x_n$  of  $\mathbf{Ab}_j$ , and  $rs_{AIA}$  = repertoire (population) size of the AIA approach.

The **Ab**<sup>\*</sup> is recognized by other **Ab**<sub>j</sub> in a current **Ab** repertoire. Large  $p_{j,rec}$  implies that **Ab**<sub>j</sub> can effectively recognize **Ab**<sup>\*</sup>. The **Ab**<sub>j</sub> with  $p_{j,rec}$  that is equivalent to or larger than the threshold degree of AIA approach  $p_{rt,AIA}$  is reproduced to generate an intermediate **Ab** repertoire.

*2.3.3. Hypermutation Operation.* The somatic hypermutation operation can be expressed as follows:

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x_n^u - x_{\text{current},n}) \operatorname{pert}(g_{\text{AIA}}), \\ & \text{if } U_4(0,1) < 0.5, \\ x_{\text{current},n} - (x_{\text{current},n} - x_n^l) \operatorname{pert}(g_{\text{AIA}}), \\ & \text{if } U_4(0,1) \ge 0.5, \end{cases}$$
(13)

where pert( $g_{AIA}$ ) = { $U_5(0, 1)(1 - g_{AIA}/g_{max,AIA})$ }<sup>2</sup> = perturbation factor,  $g_{AIA}$  = current generation of the AIA method,  $g_{max,AIA}$  = maximum generation number of the AIA method, and  $U_4(0, 1)$  and  $U_5(0, 1)$  = uniform random number in the interval [0, 1].

This operation has two tasks, that is, a uniform search and local fine tuning.

2.3.4. Receptor-Editing Operation. A receptor-editing operation is developed based on the standard Cauchy distribution C(0, 1), in which the local parameter is zero and the scale parameter is one. Receptor editing is implemented using Cauchy random variables that are created from C(0, 1), owing to their ability to provide a large jump in the **Ab-Ag** affinity landscape to increase the probability of escaping from the local **Ab-Ag** affinity landscape. Cauchy receptor editing can be defined by

$$\mathbf{x}_{\text{trial}} = \mathbf{x}_{\text{current}} + U_6(0, 1)^2 \times \boldsymbol{\sigma}, \tag{14}$$

where  $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \dots, \sigma_N]^T$ , vector of Cauchy random variables, and  $U_6(0, 1) =$  uniform random number in the interval [0, 1].

This operation is used in local fine-tuning and large perturbation.

2.3.5. Bone Marrow Operation. The paratope of an Ab can be created by recombining gene segments  $V_H D_H J_H$  and  $V_L J_L$  [25]. Therefore, based on this metaphor, diverse Abs are synthesized using a bone marrow operation. This operation randomly selects two Abs from the intermediate Ab repertoire and a recombination point from the gene segments of the paratope of the selected Abs. The selected gene segments (e.g., gene  $x_1$  of Ab<sub>1</sub> and gene  $x_1$  of the Ab<sub>2</sub>) are reproduced to create a library of gene segments. The selected gene segments in the paratope are then deleted. The new Ab<sub>1</sub> is formed by inserting the gene segment, which is gene  $x_1$  of the Ab<sub>2</sub> in the library plus a random variable created from standard normal distribution N(0, 1), at the recombination point. The details of the implementation of the bone marrow operation can be found [24].

# 3. Methods

This work develops the RGA-PSO and AIA-PSO approaches for solving UGO problems. The implementation of the RGA-PSO and AIA-PSO methods is described as follows.

*3.1. RGA-PSO Algorithm.* Figure 1 shows the pseudocode of the proposed RGA-PSO algorithm. The best parameter setting of the internal PSO algorithm is obtained by using the external RGA method. Benchmark UGO problems are solved by using the internal PSO algorithm.

### External RGA

Step 1 (initialize the parameter settings). Parameter settings such as  $ps_{RGA}$ , crossover probability of the RGA method  $p_{c,RGA}$ , mutation probability of the RGA approach  $p_{m,RGA}$ ,  $ps_{PSO}$  as well as lower and upper boundaries of the parameters  $(c_1, c_2, \text{ and } \chi)$  for a PSO algorithm are given. The candidate solution (individual) of a RGA method represents the optimized parameters of internal PSO algorithm. Figure 2 illustrates the candidate solution of the RGA method.

Step 2 (calculate the fitness function value). The fitness function value fitness, of the external RGA method is the



#### FIGURE 1: The pseudocode of the proposed RGA-PSO algorithm.

Methods	Parameter settings	Search space
External RGA	$p_{c,RGA} = 1$ $p_{m,RGA} = 0.15$ $p_{s_{RGA}} = 20$ $g_{max,RGA} = 20$	$[\chi^{l}, \chi^{u}] = [0.1, 1]$ $[c_{1}^{l}, c_{1}^{u}] = [0.1, 5]$
External AIA	$p_{\rm rt,AIA} = 0.9$ $rs_{\rm AIA} = 20$ $g_{\rm max,AIA} = 20$	$[c_2^{\prime}, c_2^{\prime \mu}] = [0.1, 5]$
Internal PSO	$p_{\text{S}_{\text{PSO}}} = 20$ $g_{\text{max},\text{PSO}} = 1000 \text{ for } N \le 3$ $g_{\text{max},\text{PSO}} = 3000 \text{ for } 5 \le N \le 30$	$[x_n^l, x_n^u]$ for a UGO problem

[able ]	l: The parameter	• settings for	the proposed	RGA-PSO and Al	A-PSO approaches
---------	------------------	----------------	--------------	----------------	------------------

Candid	ate solut	ion j	
X	$c_1$	<i>c</i> <sub>2</sub>	Fitness <sub>j</sub>

FIGURE 2: Candidate solution of the RGA method.

best objective function value  $f(\mathbf{x}_{PSO}^*)$  obtained from the best solution  $\mathbf{x}_{PSO}^*$  of each internal PSO algorithm execution as follows:

$$\operatorname{fitness}_{j} = f\left(\mathbf{x}_{\operatorname{PSO}}^{*}\right), \quad j = 1, 2, \dots, \operatorname{ps}_{\operatorname{RGA}}.$$
 (15)

The candidate solution j of the external RGA method is incorporated into the internal PSO algorithm and, then, the internal PSO algorithm is used to solve an UGO problem. The internal PSO algorithm is executed as follows.

