International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 412–437
DOI: https://doi.org/10.2991/ijcis.d.201215.005; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Ameliorated Ensemble Strategy-Based Evolutionary Algorithm with Dynamic Resources Allocations

Wali Khan Mashwani[1,], Syed Nouman Ali Shah[1], Samir Brahim Belhaouari[2,], Abdelouahed Hamdi[3,*]

[1] *Institute of Numerical Sciences, Kohat University of Science and Technology, Kohat, Pakistan*
[2] *Division of Information and Computing Technology, College of Science and Engineering Hamad Bin Khalifa University, Education City, Qatar Foundation, Doha, Qatar*
[3] *Department of Mathematics, Statistics and Physics, Qatar University, Doha, Qatar*

ARTICLE INFO

Article History
Received 26 Apr 2020
Accepted 18 Nov 2020

Keywords
Global optimization
Soft computing
Evolutionary computing
Evolutionary algorithms (EAs)
Ensemble strategy-based EAs

ABSTRACT

In the last two decades, evolutionary computing has become the mainstream to attract the attention of the experts in both academia and industrial applications due to the advent of the fast computer with multi-core GHz processors have had a capacity of processing over 100 billion instructions per second. Today's different evolutionary algorithms are found in the existing literature of evolutionary computing that is mainly belong to swarm intelligence and nature-inspired algorithms. In general, it is quite realistic that not always each developed evolutionary algorithms can perform all kinds of optimization and search problems. Recently, ensemble-based techniques are considered to be a good alternative for dealing with various benchmark functions and real-world problems. In this paper, an ameliorated ensemble strategy-based evolutionary algorithm is developed for solving large-scale global optimization problems. The suggested algorithm employs the particle swam optimization, teaching learning-based optimization, differential evolution, and bat algorithm with a self-adaptive procedure to evolve their randomly generated set of solutions. The performance of the proposed ensemble strategy-based evolutionary algorithm evaluated over thirty benchmark functions that are recently designed for the special session of the 2017 IEEE congress of evolutionary computation (CEC'17). The experimental results provided by the suggested algorithm over most CEC'17 benchmark functions are much promising in terms of proximity and diversity.

1. INTRODUCTION

Optimization is the mathematical process of finding either maximum or minimum value of a function in a domain of definition, subject to various constraints on the variable values. A local minimum of a function is a point where the function value is smaller than or equal to the value at nearby points. A global minimum is a point where the function value is smaller than or equal to the value at all other feasible points. Optimization problems can be classified as linear, quadratic, polynomial, nonlinear depending upon the nature of the objective functions and the constraints. Many real-world problems are naturally modeled as global optimization problems [1–8]. A general optimization problem can be modeled as follows:

$$\text{minimize } f_i(x), i = 1, 2, \dots, m \quad (1)$$

Subject to

$$c_j(x) \le 0, j = 1, 2, \dots, p$$

$$c_k(x) = 0, k = 1, 2, \dots, q$$

$$x_l^l \le x^i \le x_u^l, l = 1, 2, \dots, N$$

where $x = (x_1, x_2, \dots, x_n)^T$ is the solution vector with n decision variables, $f(x)$ denoted m objective functions $f_1, f_2, \cdots, f_m$, $c_j(x)$ are p inequality constraints, and $c_k(x)$ are q equality constraints. In general, it is quite difficult to tackle such problems with simple optimization methods and find the optimal solutions for them in the presence of nonlinearity, multi-modality, high dimensionality, noisy environment, and different nature of objective and constrained functions. Nonlinear optimization methods can be further divided into local and global optimization methods [9–14]. The local search optimization methods conduct their search process from solution to solution going for local changes to get an optimal solution keeping in view the allowed resources. These optimizers offer a local optimal solution with less computational cost as compared to global search optimizers. Newton steepest descent methods and sequential quadratic programming [15], 2-opt local search, Monte Carlo methods and WalkSAT are the most commonly used and popular local search optimization methods [9]. Global optimization methods are heuristic-based methods [16,17]. Among optimization methods, the gradient-based methods are quite expensive in terms of computational cost in solving the real-world problems comprising nondifferentiability, discontinuity, nonlinearity, noisy, flat, multi-dimensionality, or composed of many local optima. In these situations, evolutionary computing techniques are much effective and can provide an optimal solution instead of traditional mathematical programming techniques [18].

[*] *Corresponding author. Email: abhamdi@qu.edu.qa*

Since its inception in the late 1950s, evolutionary computing methods have received significant attention due to large domain applications [19–25]. Until the present, evolutionary algorithms (EAs) under the umbrella of (evolutionary computation [EC]) have successfully and effectively tackled various test suites of benchmark functions and many real-world problems with a non-linear, multi-modal, scalable objective, and constrained functions with complicated structures. Despite having the vast popularity of various nature-inspired and swarm-inspired EAs, still, we need to address their existing shortcomings to make them more advanced and effective by following the basic principles of rules of population evolution, procedures of self-organization and self-adaptation [26]. In general, nature-inspired-based algorithms can be categorized into EAs and swarm intelligence-based algorithms as shown in Figure 1. The EAs including the genetic algorithm (GA) [20,25,27,28], genetic programming (GP) [29], evolutionary strategies (ESs) [30,31], evolutionary programming (EP) [32] are classified as classical paradigms of EC and differential evolution (DE) [33–36] and particle swarm optimization (PSO) [37,38], ant colony optimization [39,40], artificial bee colony (ABC) [41,42], cuckoo search (CS) [43], bat algorithm (BA) [44–46], Bee algorithm [47,48], and firefly algorithm (FA) [49] are new emerging population-based algorithms [50]. Despite many key features of the aforementioned EAs, they demand maximum function evaluations and spent huge computation time to solve optimization problems with complicated search space.

However, it is still difficult and even impossible for a single EA in the existing literature of EC that always performs better on all types of benchmark functions and real-world problems [20,51,52]. In the last several years, the ensemble that multiple search operators have shown significant contribution in solving multi-objective optimization problems with the complicated shape of Pareto front (PF) and large-scale global optimization problems [3,23,53–62]. Inspired from the aforementioned algorithms, this paper also proposes an ensemble strategy-based EA with dynamic resources allocation to handle benchmark functions that are designed recently for the special session of 2017 IEEE congress on evolutionary computing (IEEE-CEC'17) [63]. The suggested algorithm integrates multiple different evolution algorithms by employing the particle swarm optimization (PSO) [37,38,64], DE [36], BA [45] and teaching learning-based optimization (TLBO) [65–68], and GA by sharing valuable information and learn from each other converging toward the known optimal solution of each benchmark function. The proposed ameliorated ensemble algorithm utilizes the

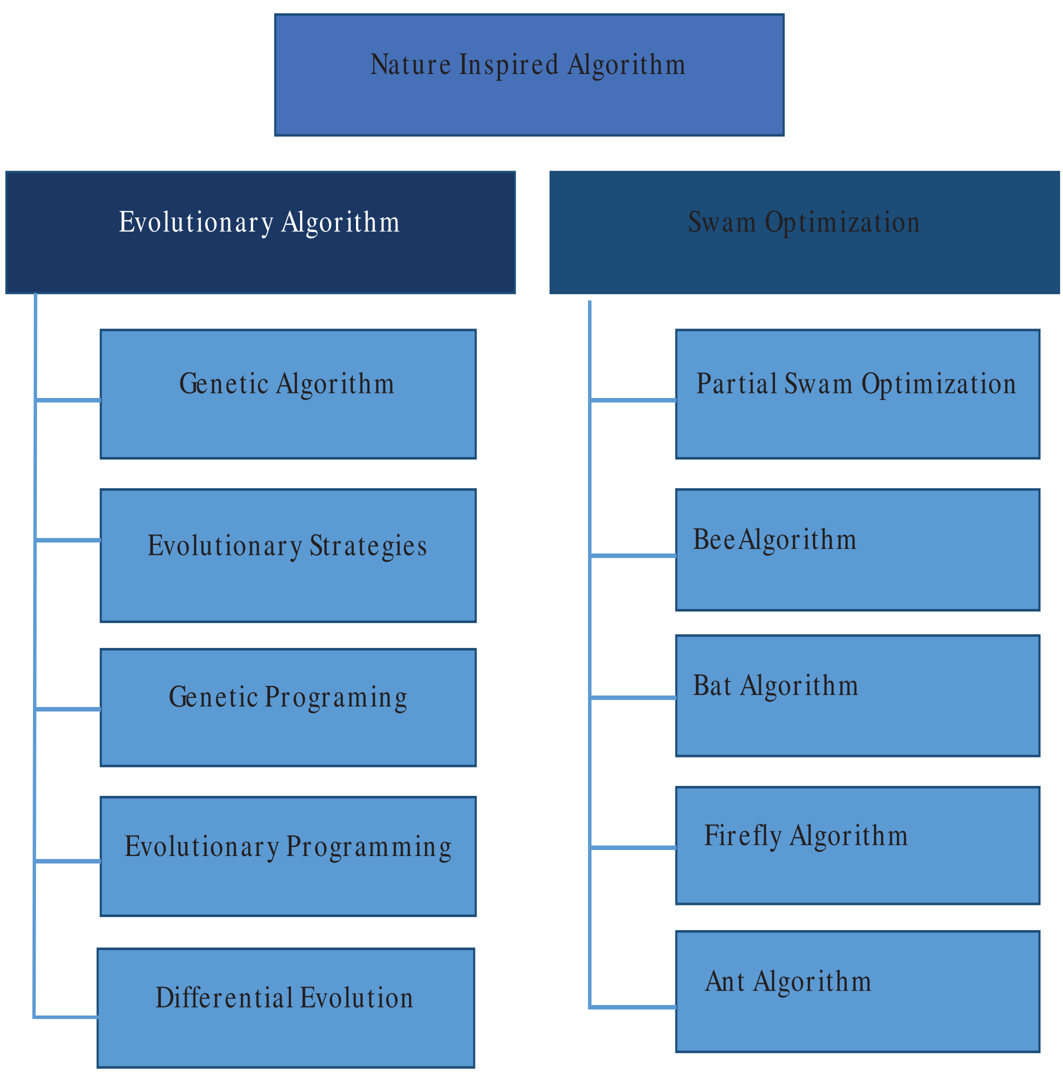

Figure 1 Flow chart for the classification of the nature-inspired algorithm.

strengths of the aforementioned constituent algorithms with novel dynamic resources allocation strategy. As a result, the proposed algorithm is more reliable and computationally efficient during the whole course of population evolution. The proposed algorithm is flexible and can employ any optimization algorithm for its population evolution. To analyze the proposed algorithm, experiments are conducted over CEC'17 benchmark functions [63]. The experimental results of the proposed algorithm are much promising as compared to their competitors in solving most of the large-scale global optimization problems.

The rest of the paper is organized as follows: Section 2 introduces the framework of the proposed ameliorated ensemble strategy-based EA with dynamic resources allocations. Section 3 demonstrates the experimental results and characteristics of the used benchmark functions. In Section 4, we conclude our work.

2. AMELIORATED ENSEMBLE STRATEGY-BASED EA WITH DYNAMIC RESOURCES ALLOCATIONS FOR LARGE-SCALE OPTIMIZATION PROBLEMS

In the recent past two decades, several evolutionary computing methods were developed [8,19,20,61,62,69,70]. They are employing various search operators and strategies for population evolution likewise one-point, two-points, uniform crossover operators [71], trigonometric mutation [72] tournament, ranking, stochastic uniform sampling selection methods, crowding [73], sharing-based niching approaches, adaptive penalty, epsilon superiority of feasible over infeasible solutions, and ensemble constraint handling methods [70,74]. The aforementioned approaches can employ in an ensemble manner the framework of any EAs for performing numerous simulations with the main purpose to approximate the best optimal solution [35]. However, the best achievement of each approach might be associated with the fine-tuning of their corresponding control parameters either statically or additively. Furthermore, the use of different strategies and different parameters might be more appropriate at different stages of evolution and exploration of search spaces. The best setup of various paraments in the EAs can be settled based on trial and error strategies. Recently, ensemble strategies-based EAs have gained much popularity in order to utilize the key features of a diverse set of approaches in the existing literature of evolutionary computing [57,58,62,74–76]. In this paper, we have proposed an ensemble strategy-based EA for solving diverse nature of benchmark functions that are recently designed for the special session of IEEE-CEC'17 [63]. The framework of the suggested ameliorated algorithm is hereby explained in the Algorithm 1. The suggested algorithm denoted by ESEA[1] engages the most popular existing EAs, namely, PSO [37,38,64], DE [36], BA [45] and TLBO [65–68] to perform their search process. The proposed algorithm allocates resources to their constituent algorithms dynamically following the steps from 34 to 38 of the Algorithm 1. As can see in steps 4 to step 5 of Algorithm 1, the constituent algorithms have allocated the same resources. Further, the formula in step 36 of the Algorithm 1 is used to allocate the resources dynamically.

[1] Ensemble Strategy based Evolutionary Algorithm.

