2012 Sixth International Conference on Genetic and Evolutionary Computing

Stochastic Global Optimization Method for Solving Constrained Engineering Design Optimization Problems

Jui-Yu Wu*
Department of Business Administration
Lunghwa University of Science and Technology, Taoyuan, Taiwan
jywu@mail.lhu.edu.tw

***Abstract*—This work presents a stochastic global optimization (SGO) approach, which integrates an artificial immune algorithm and a particle swarm optimization (AIA-PSO) approach. To solve constrained engineering design optimization problems (e.g., tension/compression string design and pressure vessel design problems), the proposed AIA-PSO algorithm uses a penalty function method to transform a constrained engineering design optimization problem to an unconstrained optimization problem. Based on an external AIA approach, the constriction coefficient, cognitive parameter, social parameter, penalty parameter and mutation probability of an internal PSO algorithm are optimized. Constrained engineering design optimization problems are then solved using the internal PSO algorithm. Moreover, numerical results obtained using the proposed AIA-PSO algorithm is compared with those of published individual genetic algorithm (GA) with AIA methods and hybrid algorithms. Experimental results indicate that the optimum parameter settings of the internal PSO algorithm can be obtained using the external AIA approach. Also, the proposed AIA-PSO algorithm performs significantly better than those of some published individual GA with AIA approaches and hybrid algorithms for solving the pressure vessel design problem. Therefore, the proposed AIA-PSO algorithm can be considered as a promising SGO approach for solving constrained engineering design optimization problems.**

Keywords- stochastic global optimization; artificial immune algorithm; particle swarm optimization; constrained engineering design optimization

I. INTRODUCTION

Constrained engineering design optimization problems (e.g., tension/compression string design, pressure vessel design problems [1], alkylation process design, heat exchanger design and optimal reactor design [2]) can be expressed as constrained global optimization (CGO) problems, as follows:

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

$$s.t. \quad g_m(\mathbf{x}) \le 0, \quad m = 1, 2, \ldots, M \tag{2}$$

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

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

where $f(\mathbf{x})$ 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], and N is the total number of decision variables x_n.

For solving CGO problems, global optimization approaches can be divided into deterministic or stochastic [3]. Often involving a sophisticated optimization process, deterministic global optimization methods typically make assumptions regarding the problem to be solved [4]. Genetic algorithms (GA), particle swarm optimization (PSO), artificial immune algorithms (AIAs) and simulated annealing algorithms are stochastic global optimization (SGO) approaches. These methods such as GAs, AIAs and PSO belong to the branches (e.g., evolutionary computing and artificial life) in the computational intelligence (CI) family [5]. These SGO methods that do not require gradient information and many assumptions have received considerable attention [6, 7].

In addition to outperforming individual algorithms in terms of solving certain problems, hybrid algorithms can solve general problems more efficiently [8]. Therefore, hybrid CI approaches have been developed [9, 10]. Above hybrid CI methods focus on developing diverse candidates (such as chromosomes and particles) to solve optimization problems more efficiently.

A standard PSO algorithm has certain limitations [11, 12]. For instance, a PSO algorithm has many parameters that must be set. The exploring and exploiting capabilities for a PSO algorithm are limited to optimum parameter settings. Moreover, a conventional PSO method has premature convergence that is a rapid loss of diversity during the optimization process.

To overcome the above limitations, this work develops a hybrid AIA and PSO (AIA-PSO) algorithm to solve CGO problems efficiently. The proposed AIA-PSO algorithm is considered to optimize two optimization problems simultaneously. Finding optimum specific parameters (e.g., cognitive parameter, social parameter, constriction coefficient, penalty parameter and mutation probability of an internal PSO algorithm based on a penalty function approach) can be viewed as an unconstrained global optimization problem, which is optimized using an external AIA approach. A CGO problem is then solved using an internal PSO algorithm. An attempt is also made to increase

978-0-7695-4763-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ICGEC.2012.134

diversity of candidate solutions of the PSO algorithm by using a multi-non-uniform mutation operation [13].

Performance of the proposed AIA-PSO algorithm is evaluated using two constrained engineering design optimization problems and compared with those of published individual GA and AIA [14, 15] methods and hybrid algorithms [1, 16].

II. RELATED WORKS

A. Artificial Immune Algorithm

Wu [14] developed an AIA method 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**). Hypermutation, receptor editing and bone marrow operations are used to create diverse **Abs** (candidate solutions). Section III describes these operations.

