Research Article

A Multiswarm Intelligence Algorithm for Expensive Bound Constrained Optimization Problems

Wali Khan Mashwani ¹, Ruqayya Haider,¹ and Samir Brahim Belhaouari ²

¹Institute of Numerical Sciences, Kohat University of Science & Technology (KUST) Technology, Kohat 26000, Khyber Pakhtunkhwa (KPK), Pakistan

²Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Education City, Qatar Foundation, Doha, Qatar

Correspondence should be addressed to Wali Khan Mashwani; mashwanigr8@gmail.com and Samir Brahim Belhaouari; sbelhaouari@hbku.edu.qa

Received 24 January 2021; Revised 12 February 2021; Accepted 16 February 2021; Published 28 February 2021

Academic Editor: Atif Khan

Copyright © 2021 Wali Khan Mashwani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Constrained optimization plays an important role in many decision-making problems and various real-world applications. In the last two decades, various evolutionary algorithms (EAs) were developed and still are developing under the umbrella of evolutionary computation. In general, EAs are mainly categorized into nature-inspired and swarm-intelligence- (SI-) based paradigms. All these developed algorithms have some merits and also demerits. Particle swarm optimization (PSO), firefly algorithm, ant colony optimization (ACO), and bat algorithm (BA) have gained much popularity and they have successfully tackled various test suites of benchmark functions and real-world problems. These SI-based algorithms follow the social and interactive principles to perform their search process while approximating solution for the given problems. In this paper, a multiswarm-intelligence-based algorithm (MSIA) is developed to cope with bound constrained functions. The suggested algorithm integrates the SI-based algorithms to evolve population and handle exploration versus exploitation issues. Thirty bound constrained benchmark functions are used to evaluate the performance of the proposed algorithm. The test suite of benchmark function is recently designed for the special session of EAs competition in IEEE Congress on Evolutionary Computation (IEEE-CEC'13). The suggested algorithm has approximated promising solutions with good convergence and diversity maintenance for most of the used bound constrained single optimization problems.

1. Introduction

Constrained optimization has indispensable applications and currently it is impossible to name a single industry that is not using optimization process. In general, bound constrained optimization is the mathematical process of finding the minimum or maximum value of an objective function subject to various constraints imposed on the search space of the problem to be solved. The roots of optimization process trace back to early work of Euler and Lagrange as found in the calculus of variations. A general constrained optimization problem can mathematically be described as follows:

$$\begin{aligned} \text{minimize } F(x) &= f_1(x), f_2(x), \dots, f_m(x), \\ g_i(x) &\leq 0, \quad i = 1, 2, \dots, p, \\ h_j(x) &= 0, \quad j = 1, 2, \dots, q, \\ x_l^i &\leq x^i \leq x_u^i, \quad i = 1, 2, \dots, N, \end{aligned} \quad (1)$$

where $x = (x_1, x_2, \dots, x_n)^T \in \Omega$ is the candidate solution with n decision variables/real parameters, $F(x)$ is the function with m objectives, $g_i(x) \leq 0$ are the i^{th} inequality constrained functions, $h_j(x) = 0$ are the j^{th} equality constrained functions, and $x_l^i \leq x^i \leq x_u^i$ are bound constrained over the search space x .

Problem (1) is said to be a linear programming (LP) or linear optimization problem if all m objective functions and all their corresponding constrained functions are linear. Likewise, if all m objective functions are quadratic and all their corresponding constraint functions are linear, then problem (1) is said to be quadratic programming. Furthermore, if $m = 1$, then problem (1) becomes a single objective optimization problem and if $m \geq 2$, then problem (1) is said to be a multiobjective optimization problem (MOP). If all objective functions and constrained functions are described in real values, then problem (1) will be a continuous MOP.

A linear programming (LP) or linear optimization problem is comparatively easy to solve as compared to nonlinear optimization (NLP) problems. The existence of equality constraints is quite difficult and more challengeable while solving nonlinear programming problems by guessing which constraints are active and which are not active [1].

In the last two decades, various optimization methods were developed as outlined in Figure 1. Among them, the linear programming was first introduced by the Soviet economist Leonid Kantorovich in World War II while designing a systemic plan of expenditures for US Army aiming at increasing losses of opposition. In 1940, American mathematical scientists Fang and Puthenpura have made further improvement and developed first generalized model of linear programming to schedule training of US Air Force [2]. LP problems have many advantages and disadvantages. This is quite unrealistic because most of the problems in sciences and engineering fields are naturally modeled as nonlinear problems [3].