Algorithm 1: The Framework of the Ameliorated Ensemble Strategy-Based Evolutionary Algorithm

1: $x = [x^1, x^2, \dots, x^N] = x_l + (x_u - x_l) * rand(N, n)$ % Generate initial population of size N uniformly and randomly.
2: Evaluate function values of N solutions, $x = [x^1, x^2, \dots, x^N]$, $[f(x^1), f(x^2), \dots, f(x^N)]$.
3: Initially, divide the population set of size N into k sub-population sets.
4: $P^k = \{P^{A1}, P^{A2}, P^{A3}, P^{A4}\}$, where $P^k = |N \times p_a^k|, p_a^k = 0.25$
5: $g = 1$; % Evolutionary process of the hybrid swarm intelligence technique.
6: **for** $g = 1 : MG$ **do**
7: **if** $i \in P^{A1}$ **then**
8: $v^i = \omega \times v^i + a_1 \times r_1 \times (x^{pb} - x^i) + a_2 \times r_2 \times (x^{gb} - x^i)$
9: $y^i = x^i + v^i, i = 1, 2, \dots, N.$
10: **end if**
11: **if** $i \in P^{A2}$ **then**
12: $y^i = x^i + F \times (x^{r_1} - x^{r_2})$, where $i \neq r_1 \neq r_2$
13: **end if**
14: **if** $i \in P^{A3}$ **then**
15: $f^i = f^{min} + (f^{max} - f^{min}) \times \beta, \beta \in [0, 1]$
16: $v^i = v^i + (x^i - x^{gb}) \times f^i$
17: $x^i = x^i + v^i$
18: $A^i = \alpha \times A^i$
19: $y^i = x^i + \zeta \times A^i, \zeta \in [-1, 1].$
20: **end if**
21: **if** $i \in P^{A4}$ **then**
22: Generate a random r^i in [0, 1];
23: **end if**
24: **if** $r^i \leq \lambda^i$ referred to Eq. (2). **then**
25: Update learner x^i based on Eqs. (3–5).
26: **end if**
27: Compute the objective function values of new set of population.
28: **if** $f(z^i) < f(x^i)$ **then**
29: $x^i = z^i$
30: **else**
31: $x^i = x^i$
32: **end if**
33: Compute the success rate r^k of each k^{th} algorithm.
34: **for** $k = 1 : 4$ **do**
35: $p_a^k = \varsigma \times p_a^k + (1 - \varsigma) \times \frac{r^k}{\sum r^k}$
36: Update the set of sub-population by using formula: $\| P^k \| = N \times p_a^k|$.
37: **end for**
38: $g = g + 1$
39: **end for**

In the proposed algorithm, TLBO has been used as a constituent algorithm that employs a group of learners whose fitness level is evaluated upon their results or grades. The grade of each learner is then updated based on their learning attained from the teacher and their interaction with other learners. Two phases are involved in TLBO operation including the teacher phase and learner phase. The learning enthusiasm values of learners as described in [77] as given under:

$$\lambda^i = \lambda^l + [\lambda^u - \lambda^l] \times \frac{N - i}{N}, \tag{2}$$

where $\lambda^u = 1$ is the maximum learning enthusiasm and the minimum learning enthusiasm values denoted by λ^l belong to $[0.1, 0.5]$. If i^{th} leaner get knowledge from the teacher then its position will update as follows:

$$x^m = \frac{1}{N}\sum_{i=1}^{N} x^i, \quad T_F = \lfloor (1 + rand) \rfloor; \tag{3}$$

$$y^i = \begin{cases} x^i + rand \times (x^b - T_F \times x^m) \text{ if } rand < 0.5 \\ x^i + F \times (x^{r_1} - x^{r_2}), \text{ where } i \neq r_1 \neq r_2 \end{cases} \tag{4}$$

The polynomial mutation is described as under:

$$z^i = \begin{cases} y^i + \sigma_i \times (x_u - x_l) \text{ with probability } p_m \\ y^i \qquad\qquad \text{with probability } 1 - p_m \end{cases} \tag{5}$$

where x_l is the lower and x_u is the upper bound over i^{th} decision variable, N is the size of population set, n is the dimension of the search space or a number of decision variables, M_g is Maximum generations, $\omega \in [0.8, 1.2]$ is the inertia weight parameter used in the framework of the PSO as can see in the Algorithm 2, f is the frequency used by the Bat while seeking its prey, kNnumber of constituent algorithms, v^i represents the velocity, g indicates the in the current generation, β is the random vector, x^{pb} indicates the personal best particle or individual, x^{gb} indicates the global best individual, x^i is the location of the i^{th} Bat in the solution space, r^i is the pulse emission rate, A^g represents the average loudness, F is the scaling factor that controls the difference of population used in the framework of the DE, CR is the probability of crossover used in DE to create offspring population.

Algorithm 2: The Framework of the Particle Swarm Optimization

$[x^1, x^2, \dots, x^N] = x_l + (x_u - x_l) \times rand(N, n)$ % Generate the initial population of size N uniformly and randomly in the search space Ω.Furhermore,
Compute the objective function values, $[f(x^1), f(x^2), \dots, f(x^N)]$.
$g = 1$
while $g \leq M_g$ **do**
  $v_i^g = \omega \times v_i^{g-1} + a_1 \times r_1 \times (x_{pb} - x_i^g) + a_2 \times r_2 \times (x_{gb} - x_i^g)$
  $y_i^g = x_i^t + v_i^g, i = 1, 2, \dots, N.$
  Compute the objective function values of new population set of solutions, $[f(y^1), f(y^2), \dots, f(y^N)]$.
  if $f(y^i) < f(x^i)$ **then**
    $x^i = y^i$
  else
    $x^i = x^i$
  end if
  $g = g + 1$
end while

2.1. Differential Evolution

DE is one of the most popular EAs that was first proposed by Rainer Storn and Kenneth Price for solving the Chebychev polynomial fitting problems [36]. DE uses the idea difference to perturb their population. DE is similar to other existing generates its population uniformly and randomly. Then, it applies the fundamental evolutionary operators include "mutation, crossover, and selection" to perform its search process in the whole course of the optimization process. The parameters involved in the framework DE including N (population size), F (mutation factor), and CR (crossover ratio) and can settle with different procedures [35,59,69,78–83]. Crossover is the main source of exploration, the mutation is used for exploitation purposes, and selection operators imposed selection pressure for survival of fittest individuals in order to evolve the population.

In DE [35,36,84], v^i is a mutant vector for each i^{th} individual of current solutions size N by employing the following mutation strategies:

1. **DE/rand/1:**

$$v^i = x^i + F \times (x^{r_1} - x^{r_2}), i \neq r_1 \neq r_2 \tag{6}$$

2. **DE/best/1:**

$$v^i = x^b + F \times (x^{r_1} - x^{r_2}) \tag{7}$$

3. **DE/rand-to-best/1:**

$$v^i = x^i + F \times (x^{r_1} - x^{r_2}) + F \times (x^b - x^{r^i}) \tag{8}$$

4. **DE/current-to-best/1:**

$$v^i = x^i + F \times (x^{r_1} - x^{r_2}) + F \times (x^b - x^i) \tag{9}$$

where $x^{r_1} - x^{r_2}$ is the difference of two different randomly chosen solutions, x^i is the current i^{th} solution, x^b is the best individual of the in current generation of population evolution, and F is the scaling factor in the interval $[0, 2]$ that controls the difference of variation in populations. Mutation plays a vital role in exploitation that is applied in combination with crossover in the framework of the DE algorithm [83,85].

2.1.1. Crossover operation

A crossover is a procedure to exchange information among the previous and current population to produce a new population. It is normally applied for the exploration of different valuable regions of search space aiming at to maintain diversity among the population for further evolution. In DE, the crossover operator applied on each i^{th} mutant vector denoted by u^i in combination with i^{th} solution of the previous population.

$$u^i = \begin{cases} v^i, & \text{if } rand \leq CR \parallel j = j^{rand}; \\ x^i, & \text{Otherwise;} \end{cases} \tag{10}$$

where $rand$ is an arbitrary random number and $CR \in [0, 1]$ is the crossover probability.

2.1.2. Selection operation

The selection operation is conducted among parent solutions x^i and trial solutions u^i on their objective function values as under:

$$x^{i,g+1} = \begin{cases} u^{(i,g)}, & \text{if } f(u^{(i,g)}) < f(x^{(i,g)}); \\ x^{(i,g)}, & \text{otherwise;} \end{cases} \tag{11}$$

where $x^{i,g+1}$ is the set of solutions for the next generation of the population. DE algorithm employs mutation strategies to pr provide the best optimal solutions in order to avoid the occurrence of premature convergence during its search process of population evolution.

2.2. Particle Swam Optimization

PSO algorithm was first proposed by James Kennedy and Russell Eberhart [38]. PSO was mainly inspired by the social behaviors and interaction of swarm as encountered in nature likewise animal herds, bird flocking, and schooling of fish. It uses a swarm of insects while performing the search process and each member of the group is called particle [86]. Due to its simple search mechanism, computational efficiency, and easy implementation, PSO has successfully tackled many optimization problems. Each Particle of PSO utilizes their personal best experience and the global experience of their neighbors to adaptively adjust its velocity v and position vector x in order to explore and exploit the search space of the problem at hand during population evolution. Moreover, each particle has a memory, remembering the best position of the search space it has ever visited. Thus, its movement is an aggregated acceleration towards its best previously visited position and towards the best particles of a topological neighborhood [87,88]. The algorithmic framework of PSO is hereby outlined in Algorithm 2.

where N is the size of the swarm, n is the dimension of the search Space, v^i represents the velocity, ω is inertia weight parameter x^{pb} is the personal best and x^{gb} is the global best of the swarm, M_g denotes maximum generations allowed for evolving the initial set of solutions or particles as generated uniformly and randomly within bounds of search space as outlined in the Algorithm 2.

3. BENCHMARK FUNCTIONS AND EXPERIMENTAL RESULTS

In this research work, we assess the performance of the proposed ameliorated ensemble strategy-based EA with dynamic resources allocations by using 29 benchmark functions recently designed for the special session of IEEE-CEC. As like other existing benchmark functions, they are playing contemporary trends in design and assessment of novel search algorithms. The characteristics of the used benchmark functions are hereby explained in Table 1.

3.1. Computing Platform and Experimental Settings

All experiments were performed on a computer with Intel Core $i5-$6200 CPU, 2.40GHz and 4G RAM, under Windows 10 pro, 64 bit OS. The proposed algorithm and all other existing algorithms used in the comparative analysis were implemented in MATLAB R2017b programming environment. To obtain the quantity of variability, all benchmark functions solving with randomized algorithms need multiple executions to solve at hand problems and approximate their solutions with reasonable statistics. The multiple runs of simulations regarding the stochastic nature algorithms are advised in the existing literature of EC from 10 to 50. In this paper, experimental results are gathered by executing all algorithms including (a) PSO, (b) DE algorithm, (c) proposed ameliorated ensemble strategy-based EA with 10 independent runs of executions with different random seeds. The Matlab command $rand(state, sum(100 * clock))$ has been used in the carried out experiments.

3.2. Parameters Settings

In our carried out experiment, we have settled different parameters as the lower bound $x_l = -100$, upper bound $x_u = 100$ for the used benchmark functions, $N = 100$ is the size of initially generated set of solutions, $n = 10, 30, 50$ are the different dimensions of the search space, $M_t = n \times 1000$ are the maximum number of function evaluations, r_1 and r_2 are the two uniformly distributed random numbers, v^t is the current velocity of the particle, x^t is the current position of the particle, $\omega = 0.729$ is the inertia factor, where $\omega \in [0.8, 1.2]$, a_1 and a_2 are the two acceleration constants or acceleration coefficients that usually lie between 1 and 4, however, we have used $a_1 = a_2 = 1.7$ and $\alpha = \gamma = 0.9$. The loudness of the bats decreases due to an increase in their pulse rate as they get closer to their prey, like $A_i^{t-1} \to 0$, $r_i^t \to r_i^0$ as $t \to \infty$. The initial loudness is $A_0 \in [1, 2]$ and the initial emission rate is $r_0 \in [0, 1]$. In our experiments, the parameters used in the 14-23-steps of the Algorithm 1 were settled according to $r_0 = 0.1$, $A_0 = 0.9$, $f_{min} = 0$, $f_{max} = 2$ and $\alpha = \gamma = 0.9$ for the purpose to examine the algorithmic behavior of the proposed ensemble-based EA. The value of $\varsigma = 0.02$ were settled in Algorithm 1 make active and switch on the constituent algorithm based on their previous performance in the case of their worst situation during the whole course of optimization process.

3.3. Characteristics of Benchmark Functions and Discussion Over Experimental Results

Benchmark functions might determine strict rules on the termination conditions to execute the suggested algorithm and establish a fair comparison against some existing algorithms. The same budget of function evaluations, $N \times n$ were allocated to the suggested algorithm Ensemble Strategy Based Evolutionary Algorithm (ESEA) to establish comparison against the existing algorithms likewise the BA and PSO to be carried out our experiment by using benchmark functions designed for the special issue of IEEE-CEC'17 [63].

In Table 1, *UF* stand for unimodal functions, *MF* denotes the multimodal functions, *CF* refer to Composition Functions, *SRF* abbreviated for shifted rotated functions, and *HF* represents hybrid functions. Benchmark functions, f_1 - f_3 are unimodal functions, f_4 - f_{10} are simple multi-modal functions, f_{11} - f_{20} are hybrid functions and f_{21} - f_{30} are composition functions.

Tables 2 and 3 clearly exhibit that the proposed ESEA algorithm has performed much better than PSO and DE, and it has tackled each benchmark functions more effectively, especially, F_2, F_4, F_5, F_8, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{22}, F_{23}, F_{25}, and F_{29} keeping their minimum (best) objective function values. The constituent algorithm, namely, PSO has performed better than DE in terms of minimum function values for the problems like F_1, F_7, F_{13}, and F_{26}. On the other hand, DE has found better results than PSO by solving benchmark functions F_9 and F_{27} in terms of best objective function values. Similarly, PSO and ESEA have produced good results in terms of minimum (best) objective function values for dealing with F_21, F_{24}, and F_{28}.The numerical results for the problems refer to F_6 and F_{20} are almost same. The average objective function values approximated by the proposed ESEA are much better than PSO and

Table 1 Features of the 2017 IEEE congress on evolutionary computing (IEEE-CEC'17) benchmark functions.