B. Particle Swarm Optimization

Kennedy and Eberhart [17] first introduced a standard PSO algorithm, which is inspired by the social behavior of bird flocks or fish schools. The particle velocities can be updated by Eq. (5), as follows:

$$v_{j,n}(g_{\mathrm{PSO}}+1)=v_{j,n}(g_{\mathrm{PSO}})+c_1 r_{1j}(g_{\mathrm{PSO}})[p^{lb}_{j,n}(g_{\mathrm{PSO}})-x_{j,n}(g_{\mathrm{PSO}})] \\ +c_2 r_{2j}(g_{\mathrm{PSO}})[p^{gb}_{j,n}(g_{\mathrm{PSO}})-x_{j,n}(g_{\mathrm{PSO}})]\; j=1,2,...,ps_{\mathrm{PSO}},\, n=1,2,...,N \tag{5}$$

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

The particle positions can be computed using Eq. (6), as follows:

$$x_{j,n}(g_{\mathrm{PSO}}+1)=x_{j,n}(g_{\mathrm{PSO}})+v_{j,n}(g_{\mathrm{PSO}}+1)\; j=1,2,...,ps_{\mathrm{PSO}},\, n=1,2,...,N \tag{6}$$

Shi and Eberhart [18] presented a modified PSO algorithm with an inertia weight (ω) to control the exploration and exploitation abilities of a PSO algorithm. Furthermore, a constriction coefficient (χ) has been developed to balance the exploration and exploitation trade-off [19-21].

This work uses the parameters ω and χ to modify the velocities of particles, as follows:

$$v_{j,n}(g_{\mathrm{PSO}}+1)=\chi\{\omega v_{j,n}(g_{\mathrm{PSO}})+c_1 r_{1j}(g_{\mathrm{PSO}})[p^{lb}_{j,n}(g_{\mathrm{PSO}})-x_{j,n}(g_{\mathrm{PSO}})] \\ +c_2 r_{2j}(g_{\mathrm{PSO}})[p^{gb}_{j,n}(g_{\mathrm{PSO}})-x_{j,n}(g_{\mathrm{PSO}})]\}\; j=1,2,...,ps_{\mathrm{PSO}},\, n=1,2,...,N \tag{7}$$

where

$\omega=\left(\dfrac{g_{\mathrm{max,PSO}}-g_{\mathrm{PSO}}}{g_{\mathrm{max,PSO}}}\right)$, increased g_{PSO} reduces the ω

$g_{\mathrm{max,PSO}}$ = maximum generation of PSO algorithm

According to Eq. (5), the optimal values of the parameters c_1, c_2 and χ are difficult to obtain when using a trial and error method. Therefore, this work uses an AIA method to optimize these parameter settings.

III. METHOD

Figure 1 shows the pseudo-code of the proposed AIA-PSO algorithm.

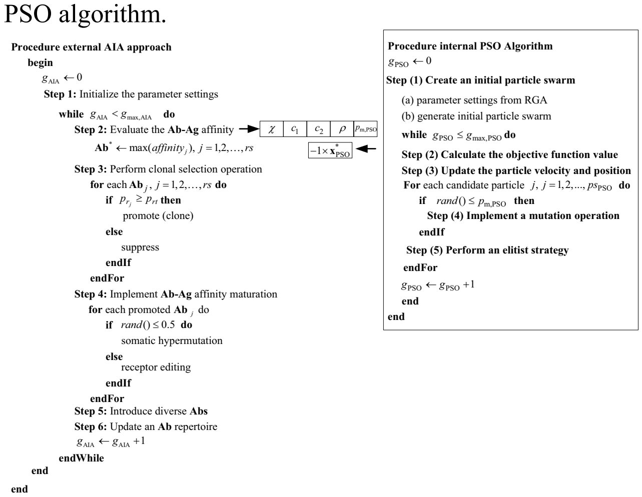

Figure 1. The pseudo-code of the AIA-PSO algorithm

External AIA:

Step 1: Initialize the parameter settings

Various parameters must be predetermined. These include repertoire size rs and the threshold for **Ab-Ab** recognition p_{rt} as well as lower and upper boundaries of these parameters c_1 , c_2 , χ , penalty function ρ and mutation probability of an internal PSO algorithm $p_{\mathrm{m,PSO}}$.

