

Research Article

Solving Constrained Global Optimization Problems by Using Hybrid Evolutionary Computing and Artificial Life Approaches

Jui-Yu Wu

Department of Business Administration, Lunghwa University of Science and Technology, No. 300, Section 1, Wanshou Road, Guishan, Taoyuan County 333, Taiwan

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

Received 28 February 2012; Revised 15 April 2012; Accepted 19 April 2012

Academic Editor: Jung-Fa Tsai

Copyright © 2012 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.

This work presents a hybrid real-coded genetic algorithm with a particle swarm optimization (RGA-PSO) algorithm and a hybrid artificial immune algorithm with a PSO (AIA-PSO) algorithm for solving 13 constrained global optimization (CGO) problems, including six nonlinear programming and seven generalized polynomial programming optimization problems. External RGA and AIA approaches are used to optimize the constriction coefficient, cognitive parameter, social parameter, penalty parameter, and mutation probability of an internal PSO algorithm. CGO problems are then solved using the internal PSO algorithm. The performances of the proposed RGA-PSO and AIA-PSO algorithms are evaluated using 13 CGO problems. Moreover, numerical results obtained using the proposed RGA-PSO and AIA-PSO algorithms are compared with those obtained using published individual GA and AIA approaches. Experimental results indicate that the proposed RGA-PSO and AIA-PSO algorithms converge to a global optimum solution to a CGO problem. Furthermore, the optimum parameter settings of the internal PSO algorithm can be obtained using the external RGA and AIA approaches. Also, the proposed RGA-PSO and AIA-PSO algorithms outperform some published individual GA and AIA approaches. Therefore, the proposed RGA-PSO and AIA-PSO algorithms are highly promising stochastic global optimization methods for solving CGO problems.

1. Introduction

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

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}), \\ &\text{s.t. } g_m(\mathbf{x}) \leq 0, \quad m = 1, 2, \dots, M, \end{aligned}$$

$$\begin{aligned}
 h_k(\mathbf{x}) &= 0, \quad k = 1, 2, \dots, K, \\
 x_n^l &\leq x_n \leq x_n^u, \quad n = 1, 2, \dots, N,
 \end{aligned}
 \tag{1.1}$$

where $f(\mathbf{x})$ denotes an objective function; $g_m(\mathbf{x})$ represents a set of m nonlinear inequality constraints; $h_k(\mathbf{x})$ refers to a set of k nonlinear equality constraints; \mathbf{x} represents a vector of decision variables which take real values, and each decision variable x_n is constrained by its lower and upper boundaries $[x_n^l, x_n^u]$; N is the total number of decision variables x_n . For instance, generalized polynomial programming (GPP) belongs to the nonlinear programming (NLP) method. The formulation of GPP is a nonconvex objective function subject to nonconvex inequality constraints and possibly disjointed feasible region. The GPP approach has been successfully used to solve problems including alkylation process design, heat exchanger design, optimal reactor design [1], inventory decision problem (economic production quantity) [2], process synthesis and the design of separations, phase equilibrium, nonisothermal complex reactor networks, and molecular conformation [3].

Traditional local NLP optimization approaches based on a gradient algorithm are inefficient for solving CGO problems, while an objective function is nondifferentiable. Global optimization methods can be divided into deterministic or stochastic [4]. Often involving a sophisticated optimization process, deterministic global optimization methods typically make assumptions regarding the problem to be solved [5]. Stochastic global optimization methods that do not require gradient information and numerous assumptions have received considerable attention. For instance, Sun et al. [6] devised an improved vector particle swarm optimization (PSO) algorithm with a constraint-preserving method to solve CGO problems. Furthermore, Tsoulos [7] developed a real-coded genetic algorithm (RGA) with a penalty function approach for solving CGO problems. Additionally, Deep and Dipti [8] presented a self-organizing GA with a tournament selection method for solving CGO problems. Meanwhile, Wu and Chung [9] developed a RGA with a static penalty function approach for solving GPP optimization problems. Finally, Wu [10] introduced an artificial immune algorithm (AIA) with an adaptive penalty function method to solve CGO problems.

Zadeh [11] defined “soft computing” as the synergistic power of two or more fused computational intelligence (CI) schemes, which can be divided into several branches: granular computing (e.g., fuzzy sets, rough sets, and probabilistic reasoning), neurocomputing (e.g., supervised, unsupervised, and reinforcement neural learning algorithms), evolutionary computing (e.g., GAs, genetic programming, and PSO algorithms), and artificial life (e.g., artificial immune systems) [12]. Besides, outperforming individual algorithms in terms of solving certain problems, hybrid algorithms can solve general problems more efficiently [13]. Therefore, hybrid CI approaches have recently attracted considerable attention as a promising field of research. Various hybrid evolutionary computing (GA and PSO methods) and artificial life (such as AIA methods) approaches have been developed for solving optimization problems. These hybrid algorithms focus on developing diverse candidate solutions (such as chromosomes and particles) of population/swarm to solve optimization problems more efficiently. These hybrid algorithms use two different algorithms to create diverse candidate solutions using their specific operations and then merge these diverse

candidate solutions to increase the diversity of the candidate population. For instance, Abd-El-Wahed et al. [14] developed an integrated PSO algorithm and GA to solve nonlinear optimization problems. Additionally, Kuo and Han [15] presented a hybrid GA and PSO algorithm for bilevel linear programming to solve a supply chain distribution problem. Furthermore, Shelokar et al. [16] presented a hybrid PSO method and ant colony optimization method for solving continuous optimization problems. Finally, Hu et al. [17] developed an immune cooperative PSO algorithm for solving the fault-tolerant routing problem.

Compared to the above hybrid CI algorithms, this work optimizes the parameter settings of an individual CI method by using another individual CI algorithm. A standard PSO algorithm has certain limitations [17, 18]. For instance, a PSO algorithm includes many parameters that must be set, such as the cognitive parameter, social parameter, and constriction coefficient. In practice, the optimal parameter settings of a PSO algorithm are tuned based on trial and error and prior knowledge is required to successfully manipulate the cognitive parameter, social parameter, and constriction coefficient. The exploration and exploitative capabilities of a PSO algorithm are limited to optimum parameter settings. Moreover, conventional PSO methods involve premature convergence that rapidly losses diversity during optimization.

Fortunately, optimization of parameter settings for a conventional PSO algorithm can be considered an unconstrained global optimization (UGO) problem, and the diversity of candidate solutions of the PSO method can be increased using a multi-nonuniform mutation operation [19]. Moreover, the parameter manipulation of a GA and AIA method is easy to implement without prior knowledge. Therefore, to overcome the limitations of a standard PSO algorithm, this work develops two hybrid CI algorithms to solve CGO problems efficiently. The first algorithm is a hybrid RGA and PSO (RGA-PSO) algorithm, while the second algorithm is a hybrid AIA and PSO (AIA-PSO) algorithm. The proposed RGA-PSO and AIA-PSO algorithms are considered to optimize two optimization problems simultaneously. The UGO problem (optimization of cognitive parameter, social parameter, constriction coefficient, penalty parameter, and mutation probability of an internal PSO algorithm based on a penalty function approach) is optimized using external RGA and AIA approaches, respectively. A CGO 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 CGO problems (e.g., six benchmark NLP and seven GPP optimization problems).

The rest of this paper is organized as follows. Section 2 describes the RGA, PSO algorithm, AIA, and penalty function approaches. Section 3 then introduces the proposed RGA-PSO and AIA-PSO algorithms. Next, Section 4 compares the experimental results of the proposed RGA-PSO and AIA-PSO algorithms with those of various published individual GAs and AIAs [9, 10, 20–22] and hybrid algorithms [23, 24]. Finally, conclusions are drawn in Section 5.

2. Related Works

2.1. Real-Coded Genetic Algorithm

GAs are stochastic global optimization methods based on the concepts of natural selection and use three genetic operators, that is, selection, crossover, and mutation, to explore and exploit the solution space. RGA outperforms binary-coded GA in solving continuous function optimization problems [19]. This work thus describes operators of a RGA [25].

2.1.1. Selection

A selection operation selects strong individuals from a current population based on their fitness function values and then reproduces these individuals into a crossover pool. The several selection operations developed include the roulette wheel, ranking, and tournament methods [19, 25]. This work uses the normalized geometric ranking method, as follows:

$$p_j = q'(1-q)^{r-1}, \quad j = 1, 2, \dots, p_{\text{SRGA}}, \quad (2.1)$$

p_j = probability of selecting individual j , q = probability of choosing the best individual (here $q = 0.35$)

$$q' = \frac{q}{1 - (1-q)^{p_{\text{SRGA}}}}, \quad (2.2)$$

r = individual ranking based on fitness value, where 1 represents the best, $r = 1, 2, \dots, p_{\text{SRGA}}$, p_{SRGA} = population size of the RGA.

2.1.2. Crossover

While exploring the solution space by creating new offspring, the crossover operation randomly selects two parents from the crossover pool and then uses these two parents to generate two new offspring. This operation is repeated until the $p_{\text{SRGA}}/2$ is satisfied. The whole arithmetic crossover is easily implemented, as follows:

$$\begin{aligned} \mathbf{v}'_1 &= \beta \times \mathbf{v}_1 + (1 - \beta) \times \mathbf{v}_2, \\ \mathbf{v}'_2 &= (1 - \beta) \mathbf{v}_1 + \beta \times \mathbf{v}_2, \end{aligned} \quad (2.3)$$

where \mathbf{v}_1 and \mathbf{v}_2 = parents, \mathbf{v}'_1 and \mathbf{v}'_2 = offspring, β = uniform random number in the interval $[0, 1.5]$.

2.1.3. Mutation

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

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x_n^u - x_{\text{current},n})\text{pert}(g_{\text{RGA}}) & \text{if } U_1(0,1) < 0.5, \\ x_{\text{current},n} - (x_{\text{current},n} - x_n^l)\text{pert}(g_{\text{RGA}}) & \text{if } U_1(0,1) \geq 0.5, \end{cases} \quad (2.4)$$

where $\text{pert}(g_{\text{RGA}}) = [U_2(0,1)(1 - g_{\text{RGA}}/g_{\text{max,RGA}})]^2$, 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, g_{RGA} = current generation of the RGA, $x_{\text{current},n}$ = current decision variable x_n , $x_{\text{trial},n}$ = trial candidate solution x_n .

2.2. Particle Swarm Optimization

Kennedy and Eberhart [26] first introduced a conventional PSO algorithm, which is inspired by the social behavior of bird flocks or fish schools. Like GAs, a PSO algorithm is a population-based algorithm. A population of candidate solutions is called a particle swarm. The particle velocities can be updated by (2.5), as follows:

$$\begin{aligned} v_{j,n}(g_{\text{PSO}} + 1) = & v_{j,n}(g_{\text{PSO}}) + c_1 r_{1j}(g_{\text{PSO}}) \left[p_{j,n}^{lb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \\ & + c_2 r_{2j}(g_{\text{PSO}}) \left[p_{j,n}^{gb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \quad j = 1, 2, \dots, \text{ps}_{\text{PSO}}, \quad n = 1, 2, \dots, N, \end{aligned} \quad (2.5)$$

$v_{j,n}(g_{\text{PSO}} + 1)$ = particle velocity of decision variable x_n of particle j at generation $g_{\text{PSO}} + 1$, $v_{j,n}(g_{\text{PSO}})$ = particle velocity of decision variable x_n of particle j at generation g_{PSO} , c_1 = cognitive parameter, c_2 = social parameter, $x_{j,n}(g_{\text{PSO}})$ = particle position of decision variable x_n of particle j at generation g_{PSO} , $r_{1j}(g_{\text{PSO}})$, $r_{2j}(g_{\text{PSO}})$ = independent uniform random numbers in the interval $[0, 1]$ at generation g_{PSO} , $p_{j,n}^{lb}(g_{\text{PSO}})$ = best local solution at generation g_{PSO} , $p_{j,n}^{gb}(g_{\text{PSO}})$ = best global solution at generation g_{PSO} , ps_{PSO} = population size of the PSO algorithm.