	No	Objective Function	Optimal Solution
UF	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function*	200
	3	Shifted and Rotated Zakharov Function	300
SMF	4	Shifted and Rotated Rosenbrocks Function	400
	5	Shifted and Rotated Rastrigins Function	500
	6	Shifted and Rotated Expanded Scaffers F6 Function	600
	7	Shifted and Rotated Lunacek BiRastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigins Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefels Function	1000
HF	11	Hybrid Function 1 ($N = 3$)	1100
	12	Hybrid Function 2 ($N = 3$)	1200
	13	Hybrid Function 3 ($N = 3$)	1300
	14	Hybrid Function 4 ($N = 4$)	1400
	15	Hybrid Function 5 ($N = 4$)	1500
	16	Hybrid Function 6 ($N = 4$)	1600
	17	Hybrid Function 6 ($N = 5$)	1700
	18	Hybrid Function 6 ($N = 5$)	1800
	19	Hybrid Function 6 ($N = 5$)	1900
	20	Hybrid Function 6 ($N = 6$)	2000
CF	21	Composition Function 1 ($N = 3$)	2100
	22	Composition Function 2 ($N = 3$)	2200
	23	Composition Function 3 ($N = 4$)	2300
	24	Composition Function 4 ($N = 4$)	2400
	25	Composition Function 5 ($N = 5$)	2500
	26	Composition Function 6 ($N = 5$)	2600
	27	Composition Function 7 ($N = 6$)	2700
	28	Composition Function 8 ($N = 6$)	2800
	29	Composition Function 9 ($N = 3$)	2900
	30	Composition Function 10 ($N = 3$)	3000

Table 2 Numerical results supplied by our proposed ESEA *versus* DE and PSO by solving F_1 to F_{15} of CEC 2017 in ten dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
		DE	154.696	5369.28	1253.6	1068.1	0.285587
F01	100	PSO	100.4	12740.8	2081.43	2590.86	0.162900
		ESEA	100.431	12740.8	1969.4	2242.69	0.331035
		DE	15392.6	72912.9	40244.8	13288.6	0.312764
F02	200	PSO	14938.1	69107.2	36731.5	11503.8	0.191857
		ESEA	200	200	200	0	0.358209
		DE	701.114	3571.32	1851.96	594.792	0.288490
F03	300	PSO	300	300	300	$1.17436e-09$	0.167251
		ESEA	300	300	300	$1.13687e-14$	0.330349
		DE	404.989	406.559	406.069	0.331792	0.288698
F04	400	PSO	400.021	467.83	405.114	9.22678	0.165401
		ESEA	400.006	467.083	404.556	9.06866	0.332131
		DE	506.096	512.537	509.484	1.64061	0.350939
F05	500	PSO	502.985	540.793	518.788	9.31158	0.222609
		ESEA	500.995	513.93	505.961	2.78701	0.391999
		DE	600	600	600	$4.54747e-14$	0.542958
F06	600	PSO	600	600	600	$7.41501e-06$	0.369663
		ESEA	600	600	600	$1.98997e-07$	0.560691
		DE	715.605	724.133	720.49	2.03324	0.362580
F07	700	PSO	704.975	736.155	721.538	5.22767	0.238948
		ESEA	712.009	725.145	718.931	3.26071	0.413509
		DE	806.246	814.161	809.799	1.59694	0.355887
F08	800	PSO	804.975	833.829	814.205	6.36076	0.229105
		ESEA	800.995	811.939	805.336	2.27276	0.401266
		DE	900	900	900	0	0.373708
F09	900	PSO	900	900.454	900.019	0.0708882	0.244400
		ESEA	900	900.544	900.014	0.0776607	0.420489
		DE	1222.4	1701.28	1473.43	101.522	0.403186

(*Continued*)

Table 2 Numerical results supplied by our proposed ESEA *versus* DE and PSO by solving F_1 to F_{15} of CEC 2017 in ten dimension. *(Continued)*

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F10	1000	PSO	1130.13	1923.48	1557.9	202.483	0.262548
		ESEA	1000.31	1770.37	1228.97	166.742	0.445938
		DE	1101.21	1103.9	1102.88	0.643498	0.320476
F11	1100	PSO	1101.07	1138.52	1111.44	8.74652	0.196038
		ESEA	1100.1	1110.7	1104.11	2.68184	0.366345
		DE	27912.1	476347	187129	106891	0.336839
F12	1200	PSO	1817.37	43785.9	13053	10138.9	0.212372
		ESEA	1720.46	40038.7	10339.4	8455.91	0.431194
		DE	1316.36	5738.27	2209.94	990.152	0.332796
F13	1300	PSO	1311.97	26450.3	8829.74	7596.47	0.212558
		ESEA	1373.23	20484.6	7064.61	5085.88	0.362542
		DE	1403.23	1446.72	1417.69	9.73851	0.365937
F14	1400	PSO	1404.68	1502.45	1451.65	21.2507	0.247850
		ESEA	1400.22	1426.94	1405.83	5.16453	0.409165
		DE	1504.54	1593.93	1523.52	22.0173	0.302490
F15	1500	PSO	1501.07	2791.9	1596.91	190.258	0.179000
		ESEA	1500.14	1536	1504.18	5.0723	0.357709

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

Table 3 Numerical results furnished by our proposed ESEA *versus* DE and PSO by solving F_{16} to F_{30} of CEC 2017 in ten dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F16	1600	PSO	1600.09	2019.55	1800.7	139.618	0.207083
		ESEA	1600.08	1977.27	1702.79	105.03	0.382809
		DE	1700.43	1703.35	1701.39	0.548401	0.494150
F17	1700	PSO	1701.62	1845.67	1739.97	28.1929	0.357664
		ESEA	1700	1760.18	1708.53	15.0084	0.538816
		DE	1894.49	3808.99	2558.39	418.242	0.327526
F18	1800	PSO	1830.64	34654.8	7531.12	8418.07	0.205330
		ESEA	1802.78	16117.8	3995.87	3301.19	0.385206
		DE	1900.41	2096.6	1926.76	38.9099	1.320215
F19	1900	PSO	1902.71	4210.75	2057.89	368.142	1.194307
		ESEA	1900.05	1946.63	1906.38	8.66452	1.377728
		DE	2000	2000.31	2000.01	0.043713	0.500983
F20	2000	PSO	2000	2153.44	2046.57	54.9448	0.361641
		ESEA	2000	2021.31	2001.16	3.03203	0.524198
		DE	2215.98	2317.74	2260.46	35.0718	0.508035
F21	2100	PSO	2200	2344.38	2294.99	50.4192	0.372101
		ESEA	2200	2319.15	2252.45	54.6317	0.549255
		DE	2255.8	2301.61	2297.64	9.93793	0.606281
F22	2200	PSO	2212.85	2314.19	2299.8	19.497	0.487455
		ESEA	2200	2302.72	2293.06	24.8818	0.647195
		DE	2607.86	2615.29	2611.89	2.00407	0.667432
F23	2300	PSO	2610.2	2693.28	2632.82	15.2394	0.512993
		ESEA	2606.04	2635.57	2616.15	7.33072	0.691292
		DE	2608.29	2748.51	2709.12	40.8449	0.684418
F24	2400	PSO	2500	2824.9	2732.41	87.5749	0.555982
		ESEA	2500	2772.92	2712.52	85.4086	0.732236
		DE	2899.24	2944.34	2912.07	10.3212	0.571223
F25	2500	PSO	2897.76	2948.33	2920.83	23.3652	0.454348
		ESEA	2897.74	2949.51	2924.8	22.9795	0.610618
		DE	2779.72	2951.62	2915.09	28.4165	0.747483
F26	2600	PSO	2600	3170.52	2917.18	87.984	0.606484
		ESEA	2800	3061.65	2904.95	49.2865	0.782874
		DE	3089.09	3090.37	3089.61	0.250117	0.777527
F27	2700	PSO	3088.58	3199.97	3117.61	29.7043	0.644963
		ESEA	3089.31	3187.38	3101.84	14.1845	0.824549
		DE	3172.27	3402.69	3230.32	57.2878	0.695014
F28	2800	PSO	3100	3476.94	3296.04	147.356	0.559124
		ESEA	3100	3446.48	3241.6	148.445	0.748309
		DE	3149.6	3178.83	3163.24	6.34192	0.669195
F29	2900	PSO	3141.99	3322.68	3204.98	44.0286	0.546565
		ESEA	3136.75	3207.31	3170.62	16.614	0.718650

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

DE as listed in the second column of Tables 2 and 3. The standard deviation statistics of the proposed ESEA algorithm are also much better than PSO and DE algorithm in solving most of the benchmark functions. The maximum values of the ESEA regarding each benchmark function, especially, F_2, F_3, F_4, F_5, F_7, F_8, F_{10}, F_{12}, F_{15}, F_{16}, F_{19}, F_{22}, and F_{26} are quite reasonable as compared to the competitor. Based on these experimental results, one can easily judge that the proposed algorithm has obtained much promising experimental results by solving almost all benchmark functions efficiently in ten dimension [63].

Tables 4 and 5 summarizes the numerical results incurred by our proposed ESEA algorithm while solving each benchmark function of the IEEE-CEC'17 test suite [63] with thirty decision variables. The first column of Tables 4 and 5 present the best objective function values approximated by ESEA algorithm in comparison with PSO and DE Algorithm with the same parameter settings and resources allocations. The benchmark functions used in our carried out experiments, namely, F_1, F_5, F_6, F_8, F_9, F_{10}, F_{13}, F_{14}, F_{19}, F_{20}, F_{21}, F_{23}, F_{24}, F_{25}, and F_{29} are efficiently tackled by ESEA in their ten independent runs of simulations to establish a fair comparison among all used algorithms. As compared to the DE algorithm, the experimental results of the PSO better in solving benchmark functions, namely, F_5, F_7, F_8, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{18}, F_{19}, F_{20} F_{21}, F_{24}, F_{25}, F_{26}, F_{28}, and F_{29}. Likewise on the benchmark functions, namely, F_2, F_9, F_{17}, F_{23}, and F_{27}, DE has offered comparatively better experimental results against PSO. Similarly, compared to DE, PSO has solved the benchmark functions, F_{26} and F_{28} more efficiently and effectively. As recorded in the second column of Tables 4 and 5, the average function values of the PSO, DE are less effective as compared to the proposed ESEA algorithm by solving each problem in thirty dimensions. The third column of the same tables gathers the evolution in standard deviation concerning PSO, DE, and ESEA algorithms for each IEEE-CEC'17 benchmark function [63]. The maximum function values provided by ESEA for

Table 4 Numerical results provided by our proposed ESEA *versus* DE and PSO by solving F_1 to F_{15} of CEC 2017 in thirty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
		DE	2343.5	15217.6	7603.63	3358.95	0.583198
F01	100	PSO	116.029	$8.45767e+09$	$1.56863e+09$	$1.88461e+09$	0.429356
		ESEA	100.11	20941.8	3777.7	5219.27	0.623945
		DE	106830	282080	201559	37131.8	0.510951
F02	200	PSO	130585	340982	207974	43592.4	0.413869
		ESEA	200	200	200	$5.68434e-15$	0.664858
		DE	84987.5	161136	124259	17148.4	0.590673
F03	300	PSO	5207.41	15630.3	9995.55	2096.24	0.555953
		ESEA	5932.04	19834.2	10564.5	3265.49	0.662882
		DE	489.974	510.857	493.62	3.66188	0.590970
F04	400	PSO	411.475	1600.76	638.146	265.737	0.522777
		ESEA	403.997	537.632	478.475	26.9481	0.664474
		DE	632.141	679.413	657.292	10.7419	0.775978
F05	500	PSO	558.702	707.944	634.038	30.4231	0.671594
		ESEA	546.763	687.315	610.744	37.8276	0.847156
		DE	600	600	600	$5.29326e-05$	1.360746
F06	600	PSO	603.945	642.926	620.009	10.4469	1.246342
		ESEA	600	603.85	600.287	0.722317	1.407526
		DE	873.015	910.584	894.079	8.89461	0.793153
F07	700	PSO	791.393	1077.12	900.225	53.0693	0.693528
		ESEA	853.577	922.68	890.75	16.2467	0.863758
		DE	937.397	975.232	958.309	9.6358	0.803981
F08	800	PSO	868.652	997.996	920.6	26.8934	0.699017
		ESEA	841.788	963.422	900.857	31.5793	0.870075
		DE	935.076	1129.4	989.141	43.3545	0.799237
F09	900	PSO	1040.56	6864.33	2406.06	1479.61	0.697380
		ESEA	900	908.634	901.659	1.81279	0.871768
		DE	6361.91	7425.29	6983.52	219.832	0.945862
F10	1000	PSO	2897.84	6379.85	4593.4	706.574	0.811053
		ESEA	4592.78	7524.32	6402.58	680.21	1.002238
		DE	1258.45	1411.07	1335.13	28.9163	0.677599
F11	1100	PSO	1165.22	1361.83	1238.11	47.4266	0.551924
		ESEA	1125.73	1293.3	1198.22	41.2607	0.712712
		DE	$7.13614e+06$	$2.37832e+07$	$1.43974e+07$	$3.84999e+06$	0.793601
F12	1200	PSO	236450	$1.98698e+08$	$1.04516e+07$	$3.31395e+07$	0.638339
		ESEA	35979.1	$1.00722e+07$	481964	$1.39824e+06$	0.807422
		DE	79082.1	$1.70393e+06$	883763	429893	0.710579
F13	1300	PSO	1740.94	$4.44085e+06$	102730	619752	0.562704
		ESEA	1590.3	62329.4	13706.6	12446.7	0.727566
		DE	14076.6	398348	119518	71838.1	0.923934
F14	1400	PSO	2279.49	139668	32435	37598.6	0.782475
		ESEA	1761.1	72486.6	24322	19041.1	0.963656
		DE	39715	324880	131667	69389.6	0.653166
F15	1500	PSO	1704.74	42993.9	8694.26	8521.17	0.525389
		ESEA	1576.21	42915.9	9495.27	9333.96	0.691918

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

Table 5 | Numerical results gathered by our proposed ESEA *versus* DE and PSO by solving F_{16} to F_{30} of CEC 2017 in thirty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
		DE	2183.79	2918.36	2530.69	161.685	0.782739
F16	1600	PSO	2106.02	3562.08	2725.97	300.672	0.642358
		ESEA	1743	2959.96	2388.39	242.962	0.831094
		DE	1854.94	2119.54	1964.43	58.8926	1.297625
F17	1700	PSO	1890.39	2605.69	2188.25	179.483	1.086634
		ESEA	1760.47	2490.67	1971.61	166.249	1.285133
		DE	260549	$2.28682e+06$	$1.02937e+06$	461361	0.731590
F18	1800	PSO	42585.3	$4.0285e+06$	487282	669296	0.608598
		ESEA	40290.4	$1.08407e+06$	297521	238943	0.794658
		DE	26559.5	277855	120705	58586.8	3.735985
F19	1900	PSO	2111.67	56536.2	10839.5	10558.7	3.572121
		ESEA	1949.69	55269.5	11044.8	11730.9	3.763173
		DE	2143.53	2463.83	2286.22	71.8191	1.331557
F20	2000	PSO	2182.79	2888.39	2415.88	160.795	1.161222
		ESEA	2048.54	2482.94	2220.46	96.0979	1.368786
		DE	2434.78	2480.64	2461.53	8.85573	1.834007
F21	2100	PSO	2352.85	2539.18	2439.07	36.1701	1.430537
		ESEA	2346.9	2447.8	2394.96	25.8825	1.621051
		DE	3178.16	5888.01	4022.64	562.158	1.833734
F22	2200	PSO	2300	7554.14	4478.78	2111.1	1.658826
		ESEA	2300	5894.4	2371.38	503.191	1.844883
		DE	2780.52	2833.63	2807.88	11.3247	2.032975
F23	2300	PSO	2806.36	3156.27	2995.46	83.7882	1.884781
		ESEA	2729.95	2891.11	2795.12	40.9749	2.071151
		DE	2979.53	3035.24	3011.75	11.351	2.300292
F24	2400	PSO	2935.24	3581.27	3189.14	114.375	2.128731
		ESEA	2875.06	3098.07	3000.81	51.8501	2.371857
		DE	2887.48	2888.19	2887.84	0.179793	1.940677
F25	2500	PSO	2884.38	3280.15	2931.32	76.9693	1.800011
		ESEA	2883.44	2950.29	2896.84	17.9471	1.989600
		DE	4994.93	5412.17	5223.82	100.247	2.517564
F26	2600	PSO	2802.32	7569.71	5621.88	1541.49	2.342270
		ESEA	2800	6597.46	4454.28	1165.03	2.522929
		DE	3211.92	3223.21	3218.38	2.66827	2.824495
F27	2700	PSO	3218.52	3479.01	3343.57	65.1078	2.669393
		ESEA	3219.65	3357.48	3279.43	35.2231	2.861232
		DE	3236.32	3291.99	3262.33	13.3343	2.421336
F28	2800	PSO	3205.88	4174.21	3352.32	193.343	2.271840
		ESEA	3100.02	3257.96	3206.72	37.8344	2.460567
		DE	3675.48	4081.97	3877.38	87.2055	1.974353
F29	2900	PSO	3482.47	4572.83	3927.22	243.893	1.785192
		ESEA	3316.6	4007.26	3620.86	152.786	1.978978

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

each problem are much competitive as compared to PSO and DE Algorithm, especially in the case of F_1, F_2, F_4, F_5, F_7, F_8, F_{11}, F_{12}, F_{13}, F_{14}, F_{16}, F_{18}, F_{20}, F_{21}, F_{22}, F_{23}, F_{24}, F_{26}, F_{28}, and F_{29}. On the other hand, PSO has got better results in terms of maximum function values for the problem, F_3, F_5, F_8, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{18} , F_{19}, and F_{21} as compared to DE algorithm. However, the maximum function values of DE algorithm for the functions, F_1, F_2, F_4, F_6, F_7, F_9, F_{16}, F_{17}, F_{20}, F_{22}, F_{23}, F_{24}, F_{25}, F_{26}, F_{27}, F_{28}, and F_{29} are comparatively better than PSO during ten independent runs of population evolution with different random seeds.