Step 2: Evaluate the Ab-Ag affinity

Internal PSO algorithm:

The external AIA approach offers parameter settings c_1 , c_2 , χ , ρ and $p_{\mathrm{m,PSO}}$ for the internal PSO algorithm, subsequently leading to implementation of internal steps (1)−(5) of the PSO algorithm. The PSO algorithm returns the best fitness value of PSO $f(\mathbf{x}^*_{\mathrm{PSO}})$ to the external AIA approach.

Step (1) Create an initial particle swarm

An initial particle swarm is created based on ps_{PSO} from [x^l_n, x^u_n] of a CGO problem. Particles represent candidate solutions of a CGO problem.

Step (2) Calculate the objective function value

Based on a penalty function method, Eq. (8) is used as the pseudo-objective function value of the internal PSO algorithm, as follows:

$$f_{\text{pseudo},j} = f(\mathbf{x}_{\text{PSO},j}) + \left\{ \rho \times \sum_{m=1}^{M} \left\{ \max\left[0,\ g_m(\mathbf{x}_{\text{PSO},j})\right] \right\}^2 \right\}, \quad (8)$$

$j = 1, 2, \ldots, ps_{\text{PSO}}$

Step (3) Update the particle velocity and position

Equations (6) and (7) can be used to update the particle position and velocity.

Step (4) Implement a mutation operation

Diversity of the particle swarm is increased using the multi-non-uniform mutation [13].

Step (5) Perform an elitist strategy

A new particle swarm (population) is generated from internal step (3). Notably, $f(\mathbf{x}_{\text{PSO},j})$ of a candidate solution j (particle j) in the particle swarm is evaluated. Here, a pair-wise comparison is made between the $f(\mathbf{x}_{\text{PSO},j})$ value 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 is updated to the particle swarm of the next generation.

Internal steps (2) to (5) are repeated until the $g_{\max,\text{PSO}}$ value of the internal PSO algorithm is satisfied.

end

Consistent with the antibody-antigen (**Ab-Ag**) affinity metaphor, Eq. (9) is used to determine an **Ab-Ag** affinity, as follows.

$$\max(affinity_j) = -1 \times f(\mathbf{x}^*_{\text{PSO}}) \quad j = 1, 2, \ldots, rs \quad (9)$$

where $f(\mathbf{x}^*_{\text{PSO}})$ is the best fitness value obtained using the internal PSO method.

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

Step 3: Perform clonal selection operation

The idiotypic network selection operation controls the number of antigen-specific **Abs**. This operation is defined according to **Ab-Ag** and **Ab-Ab** recognition information, as follows.

$$p_{r_j} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{e^{d_{nj}}} \quad (10)$$

$$d_{nj} = \left| \frac{x^*_n - x_{nj}}{x^*_n} \right|, \ j = 1, 2, \ldots, rs, \ \ n = 1, 2, \ldots, N \quad (11)$$

where p_{r_j} = probability that **Ab** j recognizes $\mathbf{Ab}^*$ (the best solution); x^*_n = the best $\mathbf{Ab}^*$ with the highest **Ab-Ag** affinity; x_{nj} = decision variables x_n of **Ab** j.

Step 4: Implement Ab-Ag affinity maturation

The intermediate **Ab** repertoire that is created in external Step 3 is divided into two subsets. These **Abs** undergo somatic hypermutation when the random number equals or is less than 0.5. They suffer receptor editing when the random number exceeds 0.5.

The somatic hypermutation operation can be expressed as follows:

$$x_{\text{trial},n} = \begin{cases} x_{\text{current},n} + (x^u_n - x_{\text{current},n})\, pert(g_{\text{AIA}}), \ \mathit{if}\ U(0,1) < 0.5 \\ x_{\text{current},n} - (x_{\text{current},n} - x^l_n)\, pert(g_{\text{AIA}}), \ \mathit{if}\ U(0,1) \geq 0.5 \end{cases} \quad (12)$$

where

$pert(g_{\text{AIA}}) = \left\{ U_1(0,\ 1)\left(1 - \frac{g_{\text{AIA}}}{g_{\max,\text{AIA}}}\right) \right\}^2$ = perturbation factor; $x_{\text{current},n}$ = current value of decision variable x_n ; $x_{\text{trial},n}$ = trial value of decision variable x_n ; g_{AIA} = current generation of the AIA; $g_{\max,\text{AIA}}$ = maximum generation number of the AIA; $U(0,1)$ and $U_1(0,1)$ = uniform random number in the interval [0, 1].

The Cauchy receptor editing can be expressed by:

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

where $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_N]^T$, vector of Cauchy random variables; $U_2(0,1)$ = uniform random number in the interval [0, 1].