The particle positions can be computed using (2.6), as follows:

$$x_{j,n}(g_{\text{PSO}} + 1) = x_{j,n}(g_{\text{PSO}}) + v_{j,n}(g_{\text{PSO}} + 1) \quad j = 1, 2, \dots, \text{ps}_{\text{PSO}}, \quad n = 1, 2, \dots, N. \quad (2.6)$$

Shi and Eberhart [27] developed a modified PSO algorithm by incorporating an inertia weight (ω_{in}) into (2.7) to control the exploration and exploitation capabilities of a PSO algorithm, as follows:

$$\begin{aligned} v_{j,n}(g_{\text{PSO}} + 1) = & \omega_{\text{in}} v_{j,n}(g_{\text{PSO}}) + c_1 r_{1j}(g_{\text{PSO}}) \left[p_{j,n}^{lb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \\ & + c_2 r_{2j}(g_{\text{PSO}}) \left[p_{j,n}^{gb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \quad j = 1, 2, \dots, \text{ps}_{\text{PSO}}, \quad n = 1, 2, \dots, N. \end{aligned} \quad (2.7)$$

A constriction coefficient (χ) was inserted into (2.8) to balance the exploration and exploitation tradeoff [28–30], as follows:

$$\begin{aligned} v_{j,n}(g_{\text{PSO}} + 1) = & \chi \left\{ v_{j,n}(g_{\text{PSO}}) + \rho_1(g_{\text{PSO}}) \left[p_{j,n}^{lb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \right. \\ & \left. + \rho_2(g_{\text{PSO}}) \left[p_{j,n}^{gb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \right\} \quad j = 1, 2, \dots, \text{ps}_{\text{PSO}}, \quad n = 1, 2, \dots, N, \end{aligned} \quad (2.8)$$

where

$$\chi = \frac{2U_3(0,1)}{\left|2 - \tau - \sqrt{\tau(\tau - 4)}\right|}, \quad (2.9)$$

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

This work considers parameters ω_{in} and χ to update the particle velocities, as follows:

$$\begin{aligned} v_{j,n}(g_{\text{PSO}} + 1) = & \chi \left\{ \omega_{in} v_{j,n}(g_{\text{PSO}}) + c_1 r_{1j}(g_{\text{PSO}}) \left[p_{j,n}^{lb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \right. \\ & \left. + c_2 r_{2j}(g_{\text{PSO}}) \left[p_{j,n}^{gb}(g_{\text{PSO}}) - x_{j,n}(g_{\text{PSO}}) \right] \right\} \quad j = 1, 2, \dots, p_{\text{PSO}}, \quad n = 1, 2, \dots, N, \end{aligned} \quad (2.10)$$

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

According to (2.10), the optimal values of parameters c_1 , c_2 , and χ are difficult to obtain through a trial and error. This work thus optimizes these parameter settings by using RGA and AIA approaches.

2.3. Artificial Immune Algorithm

Wu [10] presented an AIA based on clonal selection and immune network theories to solve CGO problems. The AIA approach comprises 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 immune system, an antigen (**Ag**) has multiple epitopes (antigenic determinants), which can be recognized by various **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 , $n = 1, 2, \dots, N$) 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, which is based on the immune network principle [31], controls the number of antigen-specific **Ab**s. This operation is defined according to **Ab-Ag** and **Ab-Ab** recognition information, as follows:

$$p_{r_j} = \frac{1}{N} \sum_{n=1}^N \frac{1}{e^{d_{nj}}}, \quad (2.11)$$

$$d_{nj} = \left| \frac{x_n^* - x_{nj}}{x_n^*} \right|, \quad j = 1, 2, \dots, r_s, \quad n = 1, 2, \dots, N,$$

where p_{r_j} = probability that **Ab** j recognizes **Ab**^{*} (the best solution), x_n^* = the best **Ab**^{*} with the highest **Ab-Ag** affinity, x_{nj} = decision variables x_n of **Ab** j , r_s = repertoire (population) size of the AIA.

The **Ab**^{*} is recognized by other **Ab** j in a current **Ab** repertoire. Large p_{r_j} implies that **Ab** j can effectively recognize **Ab**^{*}. The **Ab** j with p_{r_j} that is equivalent to or larger than the threshold degree p_{rt} is reproduced to generate an intermediate **Ab** repertoire.

2.3.3. Hypermutation Operation

Multi-nonuniform mutation [19] is used as the somatic hypermutation operation, which can be expressed as follows:

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x_n^u - x_{\text{current},n})\text{pert}(g_{\text{AIA}}), & \text{if } U_4(0,1) < 0.5, \\ x_{\text{current},n} - (x_{\text{current},n} - x_n^l)\text{pert}(g_{\text{AIA}}), & \text{if } U_4(0,1) \geq 0.5, \end{cases} \quad (2.12)$$

where $\text{pert}(g_{\text{AIA}}) = \{U_5(0,1)(1 - g_{\text{AIA}}/g_{\text{max,AIA}})\}^2$ = perturbation factor, g_{AIA} = current generation of the AIA, $g_{\text{max,AIA}}$ = maximum generation number of the AIA, $U_4(0,1)$ 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 using the standard Cauchy distribution $C(0,1)$, in which the local parameter is zero and the scale parameter is one. Receptor editing is performed using Cauchy random variables that are generated 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

$$x_{\text{trial}} = x_{\text{current}} + U_5(0,1)^2 \times \sigma, \quad (2.13)$$

where $\sigma = [\sigma_1, \sigma_2, \dots, \sigma_N]^T$, vector of Cauchy random variables, $U_5(0,1)$ = uniform random number in the interval $[0, 1]$.

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

2.3.5. Bone Marrow Operation

The paratope of an **Ab** can be generated by recombining gene segments V_H D_H J_H and V_L J_L [32]. Therefore, based on this metaphor, diverse **Ab**s are synthesized using a bone marrow operation. This operation randomly chooses two **Ab**s from the intermediate **Ab** repertoire and a recombination point from the gene segments of the paratope of the selected **Ab**s. The selected gene segments (e.g., gene x_1 of **Ab** 1 and gene x_1 of the **Ab** 2) are reproduced to create a library of gene segments. The selected gene segments in the paratope are then deleted. The new **Ab** 1 is formed by inserting the gene segment, which is gene x_1 of the **Ab** 2 in the library plus a random variable created from standard normal distribution $N(0, 1)$, at the recombination point. The literature details the implementation of the bone marrow [10].

2.4. Penalty Function Methods

Stochastic global optimization approaches, including GAs, AIAs, and PSO, are naturally unconstrained optimization methods. Penalty function methods, which are constraint handling approaches, are commonly used to create feasible solutions to a CGO problem and transform it into an unconstrained optimization problem. Two popular penalty functions exist, namely, the exterior and interior functions. Exterior penalty functions use an infeasible solution as a starting point, and convergence is from the infeasible region to the feasible one. Interior penalty functions start from a feasible solution, then 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. The exterior penalty functions developed to date include static, dynamic, adaptive, and death penalty functions [33]. This work uses the form of a static penalty function, as follows:

$$\text{Minimize } f_{\text{pseudo}}(\mathbf{x}, \rho) = f(\mathbf{x}) + \rho \left\{ \sum_{m=1}^M \left\{ \max[0, g_m(\mathbf{x})] + \sum_{k=1}^K [h_k(\mathbf{x})] \right\}^2 \right\}, \quad (2.14)$$

where $f_{\text{pseudo}}(\mathbf{x}, \rho)$ = pseudo-objective function obtained using an original objective function plus a penalty term, ρ = penalty parameter.

Unfortunately, the penalty function scheme is limited by the need to fine-tune the penalty parameter ρ [8]. To overcome this limitation, this work attempts to find the optimum ρ for each CGO problem using the RGA and AIA approaches. Additionally, to obtain high-quality RGA-PSO and AIA-PSO solutions accurate to at least five decimal places for the violation of each constraint to a specific CGO problem, the parameter ρ is within the search space $[1 \times 10^9, 1 \times 10^{11}]$.

3. Method

3.1. RGA-PSO Algorithm

Figure 1 shows the pseudocode of the proposed RGA-PSO algorithm. The external RGA approach is used to optimize the best parameter settings of the internal PSO algorithm, and the internal PSO algorithm is employed to solve CGO problems.

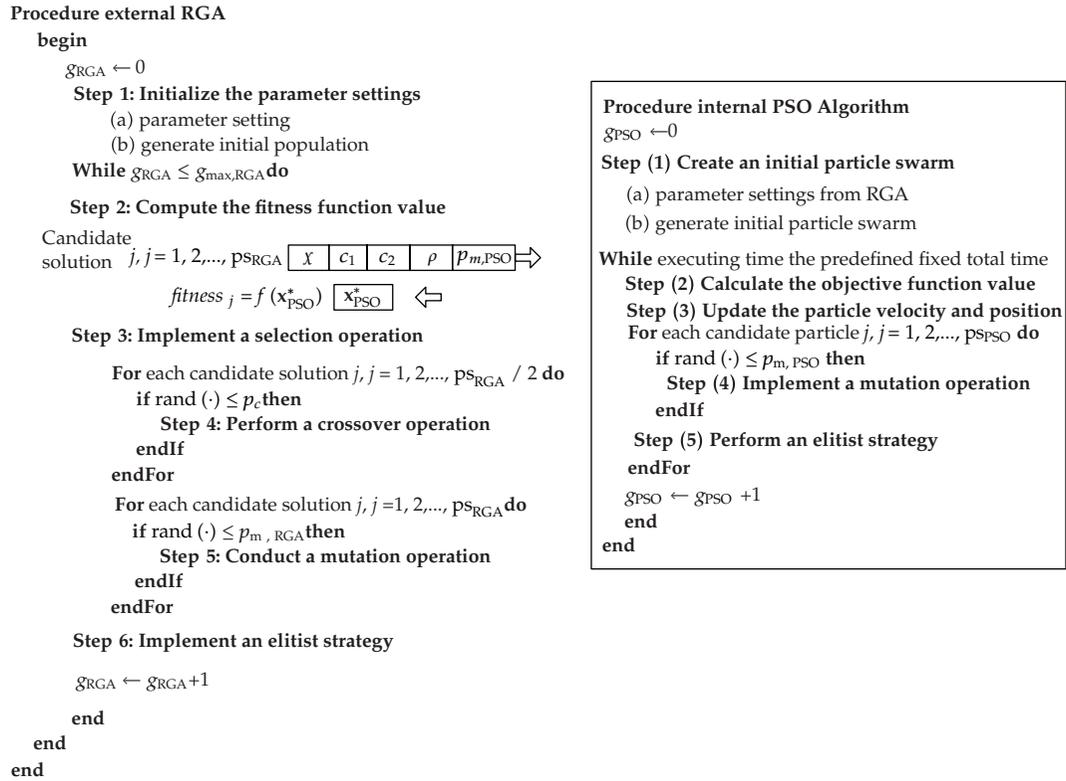


Figure 1: The pseudocode of the proposed RGA-PSO algorithm.

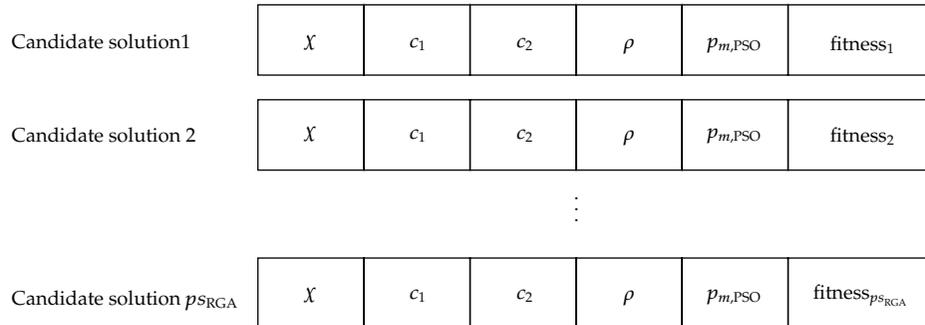


Figure 2: Chromosome representation of the external RGA.