## Internal PSO Algorithm

(1) Generate an initial particle swarm. An initial particle swarm is created based on  $ps_{PSO}$  from  $[x_n^l, x_n^u]$  of an UGO problem. A particle represents a candidate solution of an UGO problem.

```
Procedure external AIA method
   begin
      g_{\text{AIA}} \leftarrow 0
      count \leftarrow 0
                                                                                               Procedure internal PSO algorithm
          Step 1: Initialize the parameter settings
                                                                                                  begin
             while g_{AIA} < g_{max,AIA} do
                                                                                                      g_{\text{PSO}} \leftarrow 0
                 Step 2: Evaluate the Ab-Ag affinity
                                                                     Х
                                                                             c_1
                                                                                    C_2
                                                                                                      (1) Generate an initial particle swarm
                         \mathbf{Ab}^* \leftarrow \max(\operatorname{affnity}_j), j = 1, 2, \dots, \operatorname{rs} \left[ -1 \times \mathbf{x}_{PSO}^* \right]
                                                                                                       (a) parameter settings obtained from external AIA
                 Step 3: Perform a clonal selection operation
                                                                                                       (b) generate an initial particle swarm
                        for each \mathbf{Ab}_{j}, j = 1, 2, \dots, rs_{AIA}do
                                                                                                       while g_{PSO} \leq g_{max,PSO} do
                            if p_{j,rec} \ge p_{rt,AIA} then
                                                                                                      (2) Compute the fitness value
                               promote (clone)
                                                                                                      (3) Update the particle velocity and position
                            else
                                                                                                      (4) Perform an elitist strategy
                               suppress
                                                                                                      g_{\text{PSO}} \leftarrow g_{\text{PSO}} + 1
                           endIf
                                                                                                      endWhile
                        endFor
                                                                                                  end
                 Step 4: Implement an Ab-Ag affinity maturation operation
                                                                                               end
                        for each promoted Ab<sub>i</sub> do
                           if rand() \leq 0.5 do
                               somatic hypermutation
                            else
                               receptor editing
                           endIf
                        endFor
                 Step 5: Introduce diverse Abs
                 Step 6: Update an Ab repertoire
                     if |\operatorname{affinity}(x_n,\chi,c_1,c_2,g_{AIA}+1) - \operatorname{affinity}(x_n,\chi,c_1,c_2,g_{AIA})| \le 1 \times 10^{-9} then
                       count \leftarrow count + 1
                     endIf
                     if count \leq 5 then
                        break
                    endIf
                 g_{\text{AIA}} \leftarrow g_{\text{AIA}} + 1
          endWhile
   end
end
```

FIGURE 3: The pseudocode of the proposed AIA-PSO algorithm.

- (2) Compute the objective function value. The objective function value of the internal PSO algorithm of particle  $jf(\mathbf{x}_{j,PSO})$   $j = 1, 2, ..., ps_{PSO}$  is the objective function value of an UGO problem.
- (3) Update the particle velocity and position. Equations (7) and (11) can be used to update the particle position and velocity.
- (4) Perform an elitist strategy. A new particle swarm is generated from internal step (3). Notably,  $f(\mathbf{x}_{i,PSO})$ of a candidate solution j (particle j) in the particle swarm is evaluated. This work makes a pairwise comparison between the  $f(\mathbf{x}_{j,PSO})$  of candidate solutions in the new particle swarm and that in the current particle swarm. A situation in which the candidate solution j ( $j = 1, 2, ..., ps_{PSO}$ ) in the new particle swarm is better than candidate solution *j* in the current particle swarm implies that the strong candidate solution j in the new particle swarm replaces the candidate solution j in the current particle swarm. The elitist strategy guarantees that the best candidate solution is always preserved in the next generation. The current particle swarm is updated to the particle swarm of the next generation.

Internal steps from (2) to (4) are repeated until the maximum generation number of the PSO method  $g_{max,PSO}$  of the internal PSO algorithm is satisfied.

#### End

*Step 3* (perform a selection operation). Equation (3) is used to select the parents into a crossover pool.

*Step 4* (implement a crossover operation). The crossover operation performs a global search. The candidate solutions are created by using (4).

*Step 5* (conduct a mutation operation). The mutation operation implements a local search. A solution space is exploited using (5).

Step 6 (perform an elitist strategy). This work presents an elitist strategy to update the population. A situation in which the fittness<sub>j</sub> of candidate solution j in the new population is larger than the fittness<sub>j</sub> of candidate solution j in the current population suggests that a replacement of the weak candidate solution j takes place. Additionally, a situation in which the fittness<sub>j</sub> of candidate solution j in the new population is equal to or worse than that in the current population survives. In addition to maintaining the strong candidate solutions, this strategy effectively eliminates weak candidate solutions.

6



FIGURE 4: Ab and Ag representation.

External steps 2 to 6 are repeated until the  $g_{\max,RGA}$  of the external RGA method is met.