Nonlinear optimization approaches are further subdivided in local and global optimization methods [4]. Local search methods provide local optimum with low computational cost as compared to global search methods. Newton method and sequential quadratic programming are commonly used local optimization methods. These gradient-based methods can be employed as local search optimizers. However, they are not much effective in solving complex optimization problems having had high multimodality, discontinuity, and noisy landscapes. In general, the objective functions of real-world problems are mostly non-differentiable, noncontinuous, nonlinear, noisy, flat, and multidimensional or comprise many local minima. Traditional optimization techniques are mostly unable to find a quality solution for problems with nonlinearity, multimodality, and high-dimensionality structures.

Over the last two decades, various global optimization methods were developed and successfully tackled various complex optimization and search problems [5, 6]. Evolutionary algorithms (EAs) are much advanced and promising alternatives as compared to gradient-based optimization methods. EAs are population-based techniques and provide good approximated solutions in single simulation unlike the traditional mathematical programming techniques. EAs are population-based metaheuristics [7–9] and are generally divided into nine different groups: the biology-based [10, 11], physics-based [12, 13], social-based [14–17], music-

based [18], chemical-based [19], sport-based [20], mathematics-based [21], swarm-based [22–28], and hybrid methods [29–36]. Genetic algorithm (GA) [37], evolutionary strategy (ES) [38], evolutionary programming (EP) [39, 40], and genetic programming (GP) [41] are the classical paradigms of evolutionary computing [8]. Differential evolution (DE) [42, 43], particle swarm optimization (PSO) [23], ant colony optimization (ACO) [44, 45], cuckoo search (CS) [46], and teaching-learning-based optimization (TLBO) algorithm [47] are the most popular and recently developed EAs [17].

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, whether they are natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Beni and Wang in 1989 [48] in the context of cellular robotic systems. Swarm-intelligence- (SI-) based algorithms, mainly inspired by the behavior of real swarm or insect colonies, have shown improved performance on large-scale optimization and decision-making problems [49]. They can work collectively following some specific rules. The decentralization and self-organization are the main sources of interactions among swarm and agents [26]. SI has received great interests and attention in the collected works. In the communities of optimization, computational intelligence, and computer science, bioinspired algorithms, especially swarm-intelligence-based algorithms, have gained much popularity [50]. Nature-inspired metaheuristic algorithms are now among the most broadly used algorithms for optimization and computational intelligence [28]. For example, the need for cohesion requires individual members to stay in close proximity to one another and, at the same time, to avoid conflicts or collisions with other members of the group; whenever the group moves, all the members of the group should move together.

The use of ensemble strategies in the framework of different existing evolutionary algorithms has attracted the attention of researchers in academia and industrial applications because of their capabilities. Evolutionary computation researchers have shown great interests in employing multiple search operators such as one-point, two-point, and uniform crossover operators, simplex crossover, tournament, ranking, stochastic, uniform sampling selection operator, clearing, crowding, sharing-based niching techniques, adaptive penalty, epsilon, and superiority of feasible constraint handling approaches while solving the complicated test suites of benchmark functions and many important real-world problems have had high complexity, noisy environment, imprecision, uncertainty, and vagueness in their structures [9]. It is quite useful to use different strategies at different stages of population evolutions. Recently, many ensemble strategies were proposed, which benefit from both the availability of diverse approaches and adaptive tuning of associated intrinsic parameters [51]. Many researches have shown the general applicability of the ensemble strategy in solving diverse problems by using different populated optimization algorithms [29, 52, 53]. The performance of the suggested MSIA was evaluated over recently designed benchmark functions for the special

FIGURE 1: Classification of optimization techniques.

session of evolutionary algorithms competition in 2013-IEEE-Congress on Evolutionary Computing (IEEE-CEC'13) [54]. The suggested algorithm has effective and promising optimal solutions for most used test problems. The numerical results provided by our proposed algorithm are much promising in terms of proximity and convergence toward the optimal solution of the problems with large-scale search space region. The main contributions of this paper are as follows:

A multiswarm intelligence algorithm (MSIA) is proposed to improve the exploration and exploitation capabilities of the baseline SI algorithm.