Tables 6 and 7 offer the numerical results accumulated by the proposed ESEA algorithm in terms of the minimum function values, average function values, standard deviation values, and maximum function values by solving each benchmark functions with fifty decision variables [63]. ESEA has efficiently tackled almost all test problems, especially, F_1, F_2, F_3, F_4, F_6, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{20}, F_{21}, F_{22}, F_{23}, F_{24} , F_{25}, F_{26}, F_{28}, and F_{29} as compared to their counterparts. As a constituent algorithm of the proposed algorithm, DE has performed better than PSO and found promising experimental results for the benchmark functions of the IEEE-CEC'17 [63], namely, F_3, F_5, F_7, F_8, F_{10}, F_{11}, F_{12}, F_{13}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, F_{21}, F_{22}, F_{23}, F_{26}, F_{27}, and F_{29}. The average function values of the ESEA for the benchmark functions F_2, F_4, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, and F_{22} and also standard deviation values for the same functions are much better than PSO and DE. As for as the maximum values are concerned, PSO for problems including F_3, F_5, F_8, F_{10}, F_{11}, F_{14}, F_{15}, F_{16}, F_{18}, F_{19}, F_{20}, F_{21}, and F_{22} is better than DE algorithm for the same benchmark functions.

Tables 8 and 9 provides the numerical results of the proposed algorithm *versus* GA by solving each benchmark function in ten dimensions. Tables 10 and 11 includes the numerical results of each benchmark function with thirty decision variables. The numerical results obtained by proposed ESEA *versus* GA by solving F_1 to F_{29} benchmark functions in fifty dimensions are hereby summarized in Tables 12 and 13. All results gathered in these tables were obtained

Table 6 Numerical results accumulated by our proposed ESEA *versus* DE and PSO by solving F_1 to F_{15} of CEC 2017 in fifty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
		DE	36046.6	$2.0793e+07$	$3.72693e+06$	$4.4923e+06$	1.637273
F01	100	PSO	970452	$4.09775e+10$	$1.59622e+10$	$8.95246e+09$	1.426690
		ESEA	103.918	$1.34257e+08$	$2.63524e+06$	$1.87993e+07$	1.809686
		DE	198609	464021	337500	57817.6	1.180289
F02	200	PSO	229215	527323	352243	54964.4	0.843248
		ESEA	200	200.369	200.011	0.0544288	1.260103
		DE	201456	325286	285716	26406.1	1.728365
F03	300	PSO	38033.5	112860	69877.4	18196.3	1.467334
		ESEA	35595.3	119072	70252.6	17748.3	1.805110
		DE	569.263	656.565	628.633	19.3177	1.734794
F04	400	PSO	571.555	6421.59	2285.39	1446.39	1.441616
		ESEA	428.758	628.568	536.584	56.7989	1.812784
		DE	841.223	914.623	877.926	16.2365	2.195264
F05	500	PSO	617.221	899.984	791.344	54.5483	1.945801
		ESEA	631.334	927.262	818.512	76.1552	2.266826
		DE	600.056	600.087	600.071	0.00673945	3.672042
F06	600	PSO	619.56	665.687	639.984	7.84557	3.051529
		ESEA	600	602.888	600.212	0.500449	2.496372
		DE	1097.93	1163.1	1135.75	14.3139	1.472860
F07	700	PSO	976.723	1508.27	1193.14	109.155	1.300345
		ESEA	1091.64	1166.59	1132.51	16.13	1.555495
		DE	1125.68	1203.9	1177.14	15.9981	1.485604
F08	800	PSO	996.027	1234.89	1102.58	56.4642	1.314049
		ESEA	947.253	1227.19	1107.12	73.4285	1.525953
		DE	2402.16	4456.02	3213.54	480.236	1.552365
F09	900	PSO	3083.85	27022.8	13967.2	7487.37	1.378557
		ESEA	901.528	3899.49	1082.77	442.689	1.594404
		DE	12321.8	13773.9	13236.2	374.854	1.708856
F10	1000	PSO	5687.71	9511.81	7415.06	881.436	1.501783
		ESEA	10757.5	13691.7	12485.3	630.207	1.741799
		DE	1910.13	2904.34	2338.26	251.749	1.271392
F11	1100	PSO	1235.06	1571.44	1362.03	68.9334	1.104600
		ESEA	1188.17	1398.72	1263.8	44.3493	1.313783
		DE	$1.26926e+08$	$3.80466e+08$	$2.28579e+08$	$5.69729e+07$	1.480879
F12	1200	PSO	$1.40256e+06$	$1.45961e+10$	$1.37373e+09$	$2.58693e+09$	1.283814
		ESEA	462507	$1.22911e+07$	$2.8871e+06$	$2.13282e+06$	1.511272
		DE	725002	$6.05199e+06$	$3.0167e+06$	$1.36119e+06$	1.300963
F13	1300	PSO	2158.22	$5.12002e+08$	$2.6656e+07$	$1.03773e+08$	1.125402
		ESEA	1587.47	24790.2	6587.37	5518.93	1.334661
		DE	254821	$2.33906e+06$	$1.30353e+06$	516312	1.662164
F14	1400	PSO	43345.6	811632	251465	164150	1.532055
		ESEA	18985.4	777017	135860	130431	1.739088
		DE	76740.9	977009	357016	192177	1.257321
F15	1500	PSO	2005.14	26734.7	10673.1	6840.36	1.085667
		ESEA	1792.52	20190.8	9067.54	6602.38	1.300832

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

Table 7 Numerical results accumulated by our proposed ESEA *versus* DE and PSO by solving F_{16} to F_{30} of CEC 2017 in fifty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
		DE	3736.19	4687.75	4284.34	195.447	1.360924
F16	1600	PSO	2372.06	4501.18	3427.63	462.177	1.179998
		ESEA	1972.19	4068.59	3082.1	427.816	1.392159
		DE	2761.79	3475.29	3182.58	161.092	2.137903
F17	1700	PSO	2521.44	4068.33	3258.49	331.26	1.959108
		ESEA	2112.2	3142.54	2763.91	234.666	2.159728
		DE	$3.67699e+06$	$1.55985e+07$	$8.29535e+06$	$3.09203e+06$	1.365837
F18	1800	PSO	418451	$9.67081e+06$	$3.14843e+06$	$2.01038e+06$	1.200877
		ESEA	321613	$6.91886e+06$	$2.12064e+06$	$1.51407e+06$	1.406279
		DE	32237.5	445791	168709	77543.1	6.217998
F19	1900	PSO	2839.89	358985	29598.5	51952	6.135213
		ESEA	4829.68	41806	16337.8	8610.9	6.499750
		DE	2726.59	3552.38	3230.45	165.601	2.339555
F20	2000	PSO	2469.07	3673.13	3121.47	308.106	2.142325
		ESEA	2183.41	3253.79	2729.21	244.342	2.612191
		DE	2618.97	2716.85	2682.26	18.5522	3.426479

(*Continued*)

Table 7 Numerical results accumulated by our proposed ESEA *versus* DE and PSO by solving F_{16} to F_{30} of CEC 2017 in fifty dimension. *(Continued)*

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F21	2100	PSO	2469.32	2788.7	2632.18	57.1423	3.103253
		ESEA	2452.01	2754.03	2582.1	75.9738	3.319823
		DE	12888.1	15415.1	14862.5	506.515	3.969959
F22	2200	PSO	2601.02	11797.1	9550.01	1376.43	3.890069
		ESEA	2300	15343.1	11676	4743.5	3.795784
		DE	3055.58	3139.2	3107.97	16.9574	4.318263
F23	2300	PSO	3036.15	3838.95	3459.93	150.119	4.251395
		ESEA	2928.53	3274.04	3090.59	92.1445	6.612321
		DE	3252.74	3332.06	3303.24	16.2321	4.575222
F24	2400	PSO	3289.05	4120.71	3739.28	180.829	4.429763
		ESEA	3062.97	3690.59	3365.42	133.769	4.647092
		DE	3043.43	3079.88	3054.93	8.90644	4.322516
F25	2500	PSO	3050.52	6602.35	4086.67	900.366	4.149483
		ESEA	2961.15	3113.21	3057.16	34.3966	4.384233
		DE	7094.28	7732.14	7477.59	138.581	5.294658
F26	2600	PSO	3293.14	12464.5	9825.3	2000.47	5.097603
		ESEA	2900	10283.5	7193.82	2234.38	5.317807
		DE	3377.84	3493.66	3437.76	22.3263	6.096795
F27	2700	PSO	3322.31	5142.76	4250.21	336.362	5.910171
		ESEA	3357.4	4077.81	3691.17	168.706	6.153214
		DE	3311.01	3391.35	3342.5	16.9807	5.407784
F28	2800	PSO	3414.96	6822.45	5137.51	917.433	5.231028
		ESEA	3264.88	3412.13	3317.75	31.0122	5.438561
		DE	4451.39	5201.76	4930.68	148.3	3.853938
F29	2900	PSO	4206.03	6201.02	5012.37	413.17	3.642380
		ESEA	3474.48	5387.16	4343.42	402.245	3.850753

PSO, particle swarm optimization; DE, differential eveloution; CEC, congress on evolutionary computing.

Table 8 Numerical results provided by our proposed ESEA *versus* GA by solving F_1 to F_{15} of CEC 2017 in ten dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F01	100	GA	112.092	6089.61	1676.9	1452.9	0.127179
		ESEA-II	100.005	12037.6	1590.48	2494.76	0.238215
F02	200	GA	200	200.003	200	0.00050849	0.160300
		ESEA-II	200	200	200	$2.94792e-13$	0.285582
F03	300	GA	345.22	7294.83	1536.46	1222.11	0.126667
		ESEA-II	305.7	3155.48	1096.46	718.423	0.235733
F04	400	GA	400.011	407.172	405.925	1.41251	0.125888
		ESEA-II	400.432	406.981	406.004	1.19663	0.235002
F05	500	GA	502.985	523.879	511.629	4.90388	0.135319
		ESEA-II	502.985	517.909	508.409	3.46527	0.245941
F06	600	GA	600	600	600	$3.37708e-07$	0.219071
		ESEA-II	600	600	600	0	0.341698
F07	700	GA	712.493	732.097	720.244	4.79181	0.153117
		ESEA-II	712.377	727.625	718.803	3.44615	0.270635
F08	800	GA	801.99	820.894	809.872	4.47999	0.155578
		ESEA-II	802.985	818.633	808.286	3.90871	0.269864
F09	900	GA	900	900.974	900.065	0.211771	0.144278
		ESEA-II	900	900.003	900	0.000541231	0.259670
F10	1000	GA	1125.72	2264.87	1614.76	279.236	0.184973
		ESEA-II	1013.6	2013.93	1475.29	238.52	0.302425
F11	1100	GA	1100.1	1114.05	1106.67	3.16052	0.133612
		ESEA-II	1100.89	1113.65	1104.64	2.27661	0.249511
F12	1200	GA	7653.31	$5.68181e+06$	806456	$1.22143e+06$	0.156797
		ESEA-II	1422.05	$1.03694e+06$	137324	244413	0.271253
F13	1300	GA	1318.38	25228.3	8493.43	6348.77	0.151822
		ESEA-II	1325.92	31550.8	7784.58	6584.7	0.267060
F14	1400	GA	1415.71	21307.7	3738.48	3588.45	0.154964
		ESEA-II	1408.24	9809.97	3117.73	2346.15	0.278408
F15	1500	GA	1506.45	12759.5	3628.02	2906.48	0.150104
		ESEA-II	1501.52	10003.9	2905.3	2088.73	0.263929

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

Table 9 Numerical results provided by our proposed ESEA *versus* GA by solving F_{16} to F_{29} of CEC 2017 in ten dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F16	1600	GA	1600.29	1838.93	1655.38	71.5908	0.160117
		ESEA-II	1600.51	1840.26	1632.16	52.9978	0.277952
F17	1700	GA	1700.33	1781.25	1718.05	16.4145	0.183127
		ESEA-II	1700.02	1759.52	1708.69	10.8647	0.303991
F18	1800	GA	1884.09	35113.3	11884.4	9293.88	0.156321
		ESEA-II	1877.01	37737.9	9278.55	8821.38	0.276295
F19	1900	GA	1905.54	22851.4	6401.07	4872.72	0.359799
		ESEA-II	1918.89	16530.9	4546.27	3604.3	0.497721
F20	2000	GA	2000	2033.17	2008.22	9.87318	0.197561
		ESEA-II	2000	2020.24	2001.63	3.9204	0.323643
F21	2100	GA	2202.46	2326.05	2285.75	47.8996	0.301348
		ESEA-II	2200.03	2321.12	2289.07	44.8485	0.437779
F22	2200	GA	2228.69	2306.26	2299.41	10.1576	0.339447
		ESEA-II	2300	2301.69	2300.28	0.49146	0.490457
F23	2300	GA	2603.21	2628.1	2611.59	4.61267	0.392353
		ESEA-II	2604.21	2620.5	2611.41	4.18425	0.565490
F24	2400	GA	2500	2758.83	2733.23	48.02	0.375891
		ESEA-II	2500	2764.68	2728.64	57.9942	0.513558
F25	2500	GA	2898.27	2951.49	2935.63	20.4537	0.390659
		ESEA-II	2897.74	2951.18	2933.58	21.5788	0.532507
F26	2600	GA	2800	3072.87	2920.22	43.2593	0.446788
		ESEA-II	2600	3001.18	2903.14	48.6392	0.602823
F27	2700	GA	3090.38	3103.02	3096.13	2.95396	0.516454
		ESEA-II	3089.52	3099.24	3094.35	2.80492	0.671058
F28	2800	GA	3100	3446.48	3300.48	136.51	0.447005
		ESEA-II	3100	3411.82	3303.15	129.128	0.599055
F29	2900	GA	3138.03	3227.28	3173.29	21.001	0.374510
		ESEA-II	3133.85	3227.59	3164.73	20.7597	0.519463