External RGA

Step 1 (initialize the parameter settings). Parameter settings are given such as p_{SRGA} , crossover probability p_c , mutation probability of the external RGA approach $p_{m,RGA}$, the lower and upper boundaries of these parameters c_1, c_2, χ, ρ , and the mutation probability of the internal PSO algorithm $p_{m,PSO}$. The candidate solutions (individuals) of the external RGA represent the optimized parameters of the internal PSO algorithm. Finally, Figure 2 illustrates the candidate solution of the external RGA approach.

Candidate solution 1	x_1	x_2	\dots	x_N	$f(x_{\text{PSO},1})$
Candidate solution 2	x_1	x_2	\dots	x_N	$f(x_{\text{PSO},2})$
			\vdots		
Candidate solution p_{SPSO}	x_1	x_2	\dots	x_N	$f(x_{\text{PSO},p_{\text{SPSO}}})$

Figure 3: Candidate solution of the internal PSO algorithm.

Step 2 (compute the fitness function value). The fitness function value fitness_j of the external RGA approach is the best objective function value $f(x_{\text{PSO}}^*)$ obtained from the best solution x_{PSO}^* of each internal PSO algorithm execution, as follows:

$$\text{fitness}_j = f(x_{\text{PSO}}^*), \quad j = 1, 2, \dots, p_{\text{SRGA}}. \quad (3.1)$$

Candidate solution j of the external RGA approach is incorporated into the internal PSO algorithm, and a CGO problem is then solved using the internal PSO algorithm, which is executed as follows.

Internal PSO Algorithm

Step (1) (create an initial particle swarm). An initial particle swarm is created based on the p_{SPSO} from $[x_n^l, x_n^u]$ of a CGO problem. A particle represents a candidate solution of a CGO problem, as shown in Figure 3.

Step (2) (calculate the objective function value). According to (2.14), the pseudo-objective function value of the internal PSO algorithm is defined by

$$f_{\text{pseudo},j} = f(x_{\text{PSO},j}) + \left\{ \rho \times \sum_{m=1}^M \{ \max[0, g_m(x_{\text{PSO},j})] \}^2 \right\}, \quad j = 1, 2, \dots, p_{\text{SPSO}}. \quad (3.2)$$

Step (3) (update the particle velocity and position). The particle position and velocity can be updated using (2.6) and (2.10), respectively.

Step (4) (implement a mutation operation). The standard PSO algorithm lacks evolution operations of GAs such as crossover and mutation. To maintain the diversity of particles, this work uses the multi-nonuniform mutation operator defined by (2.4).

Step (5) (perform an elitist strategy). A new particle swarm is created from internal step (3). Notably, $f(x_{\text{PSO},j})$ of a candidate solution j (particle j) in the particle swarm

is evaluated. Here, a pairwise comparison is made between the $f(\mathbf{x}_{\text{PSO},j})$ value of candidate solutions in the new and current particle swarms. A situation in which the candidate solution j ($j = 1, 2, \dots, p_{\text{PSO}}$) in the new particle swarm is superior to 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 (2) to (5) are repeated until the $g_{\text{max,PSO}}$ value of the internal PSO algorithm is satisfied.

End

Step 3 (implement selection operation). The parents in a crossover pool are selected using (2.1).

Step 4 (perform crossover operation). In GAs, the crossover operation performs a global search. Thus, the crossover probability p_c usually exceeds 0.5. Additionally, candidate solutions are created using (2.3).

Step 5 (conduct mutation operation). In GAs, the mutation operation implements a local search. Additionally, a solution space is exploited using (2.4).

Step 6 (implement an elitist strategy). This work updates the population using an elitist strategy. A situation in which the fitness $_j$ of candidate solution j in the new population is larger than that in the current population suggests that the weak candidate solution j is replaced. Additionally, a situation in which the fitness $_j$ of candidate solution j in the new population is equal to or worse than that in the current population implies that the candidate solution j in the current population survives. In addition to maintaining the strong candidate solutions, this strategy eliminates weak candidate solutions.

External Steps 2 to 6 are repeated until the $g_{\text{max,RGA}}$ value of the external RGA approach is met.

3.2. AIA-PSO Algorithm

Figure 4 shows the pseudocode of the proposed AIA-PSO algorithm, in which the external AIA approach is used to optimize the parameter settings of the internal PSO algorithm and the PSO algorithm is used to solve CGO problems.

External AIA

Step 1 (initialize the parameter settings). Several parameters must be predetermined. These include r_s and the threshold for **Ab-Ab** recognition p_{rt} , as well as the lower and upper boundaries of these parameters c_1 , c_2 , χ , ρ , and $p_{m,\text{PSO}}$. Figure 5 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 , χ , ρ , and $p_{m,PSO}$ for the internal PSO algorithm, subsequently leading to the implementation of internal steps (1)–(5) of the PSO algorithm. The PSO algorithm returns the best fitness value of PSO $f(\mathbf{x}_{PSO}^*)$ to the external AIA approach.

- Step (1) (create an initial particle swarm). An initial particle swarm is created based on p_{PSO} from $[x_n^l, x_n^u]$ of a CGO problem. A particle represents a candidate solution of a CGO problem.
- Step (2) (calculate the objective function value). Equation (3.2) is used as the pseudo-objective function value of the internal PSO algorithm.
- Step (3) (update the particle velocity and position). Equations (2.6) and (2.10) can be used to update the particle position and velocity.
- Step (4) (implement a mutation operation). The diversity of the particle swarm is increased using (2.4).
- Step (5) (perform an elitist strategy). A new particle swarm (population) is generated from internal step (3). Notably, $f(\mathbf{x}_{PSO,j})$ of a candidate solution j (particle j) in the particle swarm is evaluated. Here, a pairwise comparison is made between the $f(\mathbf{x}_{PSO,j})$ value of candidate solutions in the new and current particle swarms. 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 (2) to (5) are repeated until the $g_{\max,PSO}$ value of the internal PSO algorithm is satisfied.

End

Consistent with the **Ab-Ag** affinity metaphor, an **Ab-Ag** affinity is determined using (3.3), as follows:

$$\max(\text{affinity}_j) = -1 \times f(\mathbf{x}_{PSO}^*) \quad j = 1, 2, \dots, rs. \quad (3.3)$$

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

Step 3 (perform clonal selection operation). To control the number of antigen-specific **Ab**s, (2.11) is used.

Step 4 (implement **Ab-Ag** affinity maturation). The intermediate **Ab** repertoire that is created in external Step 3 is divided into two subsets. These **Ab**s undergo somatic hypermutation operation by using (2.12) when the random number is 0.5 or less. Notably, these **Ab**s suffer receptor editing operation using (2.13) when the random number exceeds 0.5.

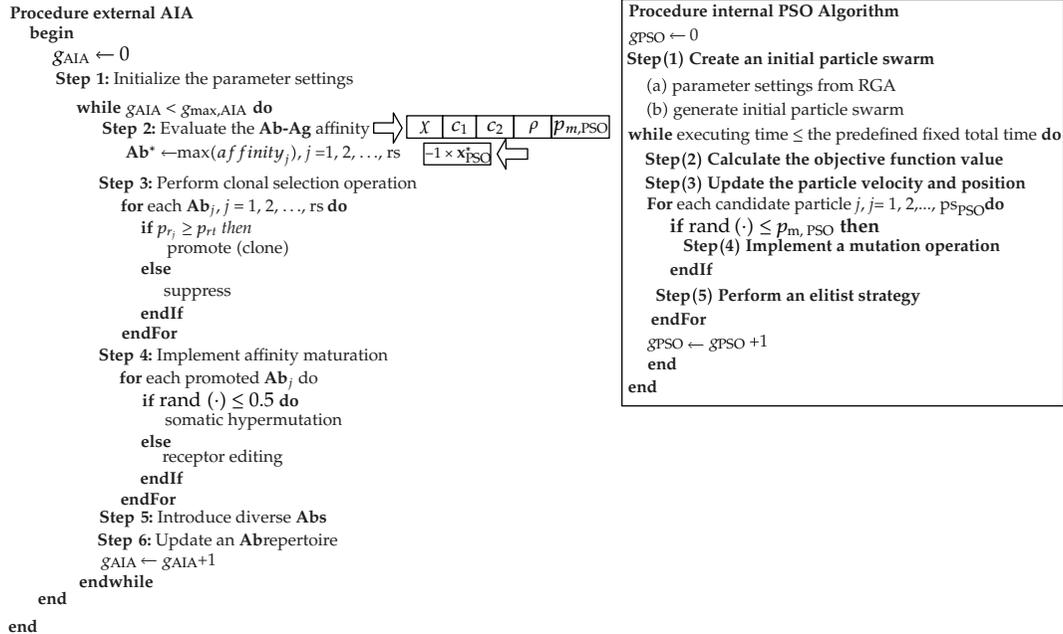


Figure 4: The pseudocode of the AIA-PSO algorithm.

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**.

External Steps 2–6 are repeated until the termination criterion $g_{max,AIA}$ is satisfied.

4. Results

The 13 CGO problems were taken from other studies [1, 20, 21, 23, 34]. The set of CGO problems comprises six benchmark NLP problems (TPs 1–4 and 12–13), and seven GPP problems, in which TP 5 (alkylation process design in chemical engineering), TP 6 (optimal reactor design), TP 12 (a tension/compression string design problem), and TP 13 (a pressure vessel design problem) are constrained engineering problems, were used to evaluate the performances of the proposed RGA-PSO and AIA-PSO algorithms. In the appendix, the objective function, constraints, boundary conditions of decision variables, and known global

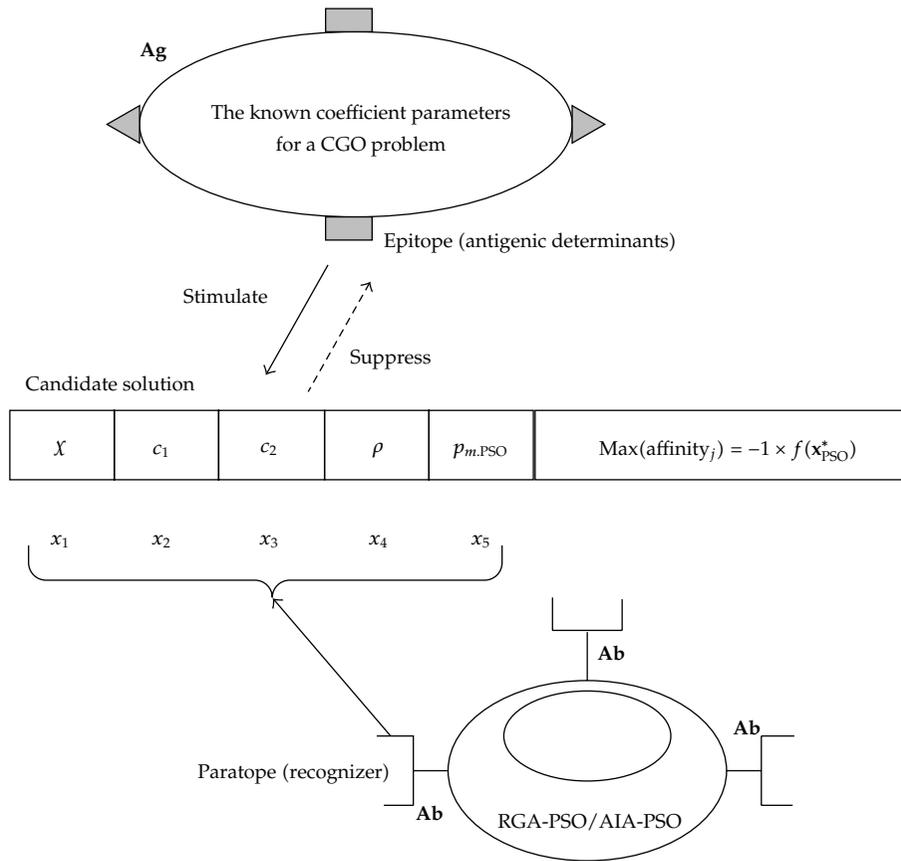


Figure 5: Ag and Ab representation.

optimum for TPs 1–11 are described and the problem characteristics of TPs 5, 12, and 13 are further detailed.