The ant colony optimization (ACO) and firefly algorithm (FA) are employed as constituent algorithms in the framework of the proposed algorithm.

The performance of the MSIA is evaluated over the benchmark functions that are recently designed for the special session of IEEE Congress on Evolutionary Computation (IEEE-CEC-2017) [55].

Extensive experimental results indicate that the proposed MSIA algorithm has performed promisingly in the parlance of evolutionary computing communities.

The rest of the paper is organized as follows: Section 2 introduces the framework of the proposed hybrid swarm-intelligence-based algorithm; Section 3 demonstrates the experimental results and characteristics of the used benchmark functions. Section 4 finally concludes this paper.

2. Multiswarm-Intelligence-Based Algorithm

In the last two decades, the nature-inspired algorithms (NIA) including artificial neural networks (ANN), fuzzy systems (FS), evolutionary computing (EC), and swarm intelligence (SI) have received much attention in both academia and industrial applications [56]. Among them, SI-based methods are mainly inspired by the behavior of real swarm or insect colonies and have shown great success to cope with large-scale optimization and decision-making problems [49]. These algorithms follow collectively some basic and specific rules. The decentralization and self-organization are the main sources of interactions among swarm and agents [57]. The ant colony algorithms (ACO) and bee algorithms, particle swarm optimization (PSO), cuckoo search, bat algorithm, and firefly algorithm offer great advantages over conventional algorithms [28]. We propose a multiswarm-intelligence-based algorithm by integrating ant colony optimization (ACO) and firefly algorithm (FA) to solve unconstrained optimization problems. The flowchart of the proposed algorithm is hereby given in Figure 2. The proposed multiswarm intelligence algorithm as outlined in Algorithm 1 employs different ants while roaming in the search space of the problem at hand. During this roaming, an evolution of ants is performed by integrating ACO and FA, where FA works as a local search to refine the positions found by the ants. Secondly, the performance of FA is improved by reducing the randomization parameter so that it decreases gradually as the optima are approaching. Finally, the performance of suggested

FIGURE 2: Flowchart of the proposed multiswarm-intelligence-based algorithm (MSIA).

methodology as described in the algorithm is examined by using the benchmark functions designed for the special session of evolutionary computation, CEC2013 [54]. The numerical results approximated by the proposed algorithm demonstrated its superiority in finding the global optimal solution for the most used benchmark functions.

3. Benchmark Functions and Experimental Results

Due to the flurry of recently developed EAs, their performances are mainly evaluated by utilizing different test suites of optimization and search problems. In the last few years, several bound constrained test suites of benchmark functions are designed for the competition of EAs in the special sessions of the IEEE Conference of Evolutionary Computation [54]. In the study of this paper, we have used twenty-eight different unconstrained continuous test functions whose characteristics are given in Table 1. The

first five test problems, F_1 to F_5 , are unimodal functions. These types of test functions are offer great challenges and trap an evolutionary algorithm in their local basin of attraction. The problems F_6 to F_{20} are multimodal functions that contain more than one local minimum. The multimodal problems are much difficult to deal with as compared to unimodal functions. The functions F_{21} to F_{28} are composite functions.

3.1. PC Configuration to Perform Experiments. The proposed algorithm was coded in MATLAB 7.8.0 (R2009a) environment and PC with a 3.0 GHz Core m3 Duo processor, 4.00 GB RAM, and 64-bit operating system Windows 7 Professional is used to carry out experiments. We have executed the proposed multiswarm intelligence algorithm (MSIA) and the baseline algorithms, namely, the ant colony optimization (ACO) and firefly algorithm (FA), 25 times indecently with different random seeds by using `rand("state", sum(100 * clock))` in MATLAB environment. To

Initialization Process of the Multiswarm Intelligence Algorithm

- (1) N : number of Swarms/size of the population set;
- (2) n : number of decision variables or dimension of the search space;
- (3) x_l : lower bound of the search space;
- (4) x_u : upper bound of the search space;
- (5) M_g : maximum number of generations;
- (6) g : generation counter;
- (7) $x^i = x_l + (x_u - x_l) \times \text{rand}(N, n)$, $i = 1, 2, \dots, N$
- (8) Evaluate the initial Population, $F(x^i) = [f(x^1), f(x^2), \dots, f(x^N)]$
- (9) Select x^{best} solution among x^i solutions based on their minimum objective function value.