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

Table 10 Numerical results provided by our proposed ESEA *versus* GA by solving F_1 to F_{16} of CEC 2017 in thirty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F01	100	GA	1152.5	$1.37974e+07$	$1.50593e+06$	$3.1845e+06$	0.660831
		ESEA-II	120.244	46288.3	5892.95	7677.1	1.119098
F02	200	GA	204.38	897.573	363.515	141.441	0.974057
		ESEA-II	200	1103.86	242.53	132.417	1.509185
F03	300	GA	4377.59	25141	11831.7	4593.96	0.702093
		ESEA-II	5015.47	33382.6	13701.5	5823.17	1.095244
F04	400	GA	432.685	580.831	511.665	27.4456	0.689760
		ESEA-II	404.091	516.448	489.816	22.3533	1.093285
F05	500	GA	519.357	584.538	551.245	15.2413	0.792540
		ESEA-II	530.509	597.441	556.28	16.1532	1.202547
F06	600	GA	600	600.025	600.001	0.00366595	1.266414
		ESEA-II	600	600	600	$2.68326e-08$	2.063278
F07	700	GA	756.267	853.569	802.344	28.5348	0.870851
		ESEA-II	758.122	833.369	789.968	16.8456	1.292511
F08	800	GA	823.457	879.466	852.081	15.0641	0.943852
		ESEA-II	830.844	894.541	854.773	14.8042	1.375917
F09	900	GA	905.076	1531.37	981.632	148.67	0.819799
		ESEA-II	900	2460.8	942.989	222.265	1.226116
F10	1000	GA	2641.81	4744.32	3714.75	508.623	1.221028
		ESEA-II	2741.09	5608.5	3763.92	561.007	1.683116
F11	1100	GA	1129.39	1748.55	1264.03	117.54	0.751172
		ESEA-II	1107	1514.34	1180.45	85.0489	1.159543
F12	1200	GA	33714.9	$8.60836e+06$	$2.06336e+06$	$1.68071e+06$	0.923895
		ESEA-II	83450	$4.529e+06$	$1.10473e+06$	943579	1.355424
F13	1300	GA	1362.76	158093	18349.1	25320.6	0.817558
		ESEA-II	1388.7	40937.8	13713.3	10575.3	1.240687
F14	1400	GA	9260.58	$2.21574e+06$	532464	509412	0.984816
		ESEA-II	12807.7	$1.47819e+06$	313357	337765	1.411275
F15	1500	GA	1580.19	34975.7	6815.7	7319.85	0.814190
		ESEA-II	1525.6	42730.3	7829.12	9214.97	1.213920

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

Table 11 Numerical results provided by our proposed ESEA *versus* GA by solving F_{16} to F_{29} of CEC 2017 in thirty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F16	1600	GA	1850.63	2974.61	2483.18	296.023	0.920448
		ESEA-II	1752.19	3182.87	2479.49	275.516	1.335441
F17	1700	GA	1711.36	2482	1987.21	189.304	1.157793
		ESEA-II	1729.76	2410.57	2003.46	160.278	1.603197
F18	1800	GA	52969.6	$3.30024e+06$	588413	531371	0.846876
		ESEA-II	58428.6	$7.34934e+06$	819487	$1.14578e+06$	1.275329
F19	1900	GA	1917.99	48445.9	8580.92	7914.14	2.719499
		ESEA-II	2067.76	56432	10527	10987	3.303746
F20	2000	GA	2032.83	2764.99	2377.22	198.368	1.309053
		ESEA-II	2009.59	2849.64	2336.61	189.388	1.749268
F21	2100	GA	2323.63	2428.27	2359.67	20.4686	2.386408
		ESEA-II	2325.35	2406.13	2352.77	16.0219	2.929541
F22	2200	GA	2300	5222.57	2359.58	408.934	2.704526
		ESEA-II	2300	6754.23	2447.79	751.223	3.321357
F23	2300	GA	2660.07	2771.96	2702.89	22.9293	3.246358
		ESEA-II	2667.51	2760.95	2707.43	19.7795	3.889651
F24	2400	GA	2839.98	2935.08	2874.17	22.8881	3.110545
		ESEA-II	2851.75	2925.55	2878.95	18.0297	3.736883
F25	2500	GA	2885.07	2937.98	2900.59	14.02	3.339694
		ESEA-II	2884.32	2926.54	2890.27	7.7786	3.966824
F26	2600	GA	3763.12	4568.42	4113.23	180.621	3.958625
		ESEA-II	3782.61	4772.88	4206.64	236.07	4.389339
F27	2700	GA	3202.4	3240.34	3222	7.78126	4.586867
		ESEA-II	3201.57	3238.95	3213.29	7.0289	4.916931
F28	2800	GA	3208.95	3317.92	3252.6	24.3448	3.920140
		ESEA-II	3162.21	3271.66	3205.44	15.9687	4.248548
F29	2900	GA	3361.51	4052.14	3626.01	159.987	2.797537
		ESEA-II	3329.41	3843.38	3557.78	141.63	3.452291

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

Table 12 Numerical results provided by our proposed ESEA *versus* GA by solving F_1 to F_{15} of CEC 2017 in fifty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F01	100	GA	116565	$1.55105e+08$	$1.35536e+07$	$2.74012e+07$	1.851306
		ESEA-II	100.825	12213.3	2510.19	2728.08	2.692427
F02	200	GA	712.758	3583.32	1735.1	735.877	1.409840
		ESEA-II	200	891.383	274.888	164.6	2.967594
F03	300	GA	25304	64592.3	41461.5	9241.66	1.856108
		ESEA-II	22612.7	95402	46528.9	13041.9	2.599555
F04	400	GA	552.507	731.424	634.296	44.6463	1.904860
		ESEA-II	428.513	611.987	502.177	49.7562	2.616350
F05	500	GA	554.344	778.794	651.423	65.8744	2.142100
		ESEA-II	561.193	659.166	604.102	23.3479	2.920949
F06	600	GA	600	600.214	600.022	0.0369354	3.961551
		ESEA-II	600	600.008	600	0.00117649	4.535922
F07	700	GA	797.821	1112.63	907.66	85.0347	2.323571
		ESEA-II	811.637	905.344	850.87	26.8966	3.086783
F08	800	GA	859.698	1081.86	951.47	64.2085	2.611588
		ESEA-II	862.217	948.58	901.296	20.3518	3.347141
F09	900	GA	973.003	12006.3	2015.82	1651.15	2.200571
		ESEA-II	900.544	6716.28	1551.85	1237.53	2.982360
F10	1000	GA	6805.89	10056.3	8347.59	796.9	3.388132
		ESEA-II	4758.78	8517.51	6628.49	818.51	4.203645
F11	1100	GA	1252.29	3127.15	1767.51	426.836	2.000565
		ESEA-II	1129.84	3057.88	1535.09	468.012	2.876479
F12	1200	GA	$2.98513e+06$	$4.00603e+07$	$8.67578e+06$	$5.93632e+06$	2.506063
		ESEA-II	269098	$5.78315e+06$	$1.74482e+06$	$1.01614e+06$	3.366186
F13	1300	GA	2114.08	76667.6	9874.46	11518.2	2.139729
		ESEA-II	1373.96	31868.4	4932.98	5252.12	2.932518
F14	1400	GA	131365	$4.14154e+06$	$1.36249e+06$	$1.08968e+06$	2.740196
		ESEA-II	121994	$1.68917e+06$	539011	377582	3.532500
F15	1500	GA	1629.11	22060.9	9566.43	6747.35	2.161552
		ESEA-II	1587.07	19471.5	8893.16	5390.16	2.936606

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

Table 13 Numerical results provided by our proposed ESEA *versus* GA by solving F_{16} to F_{29} of CEC 2017 in fifty dimension.

Problem No	Optimum	Algorithm	Best	Worst	Mean	St. Dev.	CPU Time/Run(s)
F16	1600	GA	2476.27	3851.28	3225.95	349.568	2.450315
		ESEA-II	2354.19	3996.37	3375.01	367.313	3.220540
F17	1700	GA	2290.91	3580.5	2925.46	320.594	3.148421
		ESEA-II	2136.73	3690.03	2873.87	303.226	3.894396
F18	1800	GA	78965.7	$1.34741e+07$	$3.62359e+06$	$2.68292e+06$	2.292391
		ESEA-II	232543	$1.06487e+07$	$2.56396e+06$	$2.20091e+06$	3.029270
F19	1900	GA	3882.44	43071.3	20632.3	10304.2	7.431792
		ESEA-II	4603.18	38583.1	18561.2	8814.16	8.391807
F20	2000	GA	2196.66	3510	2955.11	269.56	3.617867
		ESEA-II	2341.88	3620.78	2943.93	313.621	4.359299
F21	2100	GA	2356.43	2575.52	2460.5	69.6128	7.220507
		ESEA-II	2347.12	2500.96	2409.55	28.5521	8.674117
F22	2200	GA	2305.55	12101.9	9907.33	1812.67	8.481901
		ESEA-II	6542.18	10969.5	8688.09	945.373	9.049164
F23	2300	GA	2769.95	3032.08	2843.09	62.0126	10.131182
		ESEA-II	2788.62	2898.83	2844.46	26.4369	10.861602
F24	2400	GA	2968.36	3299.47	3045.29	89.4193	9.730596
		ESEA-II	2954.21	3064.47	3002.21	27.9304	10.493888
F25	2500	GA	3072.08	3203.05	3124.8	30.611	11.160622
		ESEA-II	2960.76	3109.66	3053.51	36.4146	11.448251
F26	2600	GA	4376.67	6186.06	4920.24	396.294	12.497063
		ESEA-II	4401.31	5537.7	4859.08	274.355	13.109032
F27	2700	GA	3302.58	3530.29	3434.49	52.595	14.539892
		ESEA-II	3247.35	3464.74	3336.11	45.5723	16.866424
F28	2800	GA	3388.9	3540.68	3448.02	33.5245	13.425890
		ESEA-II	3280.21	3324	3309.19	7.28352	14.927486
F29	2900	GA	3430.61	4547.15	3871.33	262.686	8.394708
		ESEA-II	3259.05	4421.54	3795.76	245.301	9.860898

PSO, particle swarm optimization; GA, genetic algorithm; CEC, congress on evolutionary computing.

by employing a GA to replace the PSO as a constituent algorithm in the framework of the designed ameliorated algorithm ensemble algorithm denoted by ESEA-II.

The second column of the Tables 8–13 include the known optimal solution of each used benchmark function denoted by F_1 to F_{29}. The 4^{th} column of the aforementioned tables gathered the best optimal solution found by GA and the proposed algorithm denoted by ESEA-11. The 5^{th} column of the above same tables provide the worst objective function values of each benchmark function. Similarly, 6^{th} column of each table includes the average objective function values and 7^{th} column of each table standard deviation in objective function values by executing both GA and ESEA-II with fifty-one different random seeds to solve each benchmark function. The smaller values gathered in all tables represent better results of the corresponding algorithms. The last column of each table presents the average Central processing unit (CPU) time elapsed by the proposed algorithm and each competitor algorithm.

Figure 2 displays the evolution in minimum objective function values concerning ESEA, PSO, and DE algorithm for benchmark function, $F_1 - F_5$ solved in different dimensions. The first panel is for dimension $n = 10$, the 2^{nd} panel is for $n = 30$ and the 3^{rd} one is for $n = 50$. These figures, clearly exhibit that the convergence ability of the proposed ESEA algorithm is much better than PSO and DE algorithm while converging toward the known optimal value as listed in the last column of Table 1 as comprising the characteristics of F_1 to F_5 benchmark functions [63].

Figure 3 demonstrates the convergence graph of the ESEA, PSO, and DE algorithm by solving the benchmark functions from F_6 to F_{10} in different search dimensions. The first panel is for F_6 to F_{10} with dimension $n = 10$, the 2^{nd} panel is with respect to $n = 30$, and the 3^{rd} one is for F_6- F_{10} with $n = 50$. These figures demonstrate the convergence ability of the proposed ESEA algorithm *versus* PSO and DE algorithm while moving toward the known optimal value as listed in the last column of Table 1 for F_1 to F_5 benchmark functions [63].

The evolutions in the minimum objective function values, when applying ESEA, PSO, and DE algorithms over benchmark function, F_{11} to F_{15} are given in Figure 4. The first panel is for $n = 10$, the 2^{nd} panel is for $n = 30$, and the 3^{rd} one is for $n = 50$. These figures clearly exhibit that the convergence ability of the proposed ESEA algorithm is much better than PSO and DE algorithms while marching toward the known optimal value as listed in the last column of Table 1 as comprising the characteristics of F_{11} to F_{15} benchmark functions [63].

The convergence graphs displayed by ESEA, PSO, and DE algorithm by solving benchmark functions likewise F_{16} to F_{20} in different dimensions are given in Figure 5. The first panel is with respect to dimension $n = 10$, the 2^{nd} panel is with respect to $n = 30$, and the 3^{rd} one is with respect to $n = 50$. These figures, clearly exhibit that the convergence ability of the proposed ESEA algorithm is much better than PSO and DE algorithm in term of converging toward the known optimal value as listed in the last column of Table 1 that mainly describe the key features of F_{16} to F_{20} benchmark functions [63].