*3.2. AIA-PSO Algorithm.* Figure 3 shows the pseudocode of the proposed AIA-PSO algorithm. The external AIA method is used to optimize the parameter settings of the internal PSO method, which is employed to solve benchmark UGO problems.

#### External AIA

*Step 1* (initialize the parameter settings). Several parameters must be predetermined. These include repertoire (population) size  $r_{AIA}$  and  $p_{rt,AIA}$ . An available **Ab** repertoire (population) is randomly generated using  $r_{AIA}$  from the lower and upper boundaries of parameters  $\chi [\chi^l, \chi^u]$ ,  $c_1 [c_1^l, c_1^u]$ , and  $c_2 [c_2^l, c_2^u]$ . Figure 4 shows the **Ab** and **Ag** representation. *Step 2* (evaluate the **Ab**-**Ag** affinity).

*Internal PSO Algorithm.* The external AIA approach offers parameter settings  $c_1$ ,  $c_2$ , and  $\chi$  for the internal PSO algorithm. Subsequently, internal steps (1)–(4) of the PSO algorithm are implemented. The internal PSO method returns the best fitness value of PSO  $f(\mathbf{x}_{PSO}^*)$  to the external AIA method.

- Generate an initial particle swarm. An initial particle swarm is created based on ps<sub>PSO</sub> from [x<sup>l</sup><sub>n</sub>, x<sup>u</sup><sub>n</sub>] of an UGO problem. A particle represents a candidate solution of an UGO problem.
- (2) *Compute the fitness value.* The fitness value of the internal PSO algorithm  $f(\mathbf{x}_{PSO,j})$   $j = 1, 2, ..., ps_{PSO}$  is the objective function value of an UGO problem.

- (3) Update the particle velocity and position. Equations (7) and (11) can be used to update the particle position and velocity.
- (4) *Perform an elitist strategy.* A new particle swarm (population) is generated from internal step (3). Notably,  $f(\mathbf{x}_{j,PSO})$  of a candidate solution *j* (particle *j*) in the particle swarm is evaluated. This work makes a pairwise comparison between the  $f(\mathbf{x}_{j,PSO})$  of candidate solutions in the new particle swarm and that in the current particle swarm. The elitist strategy guarantees that the best candidate solution is always preserved in the next generation. The current particle swarm of the next generation.

Internal steps from (2) to (4) are repeated until the  $g_{\text{max,PSO}}$  of the internal PSO algorithm is satisfied.

#### End

Consistent with the **Ab-Ag** affinity metaphor, an **Ab-Ag** affinity is determined using (16) as follows:

$$\max\left(\operatorname{affinity}_{j}\right) = -1 \times f\left(\mathbf{x}_{PSO}^{*}\right) \quad j = 1, 2, \dots, \operatorname{rs}_{AIA}.$$
 (16)

Following the evaluation of the **Ab-Ag** affinities of **Abs** in the current **Ab** repertoire, the **Ab** with the highest **Ab-Ag** affinity  $(Ab^*)$  is chosen to undergo clonal selection in external Step 3.

*Step 3* (perform a clonal selection operation). To control the number of antigen-specific **Abs**, (12) is employed.

*Step 4* (implement an **Ab-Ag** affinity maturation operation). The intermediate **Ab** repertoire that is created in external

TD number	Function	Global	Required	Curcase rata (06)	f( <b>v</b> * )	f(wmean)	f(worst)	MF	MCCT (sec.)
	name	minimum	accuracy	Ourcess Tate (70)	J (ARGA-PSO)	J \ARGA-PSO/	J \ARGA-PSO/		INTOCI (SEC.)
1	SHCB	-1.0316	1E-4	100	-1.0316	-1.0316	-1.0316	0.00000000	40.56
2	GP	3	1E - 3	100	33	3	3	0.00000000	38.29
3	ES	-1	1E - 3	100	-1	-1	-1	0.00000000	41.29
4	B2	0	1E-2	100	0.00000000	0.00000000	0.00000000	0.00000000	40.26
Ŋ	DJ	0	1E-4	100	0.00000000	5.533E - 226	5.533E - 224	5.533E - 226	40.50
6	Booth	0	1E-4	100	0.00000000	0.00000000	0.00000000	0.00000000	37.57
7	RC	$5/4\pi$	1E - 3	100	0.39788735	0.39788735	0.39788735	0.00000000	40.97
8	RA	-2	1E-4	100	-2	-2	-2	0.00000000	39.31
6	$RS_2$	0	1E - 3	100	0.00000000	8.709E - 28	8.686E - 26	8.709E - 28	37.47
10	$RS_5$	0	1E-3	100	9.587E - 12	9.452E - 06	9.058E - 05	9.452E - 06	306.15
11	SH	-186.7309	1E-2	100	-186.7309	-186.7309	-186.7309	0.00000000	54.11
12	$ZA_2$	0	1E-4	100	0.00000000	0.00000000	0.00000000	0.00000000	40.74
13	$ZA_5$	0	1E-4	100	0.00000000	2.660E - 117	2.660E - 115	2.660E - 117	157.09
14	$ZA_{10}$	0	1E - 3	100	2.453E - 120	4.375E - 26	2.933E - 24	4.375E - 26	194.73
$f(\mathbf{x}^*_{\mathrm{RGA-PSO}})$ : the	objective value ob	stained using the best	RGA-PSO solution.						

TABLE 2: Numerical results obtained from the proposed RGA-PSO algorithm for solving 14 UGO problems.

8

 $f(\mathbf{x}_{\rm RGA-PSO}^{\rm mean})$  : mean objective value obtained using the RGA-PSO solutions.

 $f(\mathbf{x}_{\rm RGA-PSO}^{\rm worst})$  : the worst objective value obtained using the worst RGA-PSO solution.