The proposed RGA-PSO and AIA-PSO algorithms were coded in MATLAB software and executed on a Pentium D 3.0 (GHz) personal computer. Fifty independent runs were conducted to solve each test problem (TP). Numerical results were summarized, including the best, median, mean, and worst results, as well as the standard deviation (S.D.) of objective function values obtained using RGA-PSO and AIA-PSO solutions, mean computational CPU times (MCCTs), and mean absolute percentage error MAPE, as defined by

$$\text{MAPE} = \frac{\sum_{s=1}^{50} |(f(\mathbf{x}^*) - f(\mathbf{x}_{\text{stochastic}}^s)) / f(\mathbf{x}^*)|}{50} \times 100\%, \quad s = 1, 2, \dots, 50, \quad (4.1)$$

where $f(\mathbf{x}^*)$ = value of the known global solution, $f(\mathbf{x}_{\text{stochastic}}^s)$ = values obtained from solutions of stochastic global optimization approaches (e.g., RGA-PSO and AIA-PSO algorithms).

Table 1 lists the parameter settings for the RGA-PSO and AIA-PSO algorithms, as shown in Table 1.

Table 1: The parameter settings for the RGA-PSO and AIA-PSO algorithms.

Methods	Parameter settings	Search space
	$p_c = 1$	
The external RGA	$p_{m,RGA} = 0.15$ $ps_{RGA} = 10$ $g_{max,RGA} = 3$ $p_{rt} = 0.9$	$[x^l, x^u] = [0.1, 1]$ $[c_1^l, c_1^u] = [0.1, 2]$ $[c_2^l, c_2^u] = [0.1, 2]$ $[\rho^l, \rho^u] = [1 \times 10^9, 1 \times 10^{11}]$
The external AIA	$rs = 10$ $g_{max,AIA} = 3$ $ps_{PSO} = 100$	$[p_{m,PSO}^l, p_{m,PSO}^u] = [0.1, 0.5]$
The internal PSO algorithm	$g_{max,PSO} = 3500$ for TPs 1–4 $g_{max,PSO} = 3000$ for TPs 5–13	$[x_n^l, x_n^u]$ for a CGO problem

$[x^l, x^u]$: the lower and upper boundaries of parameter x .
 $[c_1^l, c_1^u]$: the lower and upper boundaries of parameter c_1 .
 $[c_2^l, c_2^u]$: the lower and upper boundaries of parameter c_2 .
 $[\rho^l, \rho^u]$: the lower and upper boundaries of parameter ρ .
 $[p_{m,PSO}^l, p_{m,PSO}^u]$: the lower and upper boundaries of $p_{m,PSO}$ for the internal PSO algorithm $p_{m,PSO}$.

4.1. Comparison of the Results Obtained Using the RGA-PSO and AIA-PSO Algorithms

Table 2 summarizes the numerical results obtained using the proposed RGA-PSO and AIA-PSO algorithms for TPs 1–13. Numerical results indicate that the RGA-PSO and the AIA-PSO algorithms can obtain the global minimum solution to TPs 1–11, since each MAPE% is small. Moreover, the best, median, worst, and S.D. of objective function values obtained using the RGA-PSO and AIA-PSO solutions are identical for TPs 1, 2, 3, 4, 6, 7, 8, 9, and 11. Furthermore, the worst values obtained using the AIA-PSO algorithm for TPs 5 and 13 are smaller than those obtained using the RGA-PSO algorithm. Additionally, t -test is performed for each TP, indicating that the mean values obtained using the RGA-PSO and AIA-PSO algorithms are statistically significant for TPs 5, 10, 12, and 13, since P value is smaller than a significant level 0.05. Based on the results of t -test, the AIA-PSO algorithm yields better mean values than the RGA-PSO algorithm for TPs 5, 12, and 13, and the AIA-PSO algorithm yields worse mean value than the RGA-PSO algorithm for TP 10.

Tables 3 and 4 list the best solutions obtained using the RGA-PSO and AIA-PSO algorithms from TPs 1–13, respectively, indicating that each constraint is satisfied (i.e., the violation of each constraint is accurate to at least five decimal places) for every TP. Tables 5 and 6 list the best parameter settings of the internal PSO algorithm obtained using the external RGA and AIA approaches, respectively.

4.2. Comparison of the Results for the Proposed RGA-PSO and AIA-PSO Algorithms with Those Obtained Using the Published Individual GA and AIA Approaches and Hybrid Algorithms

Table 7 compares the numerical results of the proposed RGA-PSO and AIA-PSO algorithms with those obtained using published individual GA and AIA approaches for TPs 1–4. In this table, GA-1 is a GA with a penalty function methods, as used by Michalewicz [20]. Notably,

Table 2: Numerical results of the proposed RGA-PSO and AIA-PSO algorithms for TPs 1–13.

TP number	Global optimum	Methods	Best	Mean	Median	Worst	MAPE %	S.D.	MCCT (sec)	P value
1	24.306	The proposed RGA-PSO	24.323	24.568	24.521	24.831	1.078	0.13	432.39	0.643
		The proposed AIA-PSO	24.358	24.579	24.564	24.809	1.125	0.11	417.63	
2	-30665.539	The proposed RGA-PSO	-30665.539	-30665.539	-30665.539	-30665.534	1.34E-06	7.36E-04	349.55	0.478
		The proposed AIA-PSO	-30665.539	-30665.539	-30665.539	-30665.539	9.95E-07	3.81E-05	349.76	
3	680.630	The proposed RGA-PSO	680.632	680.640	680.639	680.658	1.46E-03	5.55E-03	398.72	0.839
		The proposed AIA-PSO	680.633	680.640	680.640	680.657	1.50E-03	0.01	398.84	
4	-15	The proposed RGA-PSO	-15	-15	-15	-15	1.27E-06	1.27E-06	466.00	0.294
		The proposed AIA-PSO	-15	-15	-15	-15	1.20E-10	2.64E-11	450.28	
5	1227.1978	The proposed RGA-PSO	1227.2321	1228.8471	1228.3935	1231.6003	1.34E-01	1.27	481.86	0.002*
		The proposed AIA-PSO	1227.1598	1228.1800	1227.9762	1229.6590	8.02E-02	0.71	461.30	
6	3.9511	The proposed RGA-PSO	3.9521	3.9577	3.9566	3.9722	1.67E-01	5E-03	439.16	0.682
		The proposed AIA-PSO	3.9516	3.9581	3.9569	3.9735	1.77E-01	4.79E-03	438.55	
7	-5.7398	The proposed RGA-PSO	-5.7398	-5.7398	-5.7398	-5.7398	2.54E-04	1.13E-05	325.73	0.358

Table 2: Continued.

TP number	Global optimum	Methods	Best	Mean	Median	Worst	MAPE %	S.D.	MCCT (sec)	P value
8	-83.254	The proposed AIA-PSO	-5.7398	-5.7398	-5.7398	-5.7398	2.42E-04	7.74E-06	329.12	0.291
		The proposed RGA-PSO	-83.2497	-83.2497	-83.2497	-83.2497	5.13E-03	1.26E-06	211.91	
9	-6.0482	The proposed AIA-PSO	-83.2497	-83.2497	-83.2497	-83.2497	5.13E-03	6.85E-07	213.01	0.577
		The proposed RGA-PSO	-6.0441	-5.9883	-5.9968	-5.8051	0.991	0.05	440.72	
10	6300	The proposed AIA-PSO	-6.0467	-5.9828	-5.9979	-5.8274	1.082	0.05	441.42	0.001*
		The proposed RGA-PSO	6299.8374	6299.8412	6299.8419	6299.8428	2.52E-03	1.42E-03	224.13	
11	10122.6964	The proposed AIA-PSO	6299.8395	6299.8420	6299.8423	6299.8425	2.51E-03	6.83E-04	223.91	0.884
		The proposed RGA-PSO	10122.4732	10122.4925	10122.4925	10122.6444	2.01E-03	2.26E-02	340.89	
12	—	The proposed AIA-PSO	10122.4852	10122.4920	10122.4927	10122.4931	2.02E-03	1.85E-03	338.37	0.014*
		The proposed RGA-PSO	0.012692	0.012724	0.012721	0.012784	—	1.46E-05	293.67	
13	—	The proposed AIA-PSO	0.012667	0.012715	0.012719	0.012778	—	2.00E-05	296.15	0.019*
		The proposed RGA-PSO	5885.3018	5895.0381	5885.3326	6005.4351	—	24.33	291.17	
		The proposed AIA-PSO	5885.3249	5886.5426	5885.3323	5906.7404	—	4.54	292.16	

(The “—” denotes unavailable information, and *represents that the mean values obtained using the RGA-PSO and AIA-PSO algorithms are statistically different.)

Table 3: The best solutions obtained using the RGA-PSO algorithm from TPs 1–13.