The convergence speed toward the known optimal value of the benchmark functions denoted by $F_{17} - F_{25}$ with respect to ESEA, PSO, and DE are included in Figure 6. The first panel contains convergence curves of $F_{17} - F_{25}$ in $n = 10$, the 2^{nd} panel comprising

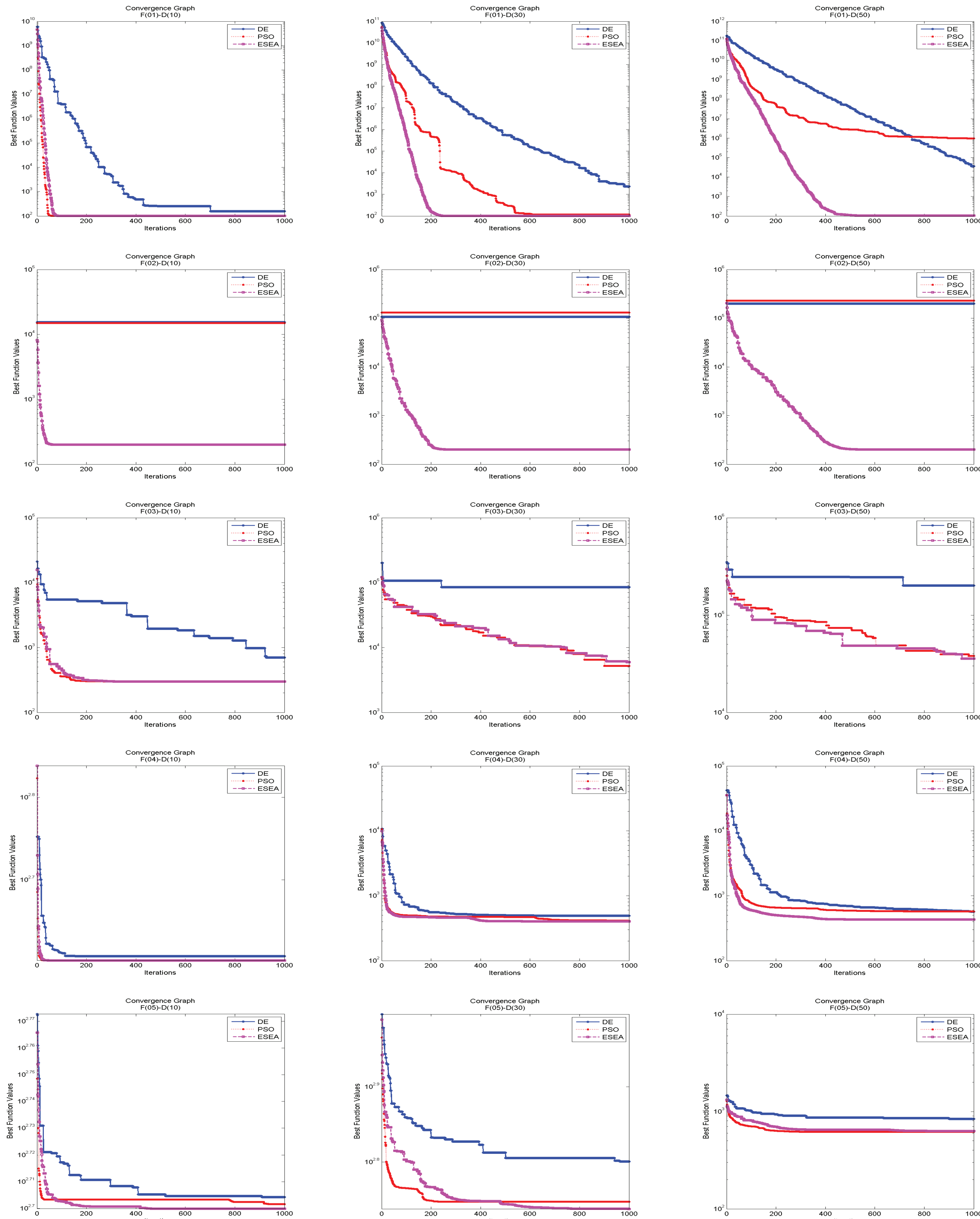

Figure 2 | Evolution in the best function values of panel is for problems with fifty dimension.

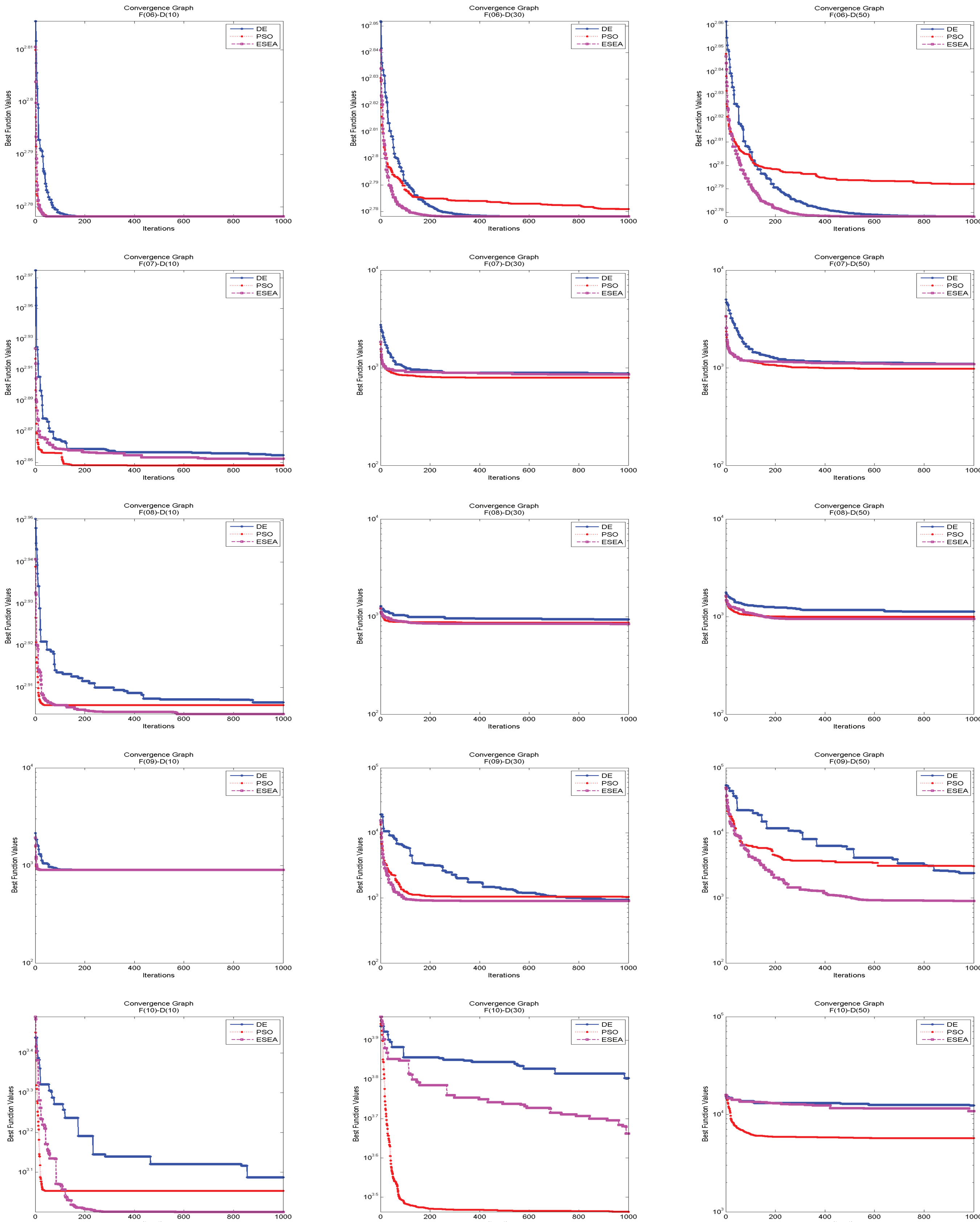

Figure 3 | Evolution in the best function values of panel is for problems with fifty dimension.

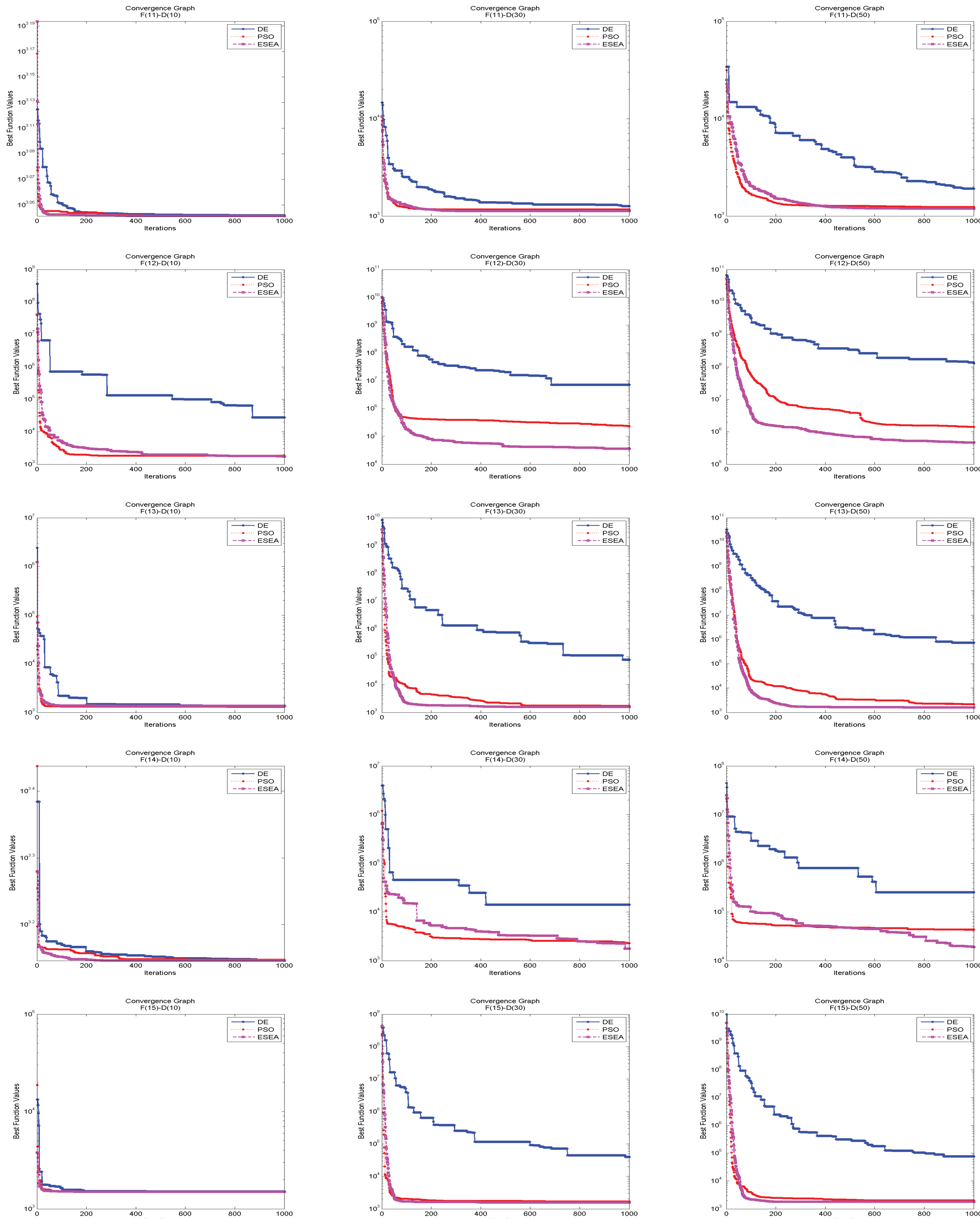

Figure 4 | Evolution in the best function values of panel is for problems with fifty dimension.

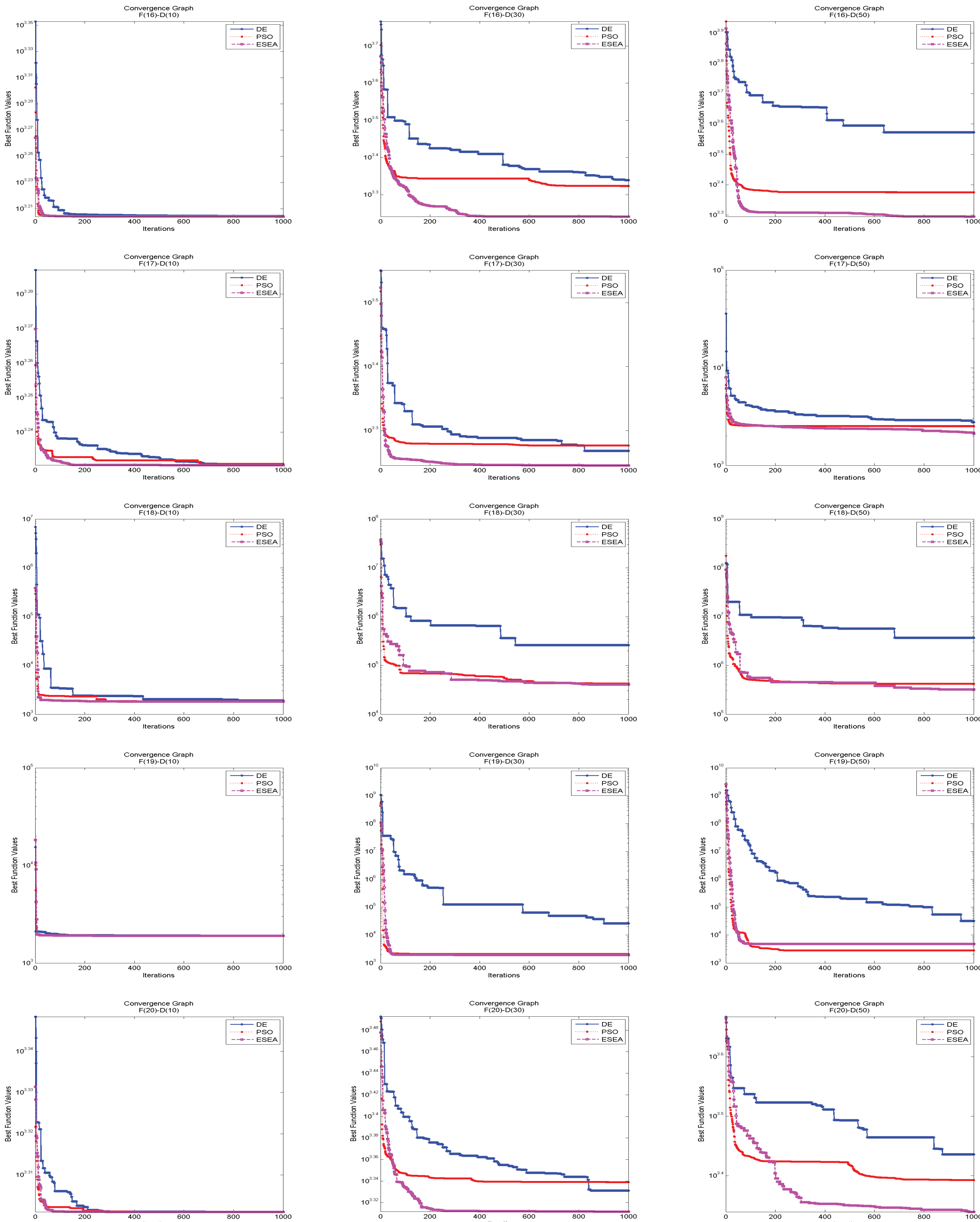

Figure 5 | Evolution in the best function values of panel is for problems with fifty dimension.