Step 3 is divided into two subsets. These **Abs** undergo somatic hypermutation operation by using (13) when the random number is 0.5 or less. Notably, these **Abs** suffer receptorediting operation using (14) when the random number exceeds 0.5.

*Step 5* (introduce diverse **Abs**). Based on the bone marrow operation, diverse **Abs** are created to recruit the **Abs** suppressed in external Step 3.

Step 6 (Update an Ab repertoire). A new Ab repertoire is generated from external Steps 3–5. The Ab-Ag affinities of the Abs in the generated Ab repertoire are evaluated. This work presents a strategy for updating the Ab repertoire. A situation in which the Ab-Ag affinity of  $Ab_j$  in the new Ab repertoire exceeds that in the current Ab repertoire implies that a strong Ab in the new Ab repertoire replaces the weak Ab in the current Ab repertoire. Additionally, a situation in which the Ab-Ag affinity of  $Ab_j$  in the new Ab repertoire equals to or is worse than that in the current Ab repertoire implies that the  $Ab_j$  in the current Ab repertoire survives. In addition to maintaining the strong Abs, this strategy eliminates nonfunctional Abs

Repeat external Steps 2–6 until the termination criterion  $g_{\text{max,AIA}}$  is satisfied.

#### 4. Results

The proposed RGA-PSO and AIA-PSO algorithms were applied to a set of benchmark UGO problems taken from other studies [9, 10, 17, 26], as detailed in the Appendix. The proposed RGA-PSO and AIA-PSO approaches were coded in MATLAB software and run on a Pentium D 3.0 (GHz) personal computer. One-hundred independent runs were conducted for each test problem (TP). To have comparable numerical results, the accuracy was chosen based on the numerical results reported in [9, 10, 17, 26]. Numerical results were summarized, including rate of successful minimizations (success rate %), best mean worst, mean computational CPU time (MCCT), and mean error ME (average value of the gap between the objective function values calculated using the AIA-PSO and RGA-PSO solutions and the known global minimum value). Table 1 lists the parameter settings for the proposed RGA-PSO and AIA-PSO approaches. The table shows 20,000 ( $20 \times 1000$ ) objective function evaluations of the internal PSO approach for an UGO problem with  $N \leq 3$ decision variables and 60,000 ( $20 \times 3000$ ) objective function evaluations of the internal PSO for an UGO problem with  $5 \le N \le 30$  decision variables. Moreover, the external AIA and RGA methods stop when  $g_{\text{max},\text{RGA}} = 20$  and  $g_{\text{max},\text{AIA}} = 20$ are met or the best fitness value of the RGA approach (or the best Ab-Ag affinity of the AIA method) does not significantly change for the past five generations.

4.1. Numerical Results Obtained Using the RGA-PSO and AIA-PSO Algorithms for Low-Dimensional UGO Problems ( $2 \le N \le 10$ ). Table 2 lists the numerical results obtained using the proposed RGA-PSO algorithm. The numerical results indicate that the RGA-PSO algorithm can obtain the global minimum for each test UGO problem since these MEs equal or closely approximate "0," and the RGA-PSO algorithm has an acceptable MCCT for each TP. Table 3 lists the optimal parameter settings obtained using the proposed RGA-PSO algorithm to solve 14 UGO problems.

Table 4 lists the numerical results obtained using the proposed AIA-PSO algorithm. Numerical results indicate that the AIA-PSO algorithm can obtain the global minimum for each test UGO problem since these MEs equal or closely approximate "0," and that the AIA-PSO algorithm has an acceptable MCCT for each TP. Table 5 lists the optimal parameter settings obtained using the proposed AIA-PSO algorithm for solving 14 UGO problems.

4.2. Numerical Results Obtained Using the RGA-PSO and AIA-PSO Algorithms for a Standard-Dimensional UGO Problem (N = 30). To investigate the effectiveness of the RGA-PSO and AIA-PSO methods for solving a standard-dimensional UGO problem, the Zakharov problem with 30 decision variables (ZA<sub>30</sub>), as described in the Appendix, has been solved using the RGA-PSO and AIA-PSO approaches. Fifty independent runs were performed to solve the UGO problem. To increase the diversity of candidate solutions for use in the external RGA method, the parameter  $p_{m,RGA}$  was set from 0.15 to "1." Table 6 lists the numerical results obtained using the RGA-PSO and AIA-PSO approaches. This table indicates that the two approaches converge to the global optimum value, since the MEs closely approximate "0," and the MCCT of the RGA-PSO method is larger than that of the AIA-PSO method. Moreover, the success rates of the proposed RGA-PSO and AIA-PSO approaches are 100%. The Wilcoxon test is performed to the difference of median values of the MEs obtained using the RGA-PSO and AIA-PSO methods. The P value of the Wilcoxon test is 0.028, which is smaller than the significance level of 0.05, indicating that the performance of the RGA-PSO method is statistically different to that of the AIA-PSO method. Table 7 summarizes the optimal parameter settings obtained using the proposed RGA-PSO and AIA-PSO algorithms for the UGO problem ZA<sub>30.</sub>

The UGO problem  $ZA_{50}$  was solved using the RGA-PSO and AIA-PSO methods. The RGA-PSO and AIA-PSO methods fail to solve the UGO problem, since the diversity of the particle swarm in the internal PSO method cannot be maintained. Hence, future work will focus on improving the diversity of the particle swarm by applying mutation operations.