Step 5: Introduce diverse Abs

The bone marrow operation is used to create diverse **Abs** to recruit the **Abs** that are suppressed in external Step 3. The implementation of the bone marrow operation can be found in the literature [12].

Step 6: Update an Ab repertoire

A new **Ab** repertoire is generated from the 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. To maintaining the strong **Abs**, this strategy eliminates non-functional **Abs**.

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

IV. Results

The proposed AIA-PSO algorithm is applied two constrained engineering design optimization problems taken from other work [1], as described in the Appendix. The proposed AIA-PSO algorithm was coded in MATLAB software and executed on a Pentium D 3.0 (GHz) personal computer. Fifty independent runs were conducted to solve the test problem (TP). Numerical results were summarized. They include the best, median, mean, worst, mean computational CPU times (MCCTs) and standard deviation (S.D.) of objective function values obtained using AIA-PSO solutions.

Table I lists the parameter settings of the AIA-PSO algorithm. Table II summarizes the numerical results obtained using the proposed AIA-PSO algorithm. Table III summarizes the best solutions obtained using the AIA-PSO algorithm, indicating that each constraint is satisfied (i.e. an accuracy of at least five decimal places for the violation of each constraint) to each TP. Table IV lists the best parameter settings of the internal PSO algorithm obtained using the AIA approach.

TABLE I. THE PARAMETER SETTINGS OF THE AIA-PSO ALGORITHM

the external AIA method	the internal PSO algorithm
Parameter settings: $p_{rt} = 0.9$, $rs = 10$, $g_{\max,\text{AIA}} = 3$	**Parameter settings:** $g_{\max,\text{PSO}} = 3000$
Search space: $[\chi^l, \chi^u] = [0.1, 1]$, $[c_1^l, c_1^u] = [0.1, 2]$, $[c_2^l, c_2^u] = [0.1, 2]$, $[\rho^l, \rho^u] = [1\times10^9, 1\times10^{11}]$, $[p^l_{\text{m,PSO}}, p^u_{\text{m,PSO}}] = [0.1, 0.5]$	**Search space:** $[x_n^l, x_n^u]$ for a constrained engineering design optimization problem

Table V compares the numerical results obtained using the AIA-PSO algorithm and those obtained using individual AIA [14] with real-coded GA (RGA) [15] approaches and hybrid algorithms (co-evolutionary differential evolution (CDE) [1] and hybrid Nelder-Mead simplex search method and a PSO algorithm (NM-PSO) [16]) for two constrained engineering design optimization problems. Although, the best, mean, worst values and S.D. obtained using the NM-PSO approach are lower than those obtained using the AIA, RGA, CDE and the AIA-PSO approaches for TP 1, their numerical results are identical. Moreover, the best, mean, worst values and S.D. obtained using the proposed AIA-PSO method are significantly lower than those using the CDE, NM-PSO, AIA and RGA methods for TP 2.

V. CONCLUSIONS

This work presents a novel AIA-PSO algorithm. The synergistic power of an AIA approach and a PSO algorithm is also demonstrated by two constrained engineering design optimization problems. Numerical results indicate that the proposed AIA-PSO algorithm obtains the optimum parameter settings of the internal PSO algorithm. Moreover, the numerical results obtained using the AIA-PSO algorithm is significantly superior to those obtained using alternative SGO methods such as individual GA and AIA methods and hybrid algorithms for solving pressure vessel design problem. The AIA-PSO algorithm can be considered as a promising SGO method for solving constrained engineering optimization problems.

TABLE II. NUMERICAL RESULTS OF THE PROPOSED AIA-PSO ALGORITHM

TP No.	Best	Mean	Median	Worst	S.D.	MCCT (sec.)
1	0.012667	0.012715	0.012719	0.012778	2.00E-05	296.15
2	5885.3310	5886.5426	5885.3323	5906.7404	4.54	292.16

TABLE III. THE BEST SOLUTIONS OBTAINED USING THE AIA-PSO ALGORITHM

$f(\mathbf{x}^*_{\text{AIA-PSO}})$	$\mathbf{x}^*_{\text{AIA-PSO}}$
0.012667	$\mathbf{x}^*_{\text{AIA-PSO}}$ =(0.05164232, 0.35558085, 11.35742676) $g_m(\mathbf{x}^*_{\text{AIA-PSO}})$ =(−8.83E-05 ≤ 0, −3.04E-05 ≤ 0, −4.050924 ≤ 0, −0.728518 ≤ 0)
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 ≤ 0, 7.80E-08 ≤ 0, −0.006015 ≤ 0, −40.000007 ≤ 0)