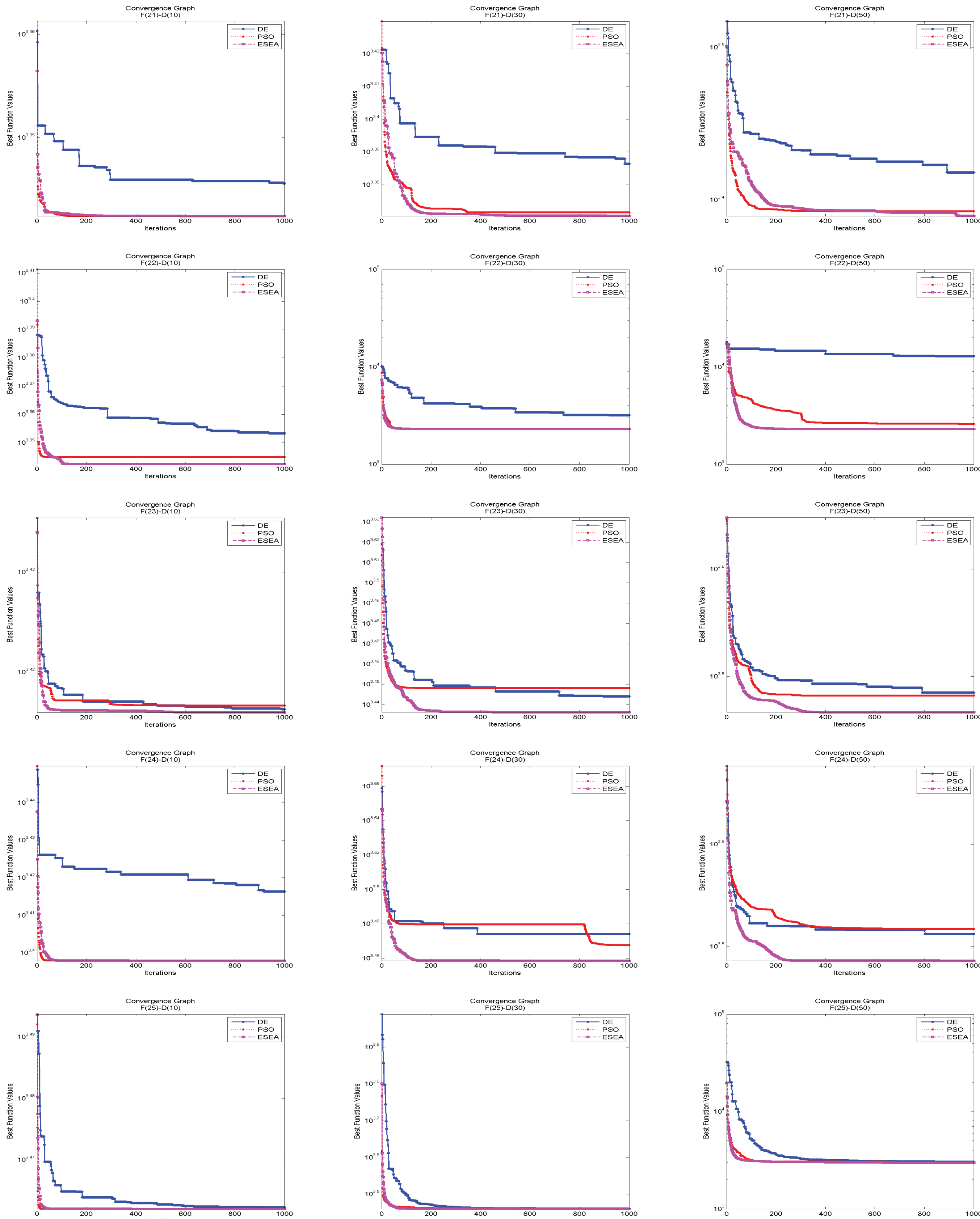

Figure 6 Evolution in the best function values of panel is for problems with fifty dimension.

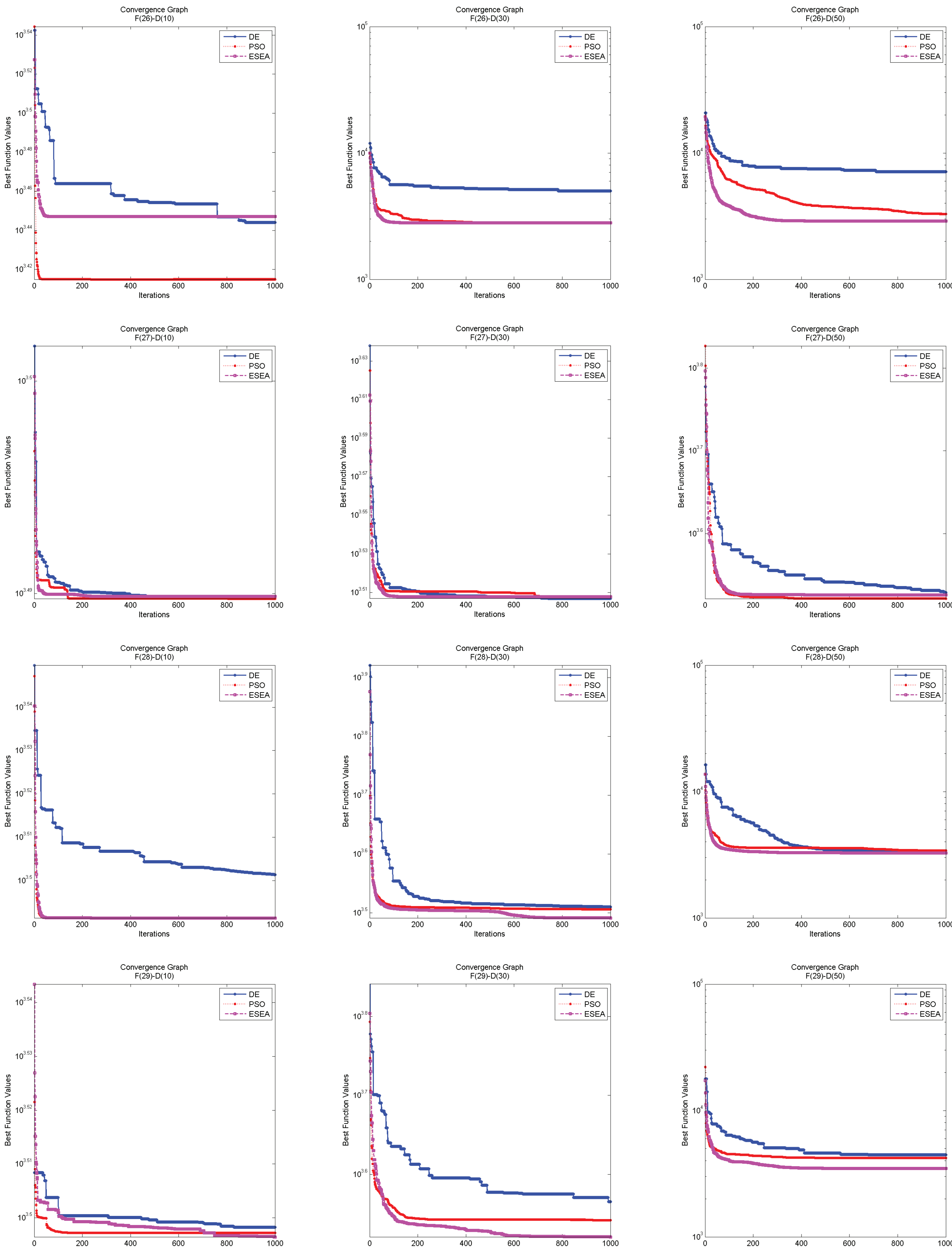

Figure 7 | Evolution in the best function values of panel is for problems with fifty dimension.

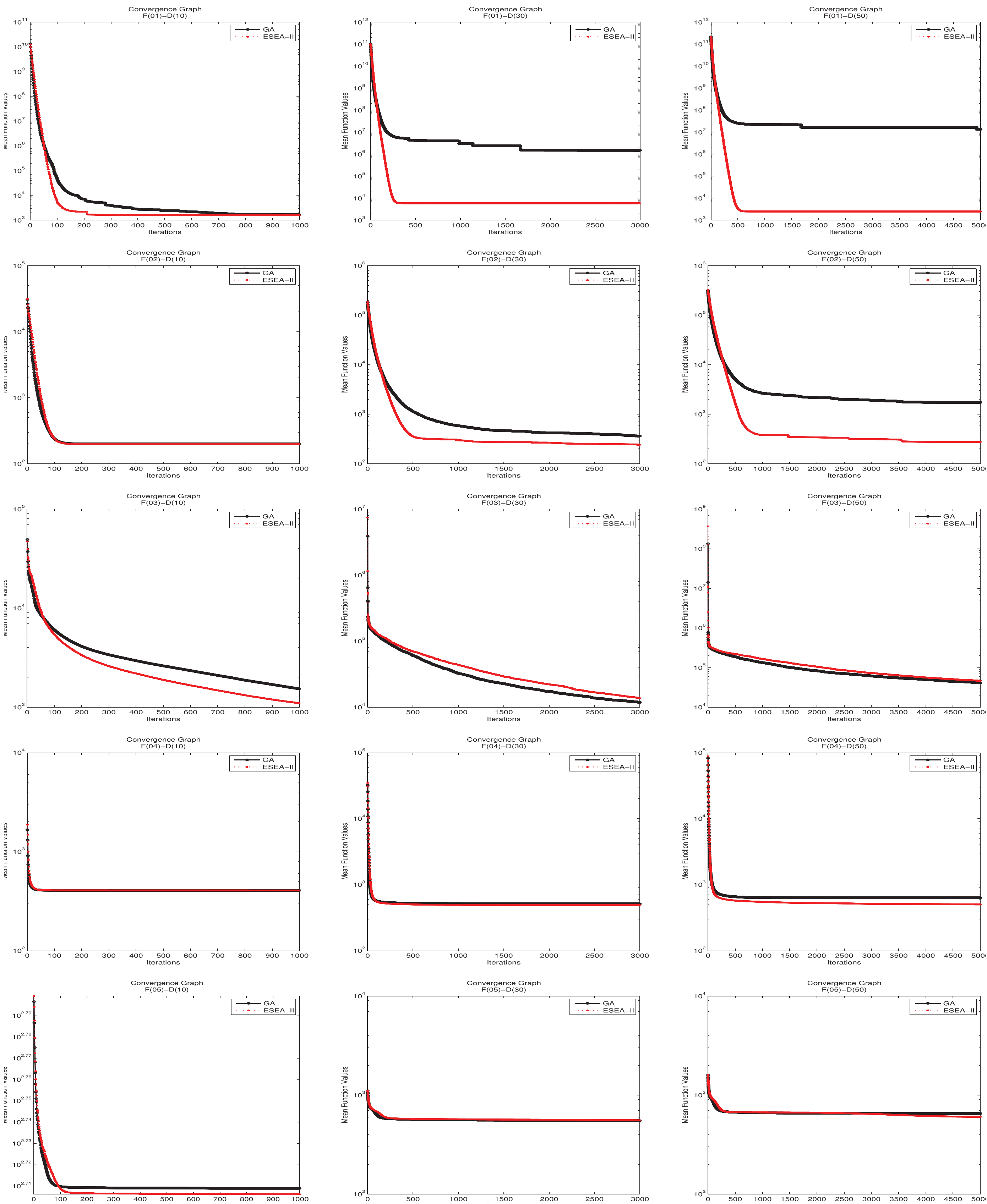

Figure 8 | Evolution in the average function values of panel is for problems with fifty dimension.

Figure 9 Comparison of best function values *versus* optimal values of benchmark functions panel is for problems with fifty dimension.

the convergence of the $F_{17} - F_{25}$ in $n = 30$, and the 3^{rd} panel provide the convergence graph of the problems including $F_{17} - F_{25}$ in $n = 50$ dimension. These figures, clearly exhibit that the convergence ability of the proposed ESEA algorithm is much better than PSO and DE algorithm while converging toward the known optimal value as listed in the last column of Table 1 as comprising the characteristics of F_{17} to F_{25} benchmark functions [63].

Figure 7 presents the visual convergence graph of the ESEA, PSO, and DE algorithm while solving the benchmark functions from F_{26} to F_{29} in different search dimensions. The first panel is with respect to solving F_{26} to F_{29} in $n = 10$ dimension, the 2^{nd} panel is with respect to $n = 30$, and the 3^{rd} one for F_{26}- F_{29} with $n = 50$. These figures, demonstrate the convergence ability of the proposed ESEA algorithm *versus* PSO and DE algorithm while moving toward the known optimal value as listed in the last column of Table 1 for F_{26} to F_{29} benchmark functions [63].

In Figure 8, the evolution in the average objective function value displayed by ESEA-II *versus* GA while solving benchmark functions denoted by F_1 to F_5. The 1^{st} panel in the 8 for F_1 to F_5 in ten dimension, 2^{nd} panel by solving F_1 to F_5 in thirty dimension, and 3^{rd} panel is for the same functions in fifty dimension.

Figure 9 presents comparison of best function values *versus* optimal values of benchmark functions F_1 to F_{29} in $n = 10$, 30, and 50 dimensions which are solved by DE, PSO, and ESEA. The 1^{st} panel is designated for the mentioned problem with ten dimension, 2^{nd} panel attributes problems with thirty dimension, and 3^{rd} panel deal with problems with fifty dimension.

The numerical results as summarized in all tables indicate the consistency of the proposed ESEA algorithm while solving large-scale global optimization problems having had continuous search space. We have also performed the statistical analysis by employing Wilcoxons rank-sum test with $\lambda = 0.05$ of freedom as commonly used in the existing literature of evolutionary computing, in order to establish a fair comparison between the proposed ESEA algorithm *versus* two well established constituent algorithms including the PSO and DE algorithm. After statistical investigation, we awarded the suggested algorithm first rank, PSO got the second position, and DE was declared third by solving most of the benchmark functions in terms of fast convergence toward their respective known optimal solution.