4.3. Comparison. Table 8 lists the results of the Wilcoxon test for the *MEs* obtained using the proposed RGA-PSO and AIA-PSO methods for 14 UGO problems. In this table, the "\*\*" represents the *P* value of Wilcoxon test which cannot be obtained, since the MEs obtained using the RGA-PSO and AIA-PSO methods for a TP are identical. Moreover, the median values of the *MEs* obtained using the RGA-PSO and AIA-PSO methods for TP 10 are not statistically different, since their *P* value is larger than the significance level of 0.05. Overall, the performances obtained using the RGA-PSO and AIA-PSO methods are statistically identical.

TP number	Function name	χ	<i>c</i> <sub>1</sub>	<i>c</i> <sub>2</sub>
1	SHCB	0.36497798	1.79972890	2.36717639
2	GP	0.86138264	3.74964714	0.10000000
3	ES	0.36038653	0.42043804	4.78367612
4	B2	0.75486133	0.17875492	0.95144705
5	DJ	0.39357143	3.17898242	4.72512283
6	Booth	0.56703400	4.84740898	2.58109482
7	RC	0.44153938	2.71748076	3.81076900
8	RA	0.93533829	0.76530619	3.53402840
9	RS <sub>2</sub>	0.87003776	1.28577955	2.77423057
10	RS <sub>5</sub>	0.62553666	0.10000000	5.00000000
11	SH	0.45903159	1.07452631	4.15038556
12	$ZA_2$	0.75044069	0.50883133	1.93860581
13	$ZA_5$	0.52013863	1.05471011	4.61197390
14	$ZA_{10}$	0.56126516	0.39456605	4.95735921

TABLE 3: The optimal parameter settings obtained using the proposed RGA-PSO algorithm for solving 14 UGO problems.

Table 9 compares the numerical results obtained using the RGA-PSO and AIA-PSO methods with those obtained using the hybrid algorithms for 11 TPs. Specifically, the table lists the numerical results obtained using the Nelder-Mead simplex search and PSO (NM-PSO) and GA-PSO taken from [9], those obtained using particle swarm ant colony optimization (PSACO), continuous hybrid algorithm (CHA) and continuous tabu simplex search (CTSS) taken from [10], and those obtained using DE-PSO, AMPSO1, and AMPSO2 taken from [12]. Table 9 indicates that the RGA-PSO and AIA-PSO methods yield superior accuracy of MEs obtained using the NM-PSO, GA-PSO, DE-PSO, AM-PSO1, AM-PSO2, CHA and CTSS approaches for TPs 2, 3, 4, 5, 7, 9, 11, and 12 and that RGA-PSO and AIA-PSO approaches yield superior accuracy of MEs obtained using the PSACO method for TPs 4, 5, 7, 9, 10, 11, 12, 13 and 14. Table 10 compares the percentage success rates of the proposed RGA-PSO and AIA-PSO approaches and those of the hybrid algorithms for 11 TPs, indicating that all algorithms except for the CHA and CTSS methods achieved identical performance (100% success rate) for all TPs.

4.4. Summary of Results. The proposed RGA-PSO and AIA-PSO algorithms have the following benefits.

- (1) Parameter manipulation of the internal PSO algorithm is based on the solved UGO problems. Owing to their ability to efficiently solve UGO problems, the external RGA and AIA approaches are substituted for trial and error to manipulate the parameters ( $\chi$ ,  $c_1$ , and  $c_2$ ).
- (2) Besides obtaining the optimum parameter settings of the internal PSO algorithm, the RGA-PSO and AIA-PSO algorithms can yield a global minimum for an UGO problem.
- (3) Beside, outperforming some published hybrid SGO methods, the proposed RGA-PSO and AIA-PSO

approaches reduce the parametrization for the internal PSO algorithm, despite being more complex than individual SGO approaches.

The proposed RGA-PSO and AIA-PSO algorithms are limited in that they cannot solve high-dimensional UGO problems (such as N = 50). Future work will focus on increasing the diversity of the particle swarm of the internal PSO method by applying mutation to solve high-dimensional UGO problems.

# 5. Conclusions

This work developed RGA-PSO and AIA-PSO algorithms. Performances of the proposed RGA-PSO and AIA-PSO approaches were evaluated using a set of benchmark UGO problems. Numerical results indicate that the proposed RGA-PSO and AIA-PSO methods can converge to global minimum for each test UGO problem and obtain the best parameter settings of the internal PSO algorithm. Moreover, the numerical results obtained using the RGA-PSO and AIA-PSO algorithms are superior to those obtained using many alternative hybrid SGO methods. The RGA-PSO and AIA-PSO methods can thus be considered efficient SGO approaches for solving standard-dimensional UGO problems.

## Appendix

(1) Six-hump camel back (SHCB) (two variables) [17]:

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2 \quad (A.1)$$

search domain:  $-3 \le x_1 \le 3$ ;  $-2 \le x_2 \le 2$ 

one global minimum at two different points:  $\mathbf{x}^* = (-0.0898, 0.7126)$  and  $\mathbf{x}^* = (0.0898, -0.7126)$ ,  $f(\mathbf{x}^*) = -1.0316$ .