Evolutionary Process of Multi-Swarm Intelligence Based Algorithm

- (10) $g = 1$;
- (11) **while** $g < M_g$ **do**
- (12) **for** $i = 1 : N$ **do**
- (13) **if** $\text{rem}(g, S_g) == 0$ **then**
- (14) $\nabla = |x^i - x^j|$, $i \neq j$
- (15) $\psi = (\xi \times \nabla / N - 1)$
- (16) $y^i = x^i + \psi \times \text{rand } n$
- (17) **else**
- (18) $r^i = (\|x^i - x^j\| / \|x^j - x^u\|)$, $i \neq j$
- (19) $\beta = \beta_o \times \exp^{-\gamma(r^i)^2}$
- (20) $\omega \in [0, 1]$
- (21) $y^i = x^i + \beta \times (x^{\text{best}} - x^i) - \omega \times (x^{\text{worst}} - x^i)$
- (22) **end if**
- (23) Evaluation the offspring Population, $F(y^i) = [f(y^1), f(y^2), \dots, f(y^N)]$
- (24) **if** $f(y^i) < f(x^i)$, $i = 1, 2, \dots, N$ **then**
- (25) $x^i := y^i$
- (26) **else**
- (27) $y^i := x^i$
- (28) **end if**
- (29) Update x^{best} for next generation of population evolution.
- (30) **end for**
- (31) $g = g + 1$
- (32) **end while**

ALGORITHM 1: The framework of the multiswarm intelligence algorithm.

validate the performance of any stochastic-nature-based algorithms, it is intimated in the existing literature of evolutionary computation to execute them at least 10 to 25 times to establish fair judgment and comparison against competitors.

3.2. Parameter Settings in Carried out Experiments. The lower bound $x_l = -100$, upper bound $x_u = 100$ for the used benchmark functions, $N = 100$ is size of initially generated set of solutions, $n = 10, 30$, and 50 are the different dimensions of the search space, $MG = n \times 100$ are maximum number function evaluations, $S_g = 20$, $\gamma = 1$ is the light absorption coefficient, $\beta_o = 0.8$ attraction coefficient, and α is the mutation coefficient, where δ is uniform mutation range.

Tables 2 and 3 show that the proposed MSIA has performed better than ACO and FA while providing the minimum objective function values for the benchmark functions denoted by $F_2, F_3, F_4, F_5, F_{14}, F_{16}, F_{19}, F_{23}$, and F_{27} . However, ACO has discovered better numerical results as compared to MSIA and FA keeping in view the minimum function value got for problem F_{11} . FA has discovered better

solutions against the proposed MSIA and ACO in terms of the minimum values for the problems $F_6, F_7, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{17}, F_{18}, F_{20}, F_{21}, F_{22}, F_{24}, F_{25}, F_{26}$, and F_{28} . Similarly, ACO, FA, and MSIA have obtained almost the same numerical solutions for the problems denoted by F_8 in terms of minimum value. The average values found by MSIA are better than ACO and FA as in the second columns of Tables 2 and 3. The standard deviation values of ACO are better. Similarly, by comparing the maximum values, MSIA has performed better by solving the test problems denoted by $F_2, F_3, F_4, F_5, F_{14}, F_{16}, F_{19}, F_{23}$, and F_{27} . These experimental results demonstrate that the proposed algorithm is much competitive in solving most of the benchmark functions [54].

Tables 4 and 5 clearly exhibit that the proposed MSIA has performed better than ACO and FA in approximating the minimum objective function values by solving the benchmark functions referred to as $F_1, F_5, F_6, F_8, F_9, F_{10}, F_{13}, F_{14}, F_{19}, F_{20}, F_{21}, F_{23}, F_{24}$, and F_{25} . ACO has contributed better results for the problems $F_3, F_7, F_{11}, F_{12}, F_{15}, F_{16}, F_{17}, F_{18}$, and F_{22} . FA has provided better minimum objective function value for the benchmark F_4 . For problems F_{26} and F_{28} ,

TABLE 1: The characteristics of the CEC'13 test functions [54].