TP number	$f(\mathbf{x}_{\text{RGA-PSO}}^*)$	$\mathbf{x}_{\text{RGA-PSO}}^*$
1	24.323	$\mathbf{x}_{\text{RGA-PSO}}^* = (2.16727760, 2.37789634, 8.78162804, 5.12372885, 0.97991270, 1.39940993, 1.31127142, 9.81945011, 8.30004549, 8.45891329)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (-0.000171 \leq 0, -0.003109 \leq 0, -0.000027 \leq 0, -0.000123 \leq 0, -0.001371 \leq 0, -0.002101 \leq 0, -6.245957 \leq 0, -47.846302 \leq 0)$
2	-30665.539	$\mathbf{x}_{\text{RGA-PSO}}^* = (78, 33, 29.99525450, 45, 36.77581373)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (-92.000000 \leq 0, 2.24E - 07 \leq 0, -8.840500 \leq 0, -11.159500 \leq 0, 4.28E - 07 \leq 0, -5.000000 \leq 0)$
3	680.632	$\mathbf{x}_{\text{RGA-PSO}}^* = (2.33860239, 1.95126191, -0.45483579, 4.36300325, -0.62317747, 1.02938443, 1.59588410)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (-7.76E - 07 \leq 0, -252.7211 \leq 0, -144.8140 \leq 0, -6.15E - 05 \leq 0)$
4	-15	$\mathbf{x}_{\text{RGA-PSO}}^* = (1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (0 \leq 0, 0 \leq 0, 0 \leq 0, -5 \leq 0, -5 \leq 0, -5 \leq 0, 0 \leq 0, 0 \leq 0, 0 \leq 0)$
5	1227.1139	$\mathbf{x}_{\text{RGA-PSO}}^* = (1697.13793410, 54.17332240, 3030.44600072, 90.18199040, 94.99999913, 10.42097385, 153.53771370)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (0.999978 \leq 1, 0.980114 \leq 1, 1.000005 \leq 1, 0.980097 \leq 1, 0.990565 \leq 1, 1.000005 \leq 1, 1.000000 \leq 1, 0.976701 \leq 1, 1.000003 \leq 1, 0.459311 \leq 1, 0.387432 \leq 1, 0.981997 \leq 1, 0.980364 \leq 1, -8.241926 \leq 1)$
6	3.9521	$\mathbf{x}_{\text{RGA-PSO}}^* = (6.444100620, 2.243029250, 0.642672939, 0.582321363, 5.940008650, 5.531235784, 1.018087316, 0.403665649)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (1.000000 \leq 1, 1.000000 \leq 1, 0.999996 \leq 1, 0.99986 \leq 1)$
7	-5.7398	$\mathbf{x}_{\text{RGA-PSO}}^* = (8.12997229, 0.61463971, 0.56407162, 5.63623069)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (1.000000 \leq 1, 1.000000 \leq 1)$
8	-83.2497	$\mathbf{x}_{\text{RGA-PSO}}^* = (88.35595404, 7.67259607, 1.31787691)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (1.000000 \leq 1)$
9	-6.0441	$\mathbf{x}_{\text{RGA-PSO}}^* = (6.40497368, 0.64284563, 1.02766984, 5.94729224, 2.21044814, 0.59816471, 0.42450835, 5.54339987)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (0.999997 \leq 1, 0.999959 \leq 1, 0.999930 \leq 1, 0.999975 \leq 1)$
10	6299.8374	$\mathbf{x}_{\text{RGA-PSO}}^* = (108.66882633, 85.10837983, 204.35894362)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (1.000002 \leq 1)$
11	10122.4732	$\mathbf{x}_{\text{RGA-PSO}}^* = (78, 33, 29.99564864, 45, 36.77547104)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (-0.309991 \leq 1, 1.000004 \leq 1, -0.021379 \leq 1, 0.621403 \leq 1, 1.000002 \leq 1, 0.681516 \leq 1)$
12	0.012692	$\mathbf{x}_{\text{RGA-PSO}}^* = (0.050849843, 0.336663305, 12.579603478)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (-0.000143 \leq 0, -0.000469 \leq 0, -4.009021 \leq 0, -0.741658 \leq 0)$
13	5885.3018	$\mathbf{x}_{\text{RGA-PSO}}^* = (0.77816852, 0.38464913, 40.31961883, 199.99999988)$ $\mathcal{G}_m(\mathbf{x}_{\text{RGA-PSO}}^*) = (1.23E - 07 \leq 0, 3.36E - 08 \leq 0, -0.006916 \leq 0, -40.00000012 \leq 0)$

Table 4: The best solutions obtained using the AIA-PSO algorithm from TPs 1–13.

TP number	$f(\mathbf{x}_{\text{AIA-PSO}}^*)$	$\mathbf{x}_{\text{AIA-PSO}}^*$
1	24.358	$\mathbf{x}_{\text{AIA-PSO}}^* = (2.18024296, 2.35746157, 8.75670935, 5.11326109, 1.03976363, 1.54784227, 1.32994030, 9.83127443, 8.27618717, 8.32717779)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (-0.000071 \leq 0, -0.003699 \leq 0, -0.000440 \leq 0, -0.684025 \leq 0, -0.000086 \leq 0, -0.001024 \leq 0, -5.973866 \leq 0, -47.371958 \leq 0)$
2	-30665.539	$\mathbf{x}_{\text{AIA-PSO}}^* = (78, 33, 29.99525527, 45, 36.77581286)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (-92.000000 \leq 0, 5.57E - 08 \leq 0, -8.840500 \leq 0, -11.159500 \leq 0, 2.76E - 07 \leq 0, -5.000000 \leq 0)$
3	680.633	$\mathbf{x}_{\text{AIA-PSO}}^* = (2.32925164, 1.95481519, -0.47307614, 4.35691576, -0.62313420, 1.05236194, 1.59750978)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (-4.26E - 06 \leq 0, -252.612733 \leq 0, -144.741194 \leq 0, -1.00E - 05 \leq 0)$
4	-15	$\mathbf{x}_{\text{AIA-PSO}}^* = (1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (0 \leq 0, 0 \leq 0, 0 \leq 0, -5 \leq 0, -5 \leq 0, -5 \leq 0, 0 \leq 0, 0 \leq 0, 0 \leq 0)$
5	1227.1598	$\mathbf{x}_{\text{AIA-PSO}}^* = (1697.91645044, 53.78132825, 3031.07989059, 90.12651301, 94.99999995, 10.48103708, 153.53594861)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (1.000000 \leq 1, 0.980100 \leq 1, 1.000002 \leq 1, 0.980099 \leq 1, 0.990562 \leq 1, 1.000001 \leq 1, 1.000002 \leq 1, 0.976715 \leq 1, 1.000001 \leq 1, 0.459425 \leq 1, 0.387515 \leq 1, 0.981856 \leq 1, 0.987245 \leq 1, -8.302533 \leq 1)$
6	3.9516	$\mathbf{x}_{\text{AIA-PSO}}^* = (6.44373647, 2.23335111, 0.68233303, 0.60026617, 5.93745119, 5.53146186, 1.01862958, 0.40673661)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (0.999999 \leq 1, 0.999999 \leq 1, 0.999994 \leq 1, 0.999985 \leq 1)$
7	-5.7398	$\mathbf{x}_{\text{AIA-PSO}}^* = (8.13042985, 0.61758136, 0.56393603, 5.63618191)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (0.999999 \leq 1, 0.999998 \leq 1)$
8	-83.2497	$\mathbf{x}_{\text{AIA-PSO}}^* = (88.35635930, 7.67202113, 1.31765768)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (3.00E - 18 \leq 1)$
9	-6.0467	$\mathbf{x}_{\text{AIA-PSO}}^* = (6.46393173, 0.67575021, 1.01188804, 5.94072071, 2.24639462, 0.60683404, 0.39677469, 5.52596342)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (0.999980 \leq 1, 0.999999 \leq 1, 0.998739 \leq 1, 0.999928 \leq 1)$
10	6299.8395	$\mathbf{x}_{\text{AIA-PSO}}^* = (108.97708780, 85.02248757, 204.45439729)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (1.000003 \leq 1)$
11	10122.4852	$\mathbf{x}_{\text{AIA-PSO}}^* = (78, 33, 29.99571251, 45, 36.77534593)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (-0.309991 \leq 1, 1.000001 \leq 1, -0.021380 \leq 1, 0.621403 \leq 1, 1.000001 \leq 1, 0.681517 \leq 1)$
12	0.012667	$\mathbf{x}_{\text{AIA-PSO}}^* = (0.05164232, 0.35558085, 11.35742676)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (-8.83E - 05 \leq 0, -3.04E - 05 \leq 0, -4.050924 \leq 0, -0.728518 \leq 0)$
13	5885.3310	$\mathbf{x}_{\text{AIA-PSO}}^* = (0.77816843, 0.38464909, 40.31961929, 199.99999330)$ $g_m(\mathbf{x}_{\text{AIA-PSO}}^*) = (2.22E - 07 \leq 0, 7.80E - 08 \leq 0, -0.006015 \leq 0, -40.000007 \leq 0)$

Table 5: The best parameter settings of the best solution obtained using the RGA-PSO algorithm from TPs 1–13.

TP number	χ	c_1	c_2	ρ	$p_{m,PSO}$
1	0.73759998	1.52939998	1.72753568	28709280994	0.3750
2	0.67057537	0.45010388	2	1000000000	0.2709
3	0.75493696	0.36925226	1.91198475	54900420223	0.3856
4	0.45438391	1.45998477	1.25508289	1000000000	0.1
5	1	0.77483233	2	1000000000	0.5
6	0.70600559	0.82360083	0.91946627	26622414511	0.1619
7	0.74584341	0.95474855	1.17537957	86557786169	0.1958
8	1	0.69725160	1.53028620	1000000000	0.1675
9	0.41638718	0.46594542	1.97807798	12976330236	0.3686
10	0.25378610	0.59619170	0.83891277	1000000000	0.1
11	0.48183123	2	2	1000000000	0.1
12	0.76190087	0.1	1.16855713	100000000000	0.5
13	0.71783704	1.39420750	1.33590124	29071187026	0.2098

Table 6: The best parameter settings of the best solution obtained using the AIA-PSO algorithm from TPs 1–13.

TP number	χ	c_1	c_2	ρ	$p_{m,PSO}$
1	0.44896780	0.71709211	1.93302154	72140307110	0.4058
2	1	1.36225259	1.97905466	182831007	0.5
3	0.46599492	0.90346435	1.89697456	69125199709	0.2613
4	0.98124982	0.27882671	0.87437226	85199047430	0.1
5	0.99082484	0.1	1.11371788	4231387044	0.5
6	0.82869043	0.88773247	2	1387448505	0.5
7	0.87571243	1.89936723	0.74306310	94752095153	0.2194
8	0.93583844	1.53906226	1.30374874	22520728225	0.1798
9	1	0.22556712	1.52263349	67578847151	0.4507
10	1	1.93003999	0.1	1351461763	0.5
11	1	1.51209364	1.63826995	1811017789	0.5
12	0.52068067	0.1	2	81914376144	0.1
13	0.82395890	1.60107152	0.93611204	17767111886	0.1813

GA-2 represents a GA with a penalty function, but without any penalty parameter, as used by Deb [21]. Also, GA-3 is an RGA with a static penalty function, as developed by Wu and Chung [9]. Notably, AIA-1 is an AIA method called CLONALG, as proposed by Cruz-Cortés et al. [22]. Finally, AIA-2 is an AIA approach based on an adaptive penalty function, as developed by Wu [10]. The numerical results of GA-1, GA-2, and AIA-1 methods for solving TPs 1–4 were collected from the published literature [20–22]. Furthermore, the GA-1, GA-2, and AIA-1 approaches were executed under 350,000 objective function evaluations. To fairly compare the performances of the proposed hybrid CI algorithms and the individual GA and AIA approaches, the GA-3, AIA-2, the internal PSO algorithm of RGA-PSO method, and the internal PSO algorithm of AIA-PSO method were independently executed 50 times under 350,000 objective function evaluations for solving TPs 1–4.

For solving TP 1, the median values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the GA-1, GA-3, and AIA-2 approaches, and

Table 7: Comparison of the results of the proposed RGA-PSO and AIA-PSO algorithms and those of the published individual GA and AIA approaches for TPs 1–4.

TP number	Global optimum	Methods	Best	Mean	Median	Worst	MAPE%
1	24.306	GA-1 [20]	24.690	—	29.258	36.060	—
		GA-2 [21]	24.372	—	24.409	25.075	—
		GA-3 [9]	24.935	27.314	27.194	33.160	12.377
		AIA-1 [22]	24.506	25.417	—	26.422	—
		AIA-2 [10]	24.377	24.669	24.663	24.988	1.495
		The proposed RGA-PSO	24.323	24.568	24.521	24.831	1.078
		The proposed AIA-PSO	24.358	24.579	24.564	24.809	1.125
2	-30665.539	GA-1 [20]	—	—	—	—	—
		GA-2 [21]	—	—	—	—	—
		GA-3 [9]	-30665.526	-30662.922	-30664.709	-30632.445	8.53E - 03
		AIA-1 [22]	-30665.539	-30665.539	—	-30665.539	—
		AIA-2 [10]	-30665.539	-30665.526	-30665.527	-30665.506	4.20E - 05
		The proposed RGA-PSO	-30665.539	-30665.539	-30665.539	-30665.534	1.34E - 06
		The proposed AIA-PSO	-30665.539	-30665.539	-30665.539	-30665.539	9.95E - 07
3	680.630	GA-1 [20]	680.642	—	680.718	680.955	—
		GA-2 [21]	680.634	—	680.642	680.651	—
		GA-3 [9]	680.641	680.815	680.768	681.395	2.72E - 02
		AIA-1 [22]	680.631	680.652	—	680.697	—
		AIA-2 [10]	680.634	680.653	680.650	680.681	3.45E - 03
		The proposed RGA-PSO	680.632	680.640	680.639	680.658	1.46E - 03
		The proposed AIA-PSO	680.633	680.640	680.640	680.657	1.50E - 03
4	-15	GA-1 [20]	-15	-15	-15	-15	—
		GA-2 [21]	—	—	—	—	—
		GA-3 [9]	-13.885	-12.331	-12.267	-10.467	17.795
		AIA-1 [22]	-14.987	-14.726	—	-12.917	—
		AIA-2 [10]	-14.998	-14.992	-14.992	-14.988	5.08E - 02
		The proposed RGA-PSO	-15	-15	-15	-15	1.27E - 06
		The proposed AIA-PSO	-15	-15	-15	-15	1.20E - 10

(The “—” denotes unavailable information.)

the worst values obtained using RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the GA-1, GA-2, GA-3, AIA-1, and AIA-2 approaches. For solving TP 2, the median and worst values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the GA-3 method. For solving TP 3, the median and worst values

Table 8: Results of the t -test for TPs 1–4.