TD	Function	Global	Required	C	<i>t</i> (***) <i>T</i>	f mean	f /worst	ME	
I L IIMIIDEL	name	minimum	accuracy	ouccess rate (%)	/ OSd-VIV	/ (AIA-PSO /	/ (AIA-PSO)	INLE	MICCI (Sec)
1	SHCB	-1.0316	1E-4	100	-1.0316	-1.0316	-1.0316	0.00000000	39.69
2	GP	$\mathcal{O}$	1E-3	100	3	3	3	0.00000000	37.80
3	ES	-1	1E-3	100	-1	-1	-1	0.00000000	42.08
4	B2	0	1E-2	100	0.00000000	0.00000000	0.00000000	0.00000000	40.43
5	DJ	0	1E-4	100	0.00000000	2.544E - 215	2.544E - 213	2.544E - 215	39.43
6	Booth	0	1E-4	100	0.00000000	0.00000000	0.00000000	0.00000000	37.48
7	RC	$5/4\pi$	1E-3	100	0.39788735	0.39788735	0.39788735	0.00000000	40.53
8	RA	-2	1E-4	100	-2	-2	-2	0.00000000	39.17
6	$RS_2$	0	1E-3	100	0.00000000	2.341E - 27	2.129E - 25	2.341E - 27	37.49
10	$RS_5$	0	1E-3	100	4.342E - 10	7.291E - 06	9.313E - 05	7.291E - 06	304.23
11	SH	-186.7309	1E-2	100	-186.7309	-186.7309	-186.7309	0.00000000	54.70
12	$ZA_2$	0	1E-4	100	0.00000000	0.00000000	0.00000000	0.00000000	40.31
13	$ZA_5$	0	1E-4	100	0.00000000	7.2591E - 184	7.259E - 182	7.259E - 184	144.38
14	$ZA_{10}$	0	1E-3	100	1.472E - 120	3.287E - 42	2.856E - 40	3.287E - 42	192.06
$f(\mathbf{x}^*_{AIA-PSO})$ : the	: objective value obt	ained using the best.	AIA-PSO solution.						

TABLE 4: Numerical results obtained from the proposed AIA-PSO algorithm for solving 14 UGO problems.

 $f(\mathbf{x}_{\mathrm{MIA-PSO}}^{\mathrm{mean}})$ : mean objective value obtained using the AIA-PSO solutions.  $f(\mathbf{x}_{\mathrm{MIA-PSO}}^{\mathrm{mean}})$ : the worst objective value obtained using the worst AIA-PSO solution.

TP number	Function name	χ	<i>c</i> <sub>1</sub>	<i>c</i> <sub>2</sub>
1	SHCB	0.59403239	1.75960560	4.12826475
2	GP	0.29584147	2.62468918	4.41787729
3	ES	0.39500323	0.84589484	2.64979988
4	B2	0.64806710	1.78727677	1.83995777
5	DJ	0.45630341	1.11829588	4.10382418
6	Booth	0.60889449	2.53158038	1.32827344
7	RC	1.00000000	1.48772085	0.43921335
8	RA	0.38157302	4.93248942	4.08732827
9	RS <sub>2</sub>	0.39872504	4.65992304	4.84228028
10	RS <sub>5</sub>	0.55519288	0.10000000	4.52483714
11	SH	0.54293231	2.10545153	4.52889414
12	$ZA_2$	0.44530811	1.05474921	3.20092987
13	$ZA_5$	0.51683760	1.17129205	4.94965004
14	$ZA_{10}$	0.53688659	0.53752883	4.99881022

TABLE 5: The optimal parameter settings obtained using the proposed AIA-PSO algorithm for solving 14 UGO problems.

TABLE 6: Numerical results obtained from the proposed RGA-PSO and AIA-PSO algorithms for solving a standard-dimensional UGO problem.

Function name	Global minimum	Required accuracy	Methods	Success rate (%)	$f(\mathbf{x}^*_{\text{RGA-PSO}})$	$f(\mathbf{x}_{\text{RGA-PSO}}^{\text{mean}})$	$f(\mathbf{x}_{\text{RGA-PSO}}^{\text{worst}})$	ME	MCCT (sec)
74	0	1E - 3	RGA-PSO	100	4.186E - 13	4.537E-06	5.331E-05	4.537E-06	1753.86
2130	0	1L = J	AIA-PSO	100	6.520E - 12	8.099E - 06	9.175E-05	8.099E - 06	1453.98

(2) Goldstein-price (GP) (two variables) [9, 10]:

$$f (\mathbf{x}) = \left[ 1 + (x_1 + x_2 + 1)^2 \\ \times \left( 19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2 \right) \right] \\ \times \left[ 30 + (2x_1 - 3x_2)^2 \\ \times \left( 18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2 \right) \right]$$
(A.2)

search domain:  $-2 \le x_n \le 2$ , n = 1, 2four local minima; one global minimum:  $\mathbf{x}^* = (0, -1)$ ,  $f(\mathbf{x}^*) = 3$ .

(3) Easom (ES) (two variables) [9]:

$$f(\mathbf{x}) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2 - (x_2-\pi)^2}$$
(A.3)

search domain:  $-100 \le x_n \le 100, n = 1, 2$ several local minima (exact number unspecified in usual literature);

one global minimum:  $\mathbf{x}^* = (\pi, \pi), f(\mathbf{x}^*) = -1.$ 

(4) *B2 (two variables)* [9, 10, 17]:

$$f(\mathbf{x}) = x_1^2 + 2x_1^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$
(A.4)

search domain:  $-100 \le x_n \le 100$ , n = 1, 2several local minima (exact number unspecified in usual literature);

one global minimum  $\mathbf{x}^* = (0, 0), f(\mathbf{x}^*) = 0.$ 

(5) De Jong (DJ) (three variables) [9, 10]:

$$f(\mathbf{x}) = \sum_{n=1}^{3} x_n^2$$
 (A.5)

search domain:  $-5.12 \le x_n \le 5.12$ , n = 1, 2, 3one global minimum:  $\mathbf{x}^* = (0, 0, 0)$ ,  $f(\mathbf{x}^*) = 0$ .