TABLE IV. THE BEST PARAMETER SETTINGS OF THE BEST SOLUTION OBTAINED USING THE AIA-PSO ALGORITHM

χ	c_1	c_2	ρ	$p_{\text{m,PSO}}$
0.52068067	0.1	2	81914376144	0.1
0.82395890	1.60107152	0.93611204	17767111886	0.1813

TABLE V. COMPARISON OF NUMERICAL RESULTS OF THE PROPOSED AIA-PSO ALGORITHM AND THOSE OF THE PUBLISHED INDIVIDUAL AIA WITH RGA APPROACHES AND HYBRID ALGORITHMS

TP No.	Methods	Best	Mean	Median	Worst	S.D.
1	AIA [14]	0.012665	0.012703	0.012719	0.012725	2.17E-05
	RGA [15]	0.012665	0.012739	0.012717	0.013471	1.23E-04
	CDE [1]	0.0126702	0.012703	−	0.012790	2.7E-05
	NM-PSO [16]	0.0126302	0.0126314	−	0.012633	8.73E-07
	the proposed AIA-PSO	0.012667	0.012715	0.012719	0.012778	2.00E-05
2	AIA [14]	5885.5312	5900.4860	5894.7929	6014.2198	21.72
	RGA [15]	5885.3076	5943.4714	5897.2904	6289.7314	97.01
	CDE [1]	6059.7340	6085.2303	−	6371.0455	43.01
	NM-PSO [16]	5930.3137	5946.7901	−	5960.0557	9.16
	the proposed AIA-PSO	5885.3310	5886.5426	5885.3323	5906.7404	4.54

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.

Appendix

TP 1 (a tension/compression string design problem) [1]

$$\text{Minimize} \quad f(\mathbf{x}) = (x_3+2)x_2x_1^2$$

$$s.t.\ g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_4(\mathbf{x}) = \frac{x_1+x_2}{1.5} - 1 \le 0$$

$$0.05 \le x_1 \le 2,\ 0.25 \le x_2 \le 1.3,\ 2 \le x_3 \le 15$$

TP 2 (pressure vessel design problem) [1]

$$\text{Minimize} \quad f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$s.t.\ g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(\mathbf{x}) = -\pi x_2^3x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(\mathbf{x}) = x_4 - 240 \le 0$$

$$0 \le x_1 \le 100,\ 0 \le x_2 \le 100,\ 10 \le x_3 \le 200, 10 \le x_3 \le 200$$

REFERENCES

[1] 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, 2007, pp. 340-356.

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

[3] P. Xu, "A Hybrid Global Optimization Method: The Multi-Dimensional Case," *Journal of Computational and Applied Mathematics*, vol. 155, no. 2, 2003, pp. 423-446.

[4] C. A. Floudas, *Deterministic Global Optimization*, Kluwer, 1999.

[5] A. Konar, *Computational Intelligence-Principles, Techniques and Applications*, Springer, 2005.

[6] I. G. Tsoulos, "Solving Constrained Optimization Problems Using a Novel Genetic Algorithm," *Applied Mathematics and Computation*, vol. 208, no. 1, 2009, pp. 273-283.

[7] E. Mezura-Montes and C. A. Coello Coello, "Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173-194.

[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, 2007, pp. 578-596.

[9] 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, 2011, pp. 3905-3917.

[10] 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, 2011, pp. 1446-1453.

[11] X. Zhao, "A Perturbed Particle Swarm Algorithm for Numerical Optimization," *Applied Soft Computing*, vol. 10, no. 1, 2010, pp. 119-124.

[12] 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, no. Article ID 743728, 2012, pp. 1-19.

[13] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* Springer, 1999.

[14] J. Y. Wu, "Solving Constrained Global Optimization via Artificial Immune System," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 1, 2011, pp. 1-27.

[15] 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, 2007, pp. 358-364.

[16] 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, 2009, pp. 3880-3886.

[17] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.

[18] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," *Proc. The 1998 IEEE International Conference on Evolutionary Computation*, 1998, pp. 69-73.

[19] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," *Proc. The IEEE Congress on Evolutionary Computation*, 1999, pp. 1951-1957.

[20] 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, 2002, pp. 58-73.

[21] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.