4. CONCLUSION

In the last two decades, EC has played an authoritative role in solving various complex optimization and search problems. A family of EAs under the umbrella of the EC field has many advantages and various distinguishing features as compared to existing traditional mathematical programming techniques while solving complicated optimization problems. EAs have the ability of a collective learning approach, self-adaptation, robustness, and particularly no need for derivative information to conduct their search process. EAs work on a uniformly and randomly generated set of solutions and have been successfully applied to many real-world problems that posed in different disciplines of engineering technologies, business communities, commercial applications. The combined use of different learning techniques might be useful in alleviating the shortcomings of the various baseline EAs in the form of hybridization or fusion of these techniques. Hybrid EAs have a substantial importance in attaining global optimal solutions for the problem at hand as compared to single strategy-based EAs. This paper proposes an ensemble strategy-based EA to solve large-scale global optimization problems with complicated search space. The proposed algorithm so-called ESEA has employed some existing EA as constituent algorithms based on dynamic resources allocation procedure. The constituent algorithms include the PSO, DE algorithm, BA and TLBA, and GA to evolve its population. Promising simulations results have been found by the proposed algorithm over a series of benchmark functions for most of the benchmark functions as compared PSO and DE, and GAs. This good algorithmic behavior of the suggested algorithm is attributed to the intelligent and adaptive strategy of distributed procedure. The proposed algorithm is much flexible and can easily be adapted to a parallel computing network for finding the solution of complicated real-world problems in less computation time. Furthermore, multiple search operators can be assumed in the constituent algorithms in the framework of the proposed algorithm keeping in view the fact that different problems suit different search operators.

In the future, we intend to further investigate the algorithmic behavior of the suggested algorithm upon more complicated benchmark functions and real-world optimization problems.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHORS' CONTRIBUTIONS

The main concept and experimental work were performed by Wali Khan Mashwani and Syed Nouman Ali Shah. The critical revisions and writing, analysis of this manuscript was mainly handled by Wali Khan Mashwani, Syed Nouman Ali Shah along with Abdelouahed Hamdi and Samir Brahim Belhaouari who's helped us in revising this paper. All authors have reviewed and approved the final manuscript.

Funding Statement

The first two authors are thankful to National Research Programme for Universities (NRPU) Project # 5892 awarded by Higher Education Commission (HEC), Pakistan.

ACKNOWLEDGMENTS

The authors would like to thankful to the referees for their insightful and valuable comments to significantly improve this Manuscript. The authors are highly thankful to Director of Institute of Numerical Sciences, Kohat University of Sciences and Technology for providing all experimental facilities and convenient environment to accomplish this study and successfully completed the NRPU Project 5892.

REFERENCES

[1] H. Garg, Solving structural engineering design optimization problems using an artificial bee colony algorithm, J. Ind. Manag. Optim. 10 (2014), 777.

[2] H. Garg, An Efficient Biogeography Based Optimization Algorithm for Solving Reliability Optimization Problems, Swarm Evol. Comput. 24 (2015), 1–10.

[3] H. Garg, S. Sharma, Multi-objective reliability-redundancy allocation problem using particle swarm optimization, Comput. Ind. Eng. 64 (2013), 247–255.

[4] H. Garg, A hybrid PSO-GA Algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016), 292–305.

[5] A. Lasisi, N. Tairan, R. Ghazali, W.K. Mashwani, S.N. Qasem, H. Garg, A. Arora, Predicting crude oil price using fuzzy rough set

and bio-inspired negative selection algorithm, IJSIR. 10 (2019), 25–37.
[6] W.K. Mashwani, Hybrid multiobjective evolutionary Algorithms: a survey of the state-of-the-art, Int. J. Comput. Sci. Iss. 8 (2011), 374–392.
[7] W.K. Mashwani, A. Salhi, Multiobjective memetic Algorithm based on decomposition, Appl. Soft Comput. 21 (2014), 221–243.
[8] R.S. Patwal, N. Narang, H. Garg, A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units, Energy. 142 (2018), 822-837.
[9] A. Hamdi, S.K. Mishra, Decomposition methods based on augmented lagrangians: a survey, in: Topics in Nonconvex Optimization: Theory and Applications, Springer, 2011. pp. 175–204.
[10] W.K. Mashwani, MOEA/D with DE and PSO: MOEA/D-DE+PSO, in The Thirty-first SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, 2011, pp. 217–221.
[11] W.K. Mashwani, A. Salhi, A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation, Appl. Soft Comput. 12 (2012), 2765–2780.
[12] R.E. Miller, Optimization: Foundations and Applications, John Wiley & Sons, 1999. http://bcs.wiley.com/he-bcs/Books?action=index&itemId=0471351695&bcsId=9672
[13] A. Törn, A. Žilinskas, Global Optimization, Springer, Berlin, Heidelberg, 1989.
[14] S. Wang, Z. Ma, Supervisory and optimal control of building hvac systems: a review, HVAC&R Res. 14 (2008), 3–32.
[15] P.T. Boggs, J.W. Tolle, Sequential quadratic programming, Acta Numer. 4 (1995), 1–51.
[16] E. Angel, A survey of approximation results for local search algorithms, in: E. Bampis, K. Jansen, C. Kenyon (Eds.), Efficient Approximation and Online Algorithms, Springer, Berlin, Heidelberg, Germany, 2006, pp. 30–73.
[17] G. Lindfield, J. Penny, 8-ptimization methods, Numer. Methods. (Third Edition), Academic Press, Boston, 2012, 371–432.
[18] O. Kramer, D.E. Ciaurri, S. Koziel, Derivative-free optimization, in: S. Koziel, X.-S. Yang (Eds.), Computational Optimization, Methods and Algorithms, Springer, Berlin, Heidelberg, Germany, 2011, pp. 61–83.
[19] J. Brownlee, Clever algorithms: nature-inspired programming recipes, (2011).
[20] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Second ed., Springer Springer-Verlag Berlin Heidelberg, 2015.
[21] S.M. Elsayed, R.A. Sarker, D.L. Essam, A genetic algorithm for solving the cec'2013 competition problems on real-parameter optimization, in 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, pp. 356–360.
[22] H. Mühlenbein, T. Mahnig, Mathematical analysis of evolutionary algorithms, in: Essays and Surveys in Metaheuristics, Operations Research/Computer Science Interfaces Series, Springer, Boston, MA, 15 (2002), pp. 525–556.
[23] X. Qian, X. Wang, Y. Su, L. He, An effective hybrid evolutionary algorithm for solving the numerical optimization problems, J. Phys. Conf. Ser. 2nd International Conference on Machine Vision and Information Technology (CMVIT 2018), Hong Kong, 1004 (2018), 23–25.
[24] N. Siddique, H. Adeli, Physics-based search and optimization: inspirations from nature, Expert Sys. J. Knowl. Eng. 33 (2016), 607–623.
[25] X. Yu, M. Gen, Introduction to Evolutionary Algorithms, Springer Science & Business Media, Awaji, Japan, 2010.
[26] Z. Yoshida, The Challenge of Macro-Systems. In: Nonlinear Science, Springer, Berlin, Heidelberg, 2010.
[27] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, first ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[28] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, USA, 1998.
[29] J.R. Koza, Genetic programming: On the programming of computers by means of natural selection, Complex adaptive systems, MIT Press, 1993, pp. I–XVIII, 1–419.
[30] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, Inc., New York, NY, USA, 1996.
[31] H. Beyer, H. Schwefel, Evolution strategies - a comprehensive introduction, Natural Comput. 1 (2002), 3–52.
[32] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide to Genetic Programming, Lulu Enterprises Ltd, UK, 2008.
[33] Y.Y. AO, An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design, Engineering. 2 (2010), 65–77.
[34] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (2005), 448–642.
[35] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag, Berlin, Heidelberg, Germany, 2005.
[36] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997), 341–359.
[37] J. Kennedy, The particle swarm: social adaptation of knowledge, in Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), Indianapolis, IN, USA, 1997, pp. 303–308.
[38] J. Kennedy, R. berhart, Particle swarm optimization, in Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, 1942–1948.
[39] B. Christian, Ant colony optimization: introduction and recent trends, Phys. Life Rev. 2 (2005), 353–373.
[40] K. Jung, S. Tripti, K. Brijesh, T. Geetam, B. Karan, H. Won, Research article intercluster ant colony optimization algorithm for wireless sensor network in dense environment, Int. J. Distrib. Sens. N. (2014).
[41] H. Shah, N. Tairan, H. Garg, R. Ghazali, Global gbest guided-artificial bee colonyalgorithm for numerical function optimization, Computer. 7 (2018).
[42] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010), 3166–3173.
[43] X.-S. Yang, S. Deb, Cuckoo search via lévy flights, in 2009 World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 2009, pp. 210–214.
[44] J. Xie, Y. Zhou, H. Chen, A novel bat algorithm based on differential operator and lévy flights trajectory, Comput. Intell. Neurosci. 2013 (2013).
[45] X.-S. Yang, A new metaheuristic bat-inspired algorithm, J.R. González, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence,Springer, Berlin, Heidelberg, 2010, 65–74.

[46] X.-S. Yang, X. He, Bat algorithm: literature review and applications, Int. J. Bio-Inspired Comput. 5 (2013), 141–149.
[47] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The bees algorithm, in Technical Note: MEC 0501, Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, UK, 2005.
[48] B. Yuce, M. Packianather, E. Mastrocinque, D. Pham, A. Lambiase, Honey bees inspired optimization method: the bees algorithm, Insects. 4 (2013), 646–662.
[49] N. Farahlina Johari, Z. Azlan, M. Noorfa, U. Amirmudin, Firefly Algorithm for optimization problem, Appl. Mech. Mater. 421 (2013), 512–517.
[50] X.-S. Yang, Swarm intelligence based Algorithms: a critical analysis, Evol. Intell. 7 (2014), 17–28.
[51] D.H. Wolpert, W.G. McReady, No Free Lunch Theorems for Search (Technical Report SFI-TR-02-010), Santa Fe Institute, USA, 1995.
[52] A.S. Kwasnicka, Evolutionary Algorithms and their applications to engineering problems, Neural Comput. Appl. 32 (2020), 12363–12379.
[53] Y. Feng, G.-G. Wang, X.-Z. Gao, A novel hybrid cuckoo search algorithm with global harmony search for 0-1 knapsack problems, Int. J. Comput. Intell. Syst. 9 (2016), 1174–1190.
[54] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Infor. Sci. 478 (2019), 499–523.
[55] C. Grosan, A. Abraham, Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews, A. Abraham, C. Grosan, H. Ishibuchi (Eds.), Hybrid Evolutionary Algorithms. Studies in Computational Intelligence, Springer, Berlin, Heidelberg, 75 (2007).
[56] H. Kundra, H. Sadawarti, Hybrid algorithm of cuckoo search and particle swarm optimization for natural terrain feature extraction, Res. J. Infor. Technol. 7 (2015), 58-69.
[57] R. Mallipeddi, P. Suganthan, Differential evolution algorithm with ensemble of populations for global numerical optimization, OPSEARCH. 46 (2009), 184–213.
[58] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011), 1679–1696.
[59] W.K. Mashwani, Enhanced versions of differential evolution: state-of-the-art survey, Int. J. Comput. Sci. Math. 5 (2014), 107–126.
[60] W.K. Mashwani, A. Salhi, O. Yeniay, H. Hussian, M.A. Jan, Hybrid non-dominated sorting genetic algorithm with adaptive operators selection, Appl. Soft Comput. 56 (2017), 1–18.
[61] J.A. Vrugt, B.A. Robinson, Improved evolutionary optimization from genetically adaptive multimethod search, Proc. Natl. Acad. Sci. U. S. A. 104 (2007), 708711.
[62] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2009), 243–259.
[63] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-parameter Numerical Optimization, Technical Report, Nanyang Technological University, Singapore, 2016.
[64] A.E. Dor, M. Clerc, P. Siarry, A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization, Comput. Optim. Appl. 53 (2012), 271–295.
[65] R. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (2012), 535–560.
[66] R.V. Rao, Teaching-learning-based optimization algorithm, in: Teaching Learning Based Optimization Algorithm, Springer, Cham, 2016, pp. 9–39.
[67] R.V. Rao, V. Patel, An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, Sci. Iran. 20 (2013), 710–720.
[68] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (2011), 303–315.
[69] U.K. Chakraborty, Advances in Differential Evolution, first ed., Springer Publishing Company, Incorporated, 2008.
[70] G. Wu, R. Mallipeddi, P.N. Suganthan, Ensemble strategies for population-based optimization algorithms-a survey, Swarm Evol. Comput. 44 (2019), 695–711.
[71] X. Yu, M. Gen, in: R. Roy (Ed.), Introduction to Evolutionary Algorithms, Springer, London, England, 2011.
[72] H.-Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, J. Glob. Optim. 27 (2003), 105–129.
[73] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, first ed., John Wiley and Sons, Chichester, UK, 2001.
[74] B. Qu, P. Suganthan, Constrained multi-objective optimization Algorithm with an ensemble of constraint handling methods, Eng. Optim. 43 (2011), 403–416.
[75] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017), 533–548.
[76] W.K. Mashwani, A. Hamdi, M.A. Jan, A. Gökta, F. Khan, Large-scale global optimization based on hybrid swarm intelligence algorithm, J. Intell. Fuzzy Syst. 37 (2020), 1257–1275.
[77] N. Dey, A.S. Ashour, Chapter 1 - Computing in medical image analysis, in: N. Dey, A. S. Ashour, F. Shi, V.E. Balas (Eds.), Soft Computing Based Medical Image Analysis, Academic Press, 2018, pp. 3–11.
[78] S.M. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution Algorithm With novel mutation and crossover strategies for global numerical optimization, IEEE Trans. Syst. Man Cybern. Syst. Part. B. 42 (2012), 482–500.
[79] W. Khan, Hybrid multiobjective evolutionary algorithm based on decomposition (PhD), Department of Mathematical Sciences, University of Essex, Colchester, UK, 2012. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.549297
[80] R.A. Khanum, M.A. Jan, W.K. Mashwani, N.M. Tairan, H.U. Khan, H. Shah, On the hybridization of global and local search methods, J. Intell. Fuzzy Syst. 35 (2018), 3451–3464.
[81] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (2010), 61–106.
[82] N. Noman, H. Iba, Accelerating differential evolution using an adaptive local search, IEEE Trans. Evol. Comput. 12 (2008), 107–125.
[83] J. Zhang, C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009), 945–958.
[84] A.W. Iorio, X. Li, Solving rotated multi-objective optimization problems using differential evolution, Webb G.I., Yu X. (eds) AI 2004: Advances in Artificial Intelligence. AI 2004. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3339 (2004), pp. 861–872.

[85] N. Padhye, P. Bhardawaj, K. Deb, Improving differential evolution through a unified approach, J. Glob. Optim. 55 (2013), 771–799.

[86] L. Li, F. Liu, Group Search Optimization for Applications in Structural Design, Springer, Berlin, Heidelberg,2011.

[87] K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, Natural Comput. 1 (2002), 235–306.

[88] A.P. Piotrowski, M.J. Napiorkowski, J.J. Napiorkowski, P.M. Rowinski, Swarm intelligence and evolutionary algorithms: performance versus speed, Infor. Sci. 384 (2017), 34–85.