TP number	GA-3 versus AIA-2	GA-3 versus RGA-PSO	GA-3 versus AIA-PSO	AIA-2 versus RGA-PSO	AIA-2 versus AIA-PSO	RGA-PSO versus AIA-PSO
1	0.000*	0.000*	0.000*	0.000*	0.001*	0.643
2	0.003*	0.003*	0.003*	0.000*	0.000*	0.478
3	0.000*	0.000*	0.000*	0.000*	0.000*	0.839
4	0.000*	0.000*	0.000*	0.000*	0.000*	0.294

*Represents that the mean values obtained using two algorithms are statistically different.)

obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the GA-1 and GA-3 approaches. For solving TP 4, the median and worst values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the GA-3 method, and the worst values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the AIA-1 approach. Moreover, the GA-3 method obtained the worst MAPE% for TP 1 and TP 4. Table 8 lists the results of the t -test for the GA-3, AIA-2, RGA-PSO, and AIA-PSO methods. This table indicates that the mean values of the RGA-PSO, and AIA-PSO algorithms are not statistically significant, since P values are larger than a significant level 0.05, and the mean values between GA-3 versus AIA-2, GA-3 versus RGA-PSO, GA-3 versus AIA-PSO, AIA-2 versus RGA-PSO, and AIA-2 versus AIA-PSO are statistically significant. According to Tables 7 and 8, the mean values obtained using the RGA-PSO and AIA-PSO algorithms are better than those of obtained using the GA-3 and AIA-1 methods for TPs 1–4.

Table 9 compares the numerical results obtained using the proposed RGA-PSO and AIA-PSO algorithms and those obtained using AIA-2 and GA-3 for solving TPs 5–13. The AIA-2, GA-3, the internal PSO algorithm of the RGA-PSO approach, and the internal PSO algorithm of AIA-PSO approach were independently executed 50 times under 300,000 objective function evaluations. Table 9 shows that MAPE% obtained using the proposed RGA-PSO and AIA-PSO algorithms is close to 1%, or smaller than 1% for TPs 5–11, indicating that the proposed RGA-PSO and AIA-PSO algorithms can converge to global optimum for TPs 5–11. Moreover, the worst values obtained using the RGA-PSO and AIA-PSO algorithms are significantly smaller than those obtained using the GA-3 method for TPs 5, 6, 11, and 13. Additionally, the worst values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those obtained using the AIA-2 method for TPs 5, 6, and 13.

Table 10 summarizes the results of the t -test for TPs 5–13. According to Tables 9 and 10, the mean values of the RGA-PSO and AIA-PSO algorithms are smaller than those of the GA-3 approach for TPs 5, 6, 7, 8, 9, 10, 11, and 13. Moreover, the mean values obtained using the RGA-PSO and AIA-PSO algorithms are smaller than those of the AIA-2 approach for TPs 6, 7, 8, 10, and 12. Totally, according to Tables 7–10, the performances of the hybrid CI methods are superior to those of individual GA and AIA methods.

The TPs 12 and 13 have been solved by many hybrid algorithms. For instance, Huang et al. [23] presented a coevolutionary differential evolution (CDE) that integrates a coevolution mechanism and a DE approach. Zahara and Kao [24] developed a hybrid Nelder-Mead simplex search method and a PSO algorithm (NM-PSO). Table 11 compares

Table 9: Comparison of the numerical results of the proposed RGA-PSO and AIA-PSO algorithms and those of the published individual AIA and RGA for TPs 5–13.

TP number	Global optimum	Methods	Best	Mean	Median	Worst	MAPE%
5	1227.1978	AIA-2 [10]	1227.3191	1228.6097	1227.8216	1239.5205	1.15E-01
		GA-3 [9]	1228.5118	1279.3825	1263.2589	1481.3710	4.266
6	3.9511	The proposed RGA-PSO	1227.2321	1228.8471	1228.3935	1231.6003	1.34E-01
		The proposed AIA-PSO	1227.1598	1228.1800	1227.9762	1229.6590	8.02E-02
		AIA-2 [10]	3.9518	4.1005	4.0460	4.2997	3.781
		GA-3 [9]	3.9697	4.1440	4.1358	4.3743	4.881
7	-5.7398	The proposed RGA-PSO	3.9521	3.9577	3.9566	3.9722	1.67E-01
		The proposed AIA-PSO	3.9516	3.9581	3.9569	3.9735	1.77E-01
		AIA-2 [10]	-5.7398	-5.7398	-5.7398	-5.7396	6.03E-04
		GA-3 [9]	-5.7398	-5.7375	-5.7383	-5.7316	3.98E-02
8	-83.254	The proposed RGA-PSO	-5.7398	-5.7398	-5.7398	-5.7398	2.54E-04
		The proposed AIA-PSO	-5.7398	-5.7398	-5.7398	-5.7398	2.42E-04
		AIA-2 [10]	-83.2504	-83.2499	-83.2499	-83.2496	4.93E-03
		GA-3 [9]	-83.2497	-83.2490	-83.2494	-83.2461	6.06E-03
9	-6.0482	The proposed RGA-PSO	-83.2497	-83.2497	-83.2497	-83.2497	5.13E-03
		The proposed AIA-PSO	-83.2497	-83.2497	-83.2497	-83.2497	5.13E-03
		AIA-2 [10]	-6.0471	-5.9795	-6.0138	-5.7159	1.136
		GA-3 [9]	-6.0417	-5.8602	-5.8772	-5.6815	3.108
10	6300	The proposed RGA-PSO	-6.0441	-5.9883	-5.9968	-5.8051	0.991
		The proposed AIA-PSO	-6.0466	-5.9808	-5.9969	-5.8274	1.082
		AIA-2 [10]	6299.8232	6299.8388	6299.8403	6299.8494	2.56E-03
		GA-3 [9]	6299.8399	6299.8546	6299.8453	6299.9407	2.31E-03
11	10122.6964	The proposed RGA-PSO	6299.8374	6299.8412	6299.8419	6299.8428	2.52E-03
		The proposed AIA-PSO	6299.8395	6299.8420	6299.8423	6299.8425	2.51E-03
		AIA-2 [10]	10122.4343	10122.5014	10122.5023	10122.6271	1.93E-03
		GA-3 [9]	10122.4809	10125.2441	10123.9791	10155.1349	2.58E-04
12	—	The proposed RGA-PSO	10122.4732	10122.4925	10122.4925	10122.6444	2.01E-03
		The proposed AIA-PSO	10122.4852	10122.4920	10122.4927	10122.4931	2.02E-03
		AIA-2 [10]	0.012665	0.012703	0.012719	0.012725	—
		GA-3 [9]	0.012665	0.012739	0.012717	0.013471	—
12	—	The proposed RGA-PSO	0.012692	0.012724	0.012721	0.012784	—
		The proposed AIA-PSO	0.012667	0.012715	0.012719	0.012778	—

Table 9: Continued.

TP number	Global optimum	Methods	Best	Mean	Median	Worst	MAPE%
13	—	AIA-2 [10]	5885.5312	5900.4860	5894.7929	6014.2198	—
		GA-3 [9]	5885.3076	5943.4714	5897.2904	6289.7314	—
		The proposed RGA-PSO	5885.3018	5895.0381	5885.3326	6005.4351	—
		The proposed AIA-PSO	5885.3310	5886.5426	5885.3323	5906.7404	—

(The “—” denotes unavailable information.)

Table 10: Results of *t*-test for TPs 5–13.

TP number	GA-3 versus AIA-2	GA-3 versus RGA-PSO	GA-3 versus AIA-PSO	AIA-2 versus RGA-PSO	AIA-2 versus AIA-PSO	RGA-PSO versus AIA-PSO
5	0.000*	0.000*	0.000*	0.485	0.163	0.002*
6	0.112	0.000*	0.000*	0.000*	0.000*	0.682
7	0.000*	0.000*	0.000*	0.000*	0.000*	0.358
8	0.000*	0.000*	0.000*	0.000*	0.000*	0.291
9	0.000*	0.000*	0.000*	0.516	0.814	0.577
10	0.000*	0.000*	0.000*	0.009*	0.001*	0.001*
11	0.000*	0.000*	0.000*	0.069	0.013*	0.884
12	0.049*	0.389	0.178	0.000*	0.007*	0.014*
13	0.003*	0.001*	0.000*	0.240	0.000*	0.019*

*Represents that the mean values obtained using two algorithms are statistically different.)

Table 11: Comparison of the numerical results of the proposed RGA-PSO and AIA-PSO algorithms and those of the published hybrid algorithms for TPs 12-13.

TP number	Methods	Best	Mean	Median	Worst	S.D.
12	CDE [23]	0.0126702	0.012703	—	0.012790	2.7E – 05
	NM-PSO [24]	0.0126302	0.0126314	—	0.012633	8.73E – 07
	The proposed RGA-PSO	0.012692	0.012724	0.012721	0.012784	1.46E – 05
	The proposed AIA-PSO	0.012667	0.012715	0.012719	0.012778	2.00E – 05
13	CDE [23]	6059.7340	6085.2303	—	6371.0455	43.01
	NM-PSO [24]	5930.3137	5946.7901	—	5960.0557	9.16
	The proposed RGA-PSO	5885.3018	5895.0381	5885.3326	6005.4351	24.33
	The proposed AIA-PSO	5885.3310	5886.5426	5885.3323	5906.7404	4.54

the numerical results of the CDE, NM-PSO, RGA-PSO, and AIA-PSO methods for solving TPs 12–13. The table indicates that the best, mean, and worst values obtained using the NM-PSO method are superior to those obtained using the CDE, RGA-PSO, and AIA-PSO approaches for TP 12. Moreover, the best, mean, and worst values obtained using the AIA-PSO algorithm are better than those of the CDE, NM-PSO, and RGA-PSO algorithms.

According to the No Free Lunch theorem [35], if algorithm A outperforms algorithm B on average for one class of problems, then the average performance of the former must be worse than that of the latter over the remaining problems. Therefore, it is unlikely that any unique stochastic global optimization approach exists that performs best for all CGO problems.

4.3. Summary of Results

The proposed RGA-PSO and AIA-PSO algorithms with a penalty function method have the following benefits.

- (1) Parameter manipulation of the internal PSO algorithm is based on the solved CGO problems. Owing to their ability to efficiently solve an UGO problem, the external RGA and AIA approaches are substituted for trial and error to manipulate the parameters (χ , c_1 , c_2 , ρ , and $p_{m,PSO}$).
- (2) Besides obtaining the optimum parameter settings of the internal PSO algorithm, the RGA-PSO and AIA-PSO algorithms can yield a global optimum for a CGO problem.
- (3) In addition to performing better than approaches of some published individual GA and AIA approaches, the proposed RGA-PSO and AIA-PSO algorithms reduce the parametrization for the internal PSO algorithm, despite the RGA-PSO and AIA-PSO algorithms being more complex than individual GA and AIA approaches.