(6) Booth (BO) (two variables) [26]:

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$
 (A.6)

search domain:  $-10 \le x_n \le 10, n = 1, 2$ one global minimum:  $\mathbf{x}^* = (1, 3), f(\mathbf{x}^*) = 0.$ (7) *Branin RCOC (RC) (two variables) [9, 10]*:

$$f(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2} + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$$
(A.7)

Тавle 7: The optima	l parameter settings of	obtained using the pro	oposed RGA-PSO a	nd AIA-PSO algo	orithms for solving a	standard-dime	ensional
UGO problem.							

Function name	Methods	X	$c_1$	<i>C</i> <sub>2</sub>
74	RGA-PSO	0.58721675	0.56183025	4.91401312
2A <sub>30</sub>	AIA-PSO	0.61082112	0.38556559	5.0000000

TABLE 8: Results of Wilcoxon test for the MEs obtained using the proposed RGA-PSO and AIA-PSO methods for 14 UGO problems.

TD number	Eunction name	RGA-PSO versus AIA-PSO
1 P Humber	Function name	<i>P</i> value
1	SHCB	**
2	GP	**
3	ES	**
4	B2	**
5	DJ	* *
6	Booth	**
7	RC	**
8	RA	**
9	RS <sub>2</sub>	**
10	$RS_5$	0.833
11	SH	**
12	$ZA_2$	**
13	$ZA_5$	**
14	$ZA_{10}$	**

search domain:  $-5 \le x_1 \le 10, 0 \le x_1 \le 15$ no local minimum;

three global minima:  $\mathbf{x}^* = (-\pi, 12.275), \ \mathbf{x}^* = (\pi, 2.275), \ \mathbf{x}^* = (3\pi, 2.475), \ f(\mathbf{x}^*) = 5/4\pi.$ 

(8) Rastrigin (RA) (two variables) [10]:

$$f(\mathbf{x}) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$
(A.8)

search domain:  $-1 \le x_n \le 1, n = 1, 2$ 

50 local minima;

One global minimum:  $f(\mathbf{x}^*) = -2$ ,  $\mathbf{x}^* = (0, 0)$ .

(9) Rosenbrock (RSn) (N variables) [9, 10]:

$$f(\mathbf{x}) = \sum_{n=1}^{N-1} \left[ 100 (x_n^2 - x_{n+1})^2 + (x_n - 1)^2 \right]$$
(A.9)

Two functions were considered: RS<sub>2</sub>, RS<sub>5</sub>

search domain:  $-5 \le x_n \le 10, n = 1, 2, ..., N$ 

several local minima (exact number unspecified in usual literature);

global minimum:  $\mathbf{x}^* = (1, 1), f(\mathbf{x}^*) = 0.$ 

(10) Shubert (SH) (two variables) [9, 10]:

$$f(\mathbf{x}) = \sum_{n=1}^{N-1} \left[ 100 \left( x_n^2 - x_{n+1} \right)^2 + \left( x_n - 1 \right)^2 \right]$$
(A.10)

search domain:  $-10 \le x_n \le 10, n = 1, 2$ 760 local minima; 18 global minima;  $f(\mathbf{x}^*) = -186.7309.$ 

(11) Zakharov (ZAn) (N variables) [9, 10]:

$$f(\mathbf{x}) = \sum_{n=1}^{N} x_n^2 + \left(\sum_{n=1}^{N} 0.5nx_n\right)^2 + \left(\sum_{n=1}^{N} 0.5nx_n\right)^4.$$
 (A.11)

Three functions were considered:  $ZA_2$ ,  $ZA_5$ ,  $ZA_{10}$ ,  $ZA_{30}$ 

search domain:  $-5 \le x_n \le 10, n = 1, 2, ..., N$ 

several local minima (exact number unspecified in usual literature);

global minimum:  $\mathbf{x}^* = (0, ..., 0), f(\mathbf{x}^*) = 0.$ 

# **Conflict of Interests**

The author confirm that he does not has a conflict of interest with the MATLAB software.

# Acknowledgment

The author would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract no. NSC 100-2622-E-262-006-CC3.

AIA-PSO RGA-PSO (this work) (this work) 0.0000000 0.0000000	0.0000000 0.00000000		0.0000000 0.00000000	0.0000000 0.00000000	2.544E-215 5.5331E-226	0.0000000 0.00000000	2.341E - 27 $8.709E - 28$	7.291E - 06 $9.452E - 06$	0.0000000 0.00000000	0.0000000 0.00000000	7.259E - 184 $2.660E - 117$	3.287E - 42 $4.3756E - 26$		as for 11 TPs.	
	CTSS [10	0.001	0.005	0.000005	0.0002	0.005	0.0040		0.001	3E - 7		Ι		rid algorithn	
	CHA [10]	0.0010	0.0010	0.0000002	0.0002	0.0001	0.0040		0.0050	3E-6		1E - 6		se of the hybr	
	AM-PSO2 [12]	9.89E - 6	3.71E - 5	2.64E - 5	3.34E - 5	1.49E - 5	3.33E - 5	9.93E - 5	2.54E - 7	2.61ZE - 6	5.07E - 5	9.11E - 5		pproaches and tho	ate %
INLES	AM-PSO1 [12]	3.81E - 5	4.10E - 5	1.17E - 6	4.23E - 5	1.52E - 5	2.16E - 5	9.73E - 5	1.72E - 5	2.54E - 5	3.44E - 5	7.41E - 5		O and AIA-PSO a <sub>l</sub>	Success r
	DE-PSO [12]	3.67E - 5	1.67E - 5	1.11E - 5	1.67E - 5	2.29E - 6	2.46E - 5	7.74E - 5	3.03E - 6	1.12E - 5	3.49E - 5	6.91E - 5		roposed RGA-PS	
	GA-PSO [9]	0.00012	0.00003	0.0001	0.00004	0.00009	0.00064	0.00013	0.00007	0.00005	0.00000	0.00000		s rate % of the pi	
	PSACO [10]	0.00000000	0.00000000	5.5511E - 17	7.6900E - 29	2.6185E - 13	1.7152E - 10	1.8538E - 04	1.09239E - 09	5.7061E - 27	3.6352E - 17	2E - 08		ison of the succes	
	[6] OSA-WN	0.00003	0.00004	0.00003	0.00003	0.00003	0.00003	0.0056	0.00002	0.00003	0.00026		information).	ABLE 10: Compar	
Function	name	GP	ES	B2	DJ	RC	$RS_2$	$RS_5$	SH	$ZA_2$	$ZA_5$	$ZA_{10}$	es unavailable i	Τ	Function
1	number	2	3	4	5	7	6	10	11	12	13	14	("—" denot		TP