The proposed RGA-PSO and AIA-PSO algorithms have the following limitations.

- (1) The proposed RGA-PSO and AIA-PSO algorithms increase the computational CPU time, as shown in Table 2.
- (2) The proposed RGA-PSO and AIA-PSO algorithms are designed to solve CGO problems with continuous decision variables x_n . Therefore, the proposed algorithms cannot be applied to manufacturing problems such as job shop scheduling and quadratic assignment problems (combinatorial optimization problems).

5. Conclusions

This work presents novel RGA-PSO and AIA-PSO algorithms. The synergistic power of the RGA with PSO algorithm and the AIA with PSO algorithm is also demonstrated by using 13 CGO problems. Numerical results indicate that, in addition to converging to a global minimum for each test CGO problem, the proposed RGA-PSO and AIA-PSO algorithms obtain the optimum 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 alternative stochastic global optimization methods such as individual GA and AIA approaches. The RGA-PSO and AIA-PSO algorithms are highly promising stochastic global optimization approaches for solving CGO problems.

Appendices

A. TP 1 [20, 21]

TP 1 has ten decision variables, eight inequality constraints, and 20 boundary conditions, as follows:

$$\begin{aligned}
 \text{Minimize } f(\mathbf{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\
 &\quad + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 \\
 &\quad + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\
 \text{Subject to } g_1(\mathbf{x}) &\equiv -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0, \\
 g_2(\mathbf{x}) &\equiv 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0, \\
 g_3(\mathbf{x}) &\equiv -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0, \\
 g_4(\mathbf{x}) &\equiv 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0, \\
 g_5(\mathbf{x}) &\equiv 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0, \\
 g_6(\mathbf{x}) &\equiv x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0, \\
 g_7(\mathbf{x}) &\equiv 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0, \\
 g_8(\mathbf{x}) &\equiv -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0, \\
 &\quad -10 \leq x_n \leq 10, \quad n = 1, 2, \dots, 10.
 \end{aligned} \tag{A.1}$$

The global solution to TP 1 is as follows:

$$\begin{aligned}
 \mathbf{x}^* &= (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, \\
 &\quad 1.430574, 1.321644, 9.828726, 8.280092, 8.375927), \\
 f(\mathbf{x}^*) &= 24.306.
 \end{aligned} \tag{A.2}$$

B. TP 2 [21]

TP 2 involves five decision variables, six inequality constraints, and ten boundary conditions, as follows:

$$\begin{aligned}
 \text{Minimize } f(\mathbf{x}) &= 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
 \text{Subject to } g_1(\mathbf{x}) &\equiv -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0, \\
 g_2(\mathbf{x}) &\equiv -6.665593 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \leq 0, \\
 g_3(\mathbf{x}) &\equiv 9.48751 - 0.0071371x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 \leq 0, \\
 g_4(\mathbf{x}) &\equiv -29.48751 + 0.0071371x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leq 0, \\
 g_5(\mathbf{x}) &\equiv 10.669039 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 \leq 0, \\
 g_6(\mathbf{x}) &\equiv -15.699039 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leq 0, \\
 &78 \leq x_1 \leq 102, \quad 33 \leq x_2 \leq 45, \quad 27 \leq x_n \leq 45, \quad n = 3, 4, 5.
 \end{aligned} \tag{B.1}$$

The global solution to TP 2 is

$$\begin{aligned}
 \mathbf{x}^* &= (78.0, 33.0, 29.995256, 45.0, 36.775812), \\
 f(\mathbf{x}^*) &= -30665.539.
 \end{aligned} \tag{B.2}$$

C. TP 3 [20, 21]

TP 3 has seven decision variables, four inequality constraints, and 14 boundary conditions, as follows:

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

The global solution to TP 3 is

$$\begin{aligned} \mathbf{x}^* &= (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227), \\ f(\mathbf{x}^*) &= 680.630. \end{aligned} \quad (\text{C.2})$$

D. TP 4 [20, 21]

TP 4 involves 13 decision variables, nine inequality constraints, and 26 boundary conditions, as follows:

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

The global solution to TP 4 is

$$\mathbf{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1), \quad f(\mathbf{x}^*) = -15. \quad (\text{D.2})$$

E. TP 5 (Alkylation Process Design Problem in Chemical Engineering) [1]

TP 5 has seven decision variables subject to 12 nonconvex, two linear, and 14 boundary constraints. The objective function is to improve the octane number of some olefin feed

by reacting it with isobutane in the presence of acid. The decision variables x_n are olefin feed rate (barrels/day) x_1 , acid addition rate (thousands of pounds/day) x_2 , alkylate yield (barrels/day) x_3 , acid strength x_4 , motor octane number x_5 , external isobutane-to-olefin ratio x_6 , and F-4 performance number x_7 :

$$\begin{aligned}
 &\text{Minimize } f(\mathbf{x}) = (\omega_1 x_1 + \omega_2 x_1 x_6 + \omega_3 x_3 + \omega_4 x_2 + \omega_5 - \omega_6 x_3 x_5) \\
 &\text{s.t. } g_1(\mathbf{x}) = \omega_7 x_6^2 + \omega_8 x_1^{-1} x_3 - \omega_9 x_6 \leq 1, \\
 &g_2(\mathbf{x}) = \omega_{10} x_1 x_3^{-1} + \omega_{11} x_1 x_3^{-1} x_6 - \omega_{12} x_1 x_3^{-1} x_6^2 \leq 1, \\
 &g_3(\mathbf{x}) = \omega_{13} x_6^2 + \omega_{14} x_5 - \omega_{15} x_4 - \omega_{16} x_6 \leq 1, \\
 &g_4(\mathbf{x}) = \omega_{17} x_5^{-1} + \omega_{18} x_5^{-1} x_6 + \omega_{19} x_4 x_5^{-1} - \omega_{20} x_5^{-1} x_6^2 \leq 1, \\
 &g_5(\mathbf{x}) = \omega_{21} x_7 + \omega_{22} x_2 x_3^{-1} x_4^{-1} - \omega_{23} x_2 x_3^{-1} \leq 1, \\
 &g_6(\mathbf{x}) = \omega_{24} x_7^{-1} + \omega_{25} x_2 x_3^{-1} x_7^{-1} - \omega_{26} x_2 x_3^{-1} x_4^{-1} x_7^{-1} \leq 1, \\
 &g_7(\mathbf{x}) = \omega_{27} x_5^{-1} + \omega_{28} x_5^{-1} x_7 \leq 1, \\
 &g_8(\mathbf{x}) = \omega_{29} x_5 - \omega_{30} x_7 \leq 1, \\
 &g_9(\mathbf{x}) = \omega_{31} x_3 - \omega_{32} x_1 \leq 1, \\
 &g_{10}(\mathbf{x}) = \omega_{33} x_1 x_3^{-1} + \omega_{34} x_3^{-1} \leq 1, \\
 &g_{11}(\mathbf{x}) = \omega_{35} x_2 x_3^{-1} x_4^{-1} - \omega_{36} x_2 x_3^{-1} \leq 1, \\
 &g_{12}(\mathbf{x}) = \omega_{37} x_4 + \omega_{38} x_2^{-1} x_3 x_4 \leq 1, \\
 &g_{13}(\mathbf{x}) = \omega_{39} x_1 x_6 + \omega_{40} x_1 - \omega_{41} x_3 \leq 1, \\
 &g_{14}(\mathbf{x}) = \omega_{42} x_1^{-1} x_3 + \omega_{43} x_1^{-1} - \omega_{44} x_6 \leq 1, \\
 &1500 \leq x_1 \leq 2000, \quad 1 \leq x_2 \leq 120, \quad 3000 \leq x_3 \leq 3500, \\
 &85 \leq x_4 \leq 93, \quad 90 \leq x_5 \leq 95, \quad 3 \leq x_6 \leq 12, \quad 145 \leq x_7 \leq 162,
 \end{aligned} \tag{E.1}$$

where ω_l ($l = 1, 2, \dots, 44$) denotes positive parameters given in Table 12. The global solution to TP 5 is

$$\mathbf{x}^* = (1698.18, 53.66, 3031.30, 90.11, 10.50, 153.53), \quad f(\mathbf{x}^*) = 1227.1978. \tag{E.2}$$

Table 12: Coefficients for TP 5.

l	ω_l	l	ω_l	l	ω_l
1	1.715	16	$0.19120592E - 1$	31	0.00061000
2	0.035	17	$0.56850750E + 2$	32	0.0005
3	4.0565	18	1.08702000	33	0.81967200
4	10.000	19	0.32175000	34	0.81967200
5	3000.0	20	0.03762000	35	24500.0
6	0.063	21	0.00619800	36	250.0
7	$0.59553571E - 2$	22	$0.24623121E + 4$	37	$0.10204082E - 1$
8	0.88392857	23	$0.25125634E + 2$	38	$0.12244898E - 4$
9	0.11756250	24	$0.16118996E + 3$	39	0.00006250
10	1.10880000	25	5000.0	40	0.00006250
11	0.13035330	26	$0.48951000E + 6$	41	0.00007625
12	0.00660330	27	$0.44333333E + 2$	42	1.22
13	$0.66173269E - 3$	28	0.33000000	43	1.0
14	$0.17239878E - 1$	29	0.02255600	44	1.0
15	$0.56595559E - 2$	30	0.00759500		

F. TP 6 (Optimal Reactor Design Problem) [1]

TP 6 contains eight decision variables subject to four nonconvex inequality constraints and 16 boundary conditions, as follows:

$$\begin{aligned}
 &\text{Minimize } f(\mathbf{x}) = \left(0.4x_1^{0.67}x_7^{-0.67} + 0.4x_2^{0.67}x_8^{-0.67} + 10 - x_1 - x_2\right) \\
 &\text{s.t. } g_1(\mathbf{x}) = 0.0588x_5x_7 + 0.1x_1 \leq 1, \\
 &\quad g_2(\mathbf{x}) = 0.0588x_6x_8 + 0.1x_1 + 0.1x_2 \leq 1, \\
 &\quad g_3(\mathbf{x}) = 4x_3x_5^{-1} + 2x_3^{-0.71}x_5^{-1} + 0.0588x_3^{-1.3}x_7 \leq 1. \\
 &\quad g_4(\mathbf{x}) = 4x_4x_6^{-1} + 2x_4^{-0.71}x_6^{-1} + 0.0588x_4^{-1.3}x_8 \leq 1, \quad 0.1 \leq x_n \leq 10, \quad n = 1, 2, \dots, 8.
 \end{aligned} \tag{F.1}$$

The global solution to TP 6 is

$$\mathbf{x}^* = (6.4747, 2.2340, 0.6671, 0.5957, 5.9310, 5.5271, 1.0108, 0.4004), \quad f(\mathbf{x}^*) = 3.9511. \tag{F.2}$$

G. TP 7 [1]

TP 7 has four decision variables, two nonconvex inequality constraints, and eight boundary conditions, as follows:

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) &= \left(-x_1 + 0.4x_1^{0.67}x_3^{-0.67}\right) \\ \text{s.t. } g_1(\mathbf{x}) &= 0.05882x_3x_4 + 0.1x_1 \leq 1 \\ g_2(\mathbf{x}) &= 4x_2x_4^{-1} + 2x_2^{-0.71}x_4^{-1} + 0.05882x_2^{-1.3}x_3 \leq 1, \quad 0.1 \leq x_1, x_2, x_3, x_4 \leq 10. \end{aligned} \quad (\text{G.1})$$

The global solution to TP 7 is

$$\mathbf{x}^* = (8.1267, 0.6154, 0.5650, 5.6368), \quad f(\mathbf{x}^*) = -5.7398. \quad (\text{G.2})$$

H. TP 8 [1]

TP 8 contains three decision variables subject to one nonconvex inequality constraint and six boundary conditions, as follows:

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) &= \left(0.5x_1x_2^{-1} - x_1 - 5x_2^{-1}\right) \\ \text{s.t. } g_1(\mathbf{x}) &= 0.01x_2x_3^{-1} + 0.01x_1 + 0.0005x_1x_3 \leq 1, \quad 1 \leq x_n \leq 100, \quad n = 1, 2, 3. \end{aligned} \quad (\text{H.1})$$

The global solution to TP 8 is

$$\mathbf{x}^* = (88.2890, 7.7737, 1.3120), \quad f(\mathbf{x}^*) = -83.2540. \quad (\text{H.2})$$

I. TP 9 [1]

TP 9 contains eight decision variables subject to four nonconvex inequality constraints and 16 boundary conditions, as follows:

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) &= \left(-x_1 - x_5 + 0.4x_1^{0.67}x_3^{-0.67} + 0.4x_5^{0.67}x_7^{-0.67}\right) \\ \text{s.t. } g_1(\mathbf{x}) &= 0.05882x_3x_4 + 0.1x_1 \leq 1, \\ g_2(\mathbf{x}) &= 0.05882x_7x_8 + 0.1x_1 + 0.1x_5 \leq 1, \\ g_3(\mathbf{x}) &= 4x_2x_4^{-1} + 2x_2^{-0.71}x_4^{-1} + 0.05882x_2^{-1.3}x_3 \leq 1, \\ g_4(\mathbf{x}) &= 4x_6x_8^{-1} + 2x_6^{-0.71}x_8^{-1} + 0.05882x_6^{-1.3}x_7 \leq 1, \quad 0.01 \leq x_n \leq 10, \quad n = 1, 2, \dots, 8. \end{aligned} \quad (\text{I.1})$$

The global solution to TP 9 is

$$\mathbf{x}^* = (6.4225, 0.6686, 1.0239, 5.9399, 2.2673, 0.5960, 0.4029, 5.5288), \quad f(\mathbf{x}^*) = -6.0482. \quad (\text{I.2})$$

J. TP 10 [34]

TP 10 contains three decision variables subject to one nonconvex inequality constraint and six boundary conditions, as follows:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = \left(5x_1 + 50000x_1^{-1} + 20x_2 + 72000x_2^{-1} + 10x_3 + 144000x_3^{-1} \right) \\ \text{s.t.} \quad & g_1(\mathbf{x}) = 4x_1^{-1} + 32x_2^{-1} + 120x_3^{-1} \leq 1, \quad 1 \leq x_n \leq 1000, \quad n = 1, 2, 3. \end{aligned} \quad (\text{J.1})$$

The global solution to TP 10 is

$$\mathbf{x}^* = (107.4, 84.9, 204.5), \quad f(\mathbf{x}^*) = 6300. \quad (\text{J.2})$$

K. TP 11 [1, 34]

TP 11 involves five decision variables, six inequality constraints, and ten boundary conditions, as follows:

$$\begin{aligned} \text{Minimize} \quad & g_0(\mathbf{x}) = \left(5.3578x_3^2 + 0.8357x_1x_5 + 37.2392x_1 \right) \\ \text{s.t.} \quad & g_1(\mathbf{x}) = 0.00002584x_3x_5 - 0.00006663x_2x_5 - 0.0000734x_1x_4 \leq 1, \\ & g_2(\mathbf{x}) = 0.000853007x_2x_5 + 0.00009395x_1x_4 - 0.00033085x_3x_5 \leq 1, \\ & g_4(\mathbf{x}) = 0.00024186x_2x_5 + 0.00010159x_1x_2 + 0.00007379x_3^2 \leq 1, \\ & g_3(\mathbf{x}) = 1330.3294x_2^{-1}x_5^{-1} - 0.42x_1x_5^{-1} - 0.30586x_2^{-1}x_3^2x_5^{-1} \leq 1, \\ & g_5(\mathbf{x}) = 2275.1327x_3^{-1}x_5^{-1} - 0.2668x_1x_5^{-1} - 0.40584x_4x_5^{-1} \leq 1, \\ & g_6(\mathbf{x}) = 0.00029955x_3x_5 + 0.00007992x_1x_3 + 0.00012157x_3x_4 \leq 1, \\ & 78 \leq x_1 \leq 102, \quad 33 \leq x_2 \leq 45, \quad 27 \leq x_3 \leq 45, \quad 27 \leq x_4 \leq 45, \quad 27 \leq x_5 \leq 45. \end{aligned} \quad (\text{K.1})$$

The global solution to TP 11 is

$$\mathbf{x}^* = (78.0, 33.0, 29.998, 45.0, 36.7673), \quad f(\mathbf{x}^*) = 10122.6964. \quad (\text{K.2})$$

L. TP 12 (a Tension/Compression String Design Problem) [23]

TP 12 involves three decision variables, six inequality constraints, and six boundary conditions. This problem is taken from Huang et al. [23]. This problem attempts to minimize

the weight (i.e., $f(\mathbf{x})$) of a tension/compression spring subject to constraints on minimum deflection, shear stress, and surge frequency. The design variables are the mean coil diameter x_2 , wire diameter x_1 , and number of active coils x_3 :

$$\begin{aligned}
 &\text{Minimize } f(\mathbf{x}) = (x_3 + 2)x_2x_1^2 \\
 &\text{s.t. } g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\
 &\quad g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\
 &\quad g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 &\quad g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0, \quad 0.05 \leq x_1 \leq 2, \quad 0.25 \leq x_2 \leq 1.3, \quad 2 \leq x_3 \leq 15.
 \end{aligned} \tag{L.1}$$

M. TP 13 (Pressure Vessel Design Problem) [23]

TP 13 involves four decision variables, four inequality constraints, and eight boundary conditions. This problem attempts to minimize the total cost ($f(\mathbf{x})$), including cost of materials forming and welding. A cylindrical vessel is capped at both ends by hemispherical heads. Four design variables exist: thickness of the shell x_1 , thickness of the head x_2 , inner radius x_3 , and length of the cylindrical section of the vessel, excluding the head x_4 :

$$\begin{aligned}
 &\text{Minimize } f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 &\text{s.t. } g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0, \\
 &\quad g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leq 0, \\
 &\quad g_3(\mathbf{x}) = -\pi x_2^3x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\
 &\quad g_4(\mathbf{x}) = x_4 - 240 \leq 0, \quad 0 \leq x_1 \leq 100, \quad 0 \leq x_2 \leq 100, \quad 10 \leq x_3 \leq 200, \quad 10 \leq x_4 \leq 200.
 \end{aligned} \tag{M.1}$$

Acknowledgment

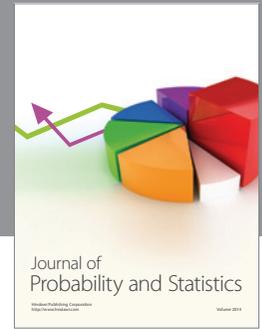
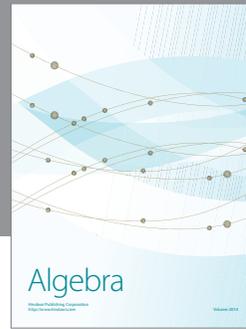
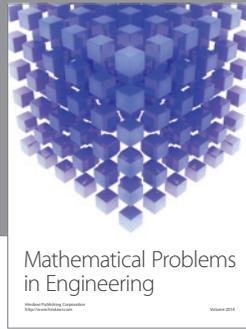
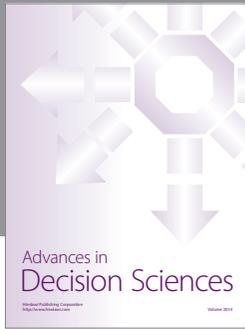
The author would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract no. NSC 100-2622-E-262-006-CC3.

References

- [1] C. A. Floudas, P. M. Pardalos, C. S. Adjiman, and W. R. Esposito, *Handbook of Test Problems in Local and Global Optimization*, Kluwer, Boston, Mass, USA, 1999.

- [2] L. Kit-Nam Francis, "A generalized geometric-programming solution to economic production quantity model with flexibility and reliability considerations," *European Journal of Operational Research*, vol. 176, no. 1, pp. 240–251, 2007.
- [3] J. F. Tsai, "Treating free variables in generalized geometric programming problems," *Computers & Chemical Engineering*, vol. 33, no. 1, pp. 239–243, 2009.
- [4] P. Xu, "A hybrid global optimization method: the multi-dimensional case," *Journal of Computational and Applied Mathematics*, vol. 155, no. 2, pp. 423–446, 2003.
- [5] C. A. Floudas, *Deterministic Global Optimization*, Kluwer Academic, Boston, Mass, USA, 1999.
- [6] C. I. Sun, J. C. Zeng, J. S. Pan et al., "An improved vector particle swarm optimization for constrained optimization problems," *Information Sciences*, vol. 181, no. 6, pp. 1153–1163, 2011.
- [7] I. G. Tsoulos, "Solving constrained optimization problems using a novel genetic algorithm," *Applied Mathematics and Computation*, vol. 208, no. 1, pp. 273–283, 2009.
- [8] K. Deep and Dipti, "A self-organizing migrating genetic algorithm for constrained optimization," *Applied Mathematics and Computation*, vol. 198, no. 1, pp. 237–250, 2008.
- [9] J. Y. Wu and Y. K. Chung, "Real-coded genetic algorithm for solving generalized polynomial programming problems," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 4, pp. 358–364, 2007.
- [10] 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.
- [11] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Communications of the ACM*, vol. 37, no. 3, pp. 77–84, 1994.
- [12] A. Konar, *Computational Intelligence-Principles, Techniques and Applications*, Springer, New York, NY, USA, 2005.
- [13] 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.
- [14] W. F. Abd-El-Wahed, A. A. Mousa, and M. A. El-Shorbagy, "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems," *Journal of Computational and Applied Mathematics*, vol. 235, no. 5, pp. 1446–1453, 2011.
- [15] R. J. Kuo and Y. S. Han, "A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem—a case study on supply chain model," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 3905–3917, 2011.
- [16] 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.
- [17] Y. Hu, Y. Ding, and K. Hao, "An immune cooperative particle swarm optimization algorithm for fault-tolerant routing optimization in heterogeneous wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2012, Article ID 743728, 19 pages, 2012.
- [18] X. Zhao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing*, vol. 10, no. 1, pp. 119–124, 2010.
- [19] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, NY, USA, 1994.
- [20] Z. Michalewicz, "Genetic algorithm, numerical optimization, and constraints," in *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 151–158, San Mateo, Calif, USA, 1995.
- [21] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [22] N. Cruz-Cortés, D. Trejo-Pérez, and C. A. Coello Coello, "Handling constraints in global optimization using an artificial immune system," in *Proceedings of the 4th International Conference on Artificial Immune Systems*, pp. 234–247, Banff, Canada, 2005.
- [23] F.-Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [24] E. Zahara and Y. T. Kao, "Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, no. 2, part 2, pp. 3880–3886, 2009.
- [25] C. R. Houck, J. A. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," in *NSCU-IE TR 95-09*, North Carolina State University, Raleigh, NC, USA, 1995.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, 1995.

- [27] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, Anchorage, Alaska, USA, 1998.
- [28] 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.
- [29] 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.
- [30] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.
- [31] N. K. Jerne, "Idiotypic networks and other preconceived ideas," *Immunological Reviews*, vol. 79, pp. 5–24, 1984.
- [32] L. N. de Castro and F. J. Von Zuben, "Artificial Immune Systems: Part I—Basic Theory and Applications," FEEC/Univ. Campinas, Campinas, Brazil, 1999, ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/tr_dca/trdca0199.pdf.
- [33] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.
- [34] M. J. Rijckaert and X. M. Martens, "Comparison of generalized geometric programming algorithms," *Journal of Optimization Theory and Applications*, vol. 26, no. 2, pp. 205–242, 1978.
- [35] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