TP	Function					Success re	ate %				
number	name	[6] OSA-MN	PSACO [10]	GA-PSO [9]	DE-PSO [12]	AM-PSO1 [12]	AM-PSO2 [12]	CHA [10]	CTSS [10]	AIA-PSO (this work)	RGA-PSO (this work)
2	GP	100	100	100	100	100	100	100	100	100	100
3	ES	100	100	100	100	100	100	100	100	100	100
4	B2	100	100	100	100	100	100	100	100	100	100
5	DJ	100	100	100	100	100	100	100	100	100	100
7	RC	100	100	100	100	100	100	100	100	100	100
6	$RS_2$	100	100	100	100	100	100	100	100	100	100
10	$RS_5$	100	100	100	100	100	100			100	100
11	SH	100	100	100	100	100	100	100	100	100	100
12	$ZA_2$	100	100	100	100	100	100	100	100	100	100
13	$ZA_5$	100	100	100	100	100	100	I		100	100
14	$ZA_{10}$	100	100	100	100	100	100	100		100	100
("—" denote	s unavailable in	formation).									

TABLE 9: Comparison of the results of the proposed RGA-PSO and AIA-PSO approaches and those of the hybrid algorithms for 11 TPs.

Mathematical Problems in Engineering

# References

- W. Y. Yang, W. Cao, T.-S. Chung, and J. Morris, *Applied Numerical Methods Using MATLAB*, John Wiley & Sons, Hoboken, NJ, USA, 2005.
- [2] C. Hamzaçebi, "Improving genetic algorithms' performance by local search for continuous function optimization," *Applied Mathematics and Computation*, vol. 196, no. 1, pp. 309–317, 2008.
- [3] C. C. Chen, "Two-layer particle swarm optimization for unconstrained optimization problems," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 295–304, 2011.
- [4] X. Zhao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 119–124, 2010.
- [5] M. D. Toksari, "Minimizing the multimodal functions with Ant Colony Optimization approach," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6030–6035, 2009.
- [6] J. Kelsey and J. Timmis, "Immune inspired somatic contiguous hypermutation for function optimisation," in *Proceedings of the Genetic and Evolutionary Computation (GECCO '03)*, pp. 207– 208, Chicago, Ill, USA, 2003.
- [7] M. Gang, Z. Wei, and C. Xiaolin, "A novel particle swarm optimization algorithm based on particle migration," *Applied Mathematics and Computation*, vol. 218, no. 11, pp. 6620–6626, 2012.
- [8] H. Poorzahedy and O. M. Rouhani, "Hybrid meta-heuristic algorithms for solving network design problem," *European Journal of Operational Research*, vol. 182, no. 2, pp. 578–596, 2007.
- [9] Y. T. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 849–857, 2008.
- [10] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 129–142, 2007.
- [11] M. R. Chen, X. Li, X. Zhang, and Y. Z. Lu, "A novel particle swarm optimizer hybridized with extremal optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 367–373, 2010.
- [12] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, "Particle swarm optimization: hybridization perspectives and experimental illustrations," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5208–5226, 2011.
- [13] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems*, *Man, and Cybernetics, Part B*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [14] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Selforganizing hierarchical particle swarm optimizer with timevarying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [15] M. Jiang, Y. P. Luo, and S. Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, no. 1, pp. 8–16, 2007.
- [16] J. Y. Wu, "Solving constrained global optimization problems by using hybrid evolutionary computing and artificial life approaches," *Mathematical Problems in Engineering*, vol. 2012, Article ID 841410, 36 pages, 2012.
- [17] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, New York, NY, USA, 1999.

- [18] C. R. Houck, J. A. Joines, M. G. Kay, and in:, "A genetic algorithm for function optimization: a MATLAB implementation," in *NSCU-IE TR 95-09*, North Carolina State University, Raleigh, NC, USA, 1995.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, December 1995.
- [20] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Anchorage, Alaska, USA, May 1998.
- [21] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1951–1957, Washington, DC, USA, 1999.
- [22] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [23] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, 2005.
- [24] J. Y. Wu, "Solving constrained global optimization via artificial immune system," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 1, pp. 1–27, 2011.
- [25] L. N. de Castro and F. J. Von Zuben, "Artificial Immune Systems—Part I—Basic Theory and Applications," FEEC/Universidade Estadual de Campinas, Campinas, Brazil, 1999, ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/tr\_dca /trdca0199.pdf.
- [26] S. K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.



The Scientific World Journal





**Decision Sciences** 







Journal of Probability and Statistics



Hindawi Submit your manuscripts at http://www.hindawi.com



(0,1),

International Journal of Differential Equations





International Journal of Combinatorics





Mathematical Problems in Engineering



Abstract and Applied Analysis



Discrete Dynamics in Nature and Society







Function Spaces



International Journal of Stochastic Analysis



Journal of Optimization