	No	Properties	$F_i^* = F_i(x^*)$
UF	1	Sphere function	-1400
	2	Rotated high conditioned elliptic function	-1300
	3	Rotated bent cigar function	-1200
	4	Rotated discus function	-1100
	5	Different power function	-1000
MF	6	Rotated Rosenbrock's function	-900
	7	Rotated Schaffer's F_7 function	-800
	8	Rotated Ackely's function	-700
	9	Rotated Weierstrass function	-600
	10	Rotated Griewank's function	-500
	11	Rastrigin function	-400
	12	Rotated Rastrigin function	-300
	13	Noncontinuous rotated Rastrigin function	-200
	14	Schwefel's function	-100
	15	Rotated Schwefel's function	100
	16	Rotated Katsuura function	200
	17	Lunacek bi-Rastrigin function	300
	18	Rotated Lunacek bi-Rastrigin function	400
	19	Expanded Griewank's plus Rosenbrock's function	500
	20	Expanded Schaffer's F_6 function	600
CF	21	Composition function 1 ($n = 5$, rotated)	700
	22	Composition function 2 ($n = 3$, unrotated)	800
	23	Composition function 3 ($n = 3$, rotated)	900
	24	Composition function 4 ($n = 3$, rotated)	1000
	25	Composition function 5 ($n = 3$, rotated)	1100
	26	Composition function 6 ($n = 5$, rotated)	1200
	27	Composition function 7 ($n = 5$, rotated)	1300
	28	Composition function 8 ($n = 5$, rotated)	1400

PSO and MSIA have got the same result in terms of minimum values. The average values of MSIA are better than ACO and FA as can be shown in the second columns of the above tables. In case of standard deviation, ACO gives better result. Similarly, by comparing the maximum values, MSIA shows better performance for problems F_4 , F_7 , F_9 , F_{18} , F_{19} , F_{20} , F_{23} , F_{24} , F_{28} , and F_{29} . ACO contributes better values in terms of maximum for the problems including F_1 , F_3 , F_5 , F_8 , F_{10} , F_{12} , F_{14} , F_{21} , F_{25} , and F_{26} . FA performed better than ACO and MSIA in terms of maximum values for the problems, F_6 , F_{11} , F_{13} , F_{15} , F_{16} , F_{17} , F_{22} , and F_{27} .

Tables 6 and 7 clearly demonstrate that the proposed MSIA has performed better than ACO and FA in terms of the minimum values by solving the benchmark functions F_1 , F_6 , F_8 , F_9 , F_{13} , F_{14} , F_{17} , F_{19} , F_{20} , F_{21} , F_{23} , F_{24} , F_{25} , F_{26} , F_{27} , and F_{28} . ACO is better than FA and MSIA in terms of minimum values for the problems F_3 , F_4 , F_5 , F_7 , F_{10} , F_{11} , F_{12} , F_{15} , F_{16} , F_{18} , and F_{22} . The average values of the MSIA are better than ACO and FA for the test problems denoted by F_5 , F_6 , F_8 , F_9 , F_{10} , F_{12} , F_{16} , F_{17} , F_{19} , F_{21} , F_{22} , F_{23} , F_{24} , F_{25} , and F_{26} . In case of standard deviation, ACO gives better performance. Similarly, the maximum values obtained by ACO are better for the problems F_5 , F_8 , F_{10} , F_{11} , F_{12} , F_{15} , F_{16} , F_{19} , F_{20} , F_{21} , F_{26} , and F_{27} . FA has shown good performance on the test problems F_1 , F_3 , F_4 , F_9 , F_{13} , F_{14} , F_{18} , F_{24} , and F_{25} . The convergence graphs of the benchmark functions were solved by proposed algorithms in different dimensions including $n = 10, 30, 50$ as shown in Figure 3. The first panel offers the convergence graphs

for the benchmark functions for F_2 to F_5 by solving them each in $n = 10$ decision variables, the second panel for the same problems to solve them $n = 30$ decision variables, and third panel offers the convergence graphs of the benchmark functions for F_2 to F_5 to solve them each in $n = 50$ decision variables or dimension. In each figure, the upward curves demonstrate the evolution in the maximum objective function values of benchmark function, the middle curves depict the evolution in the average objective function value, and the downward curves show the evolution in the minimum objective function value of F_2 to F_5 benchmark functions. The convergence graphs of the benchmark functions were displayed by proposed algorithms in different dimensions including $n = 10, 30$, and 50 as shown in Figure 4. The first panel offers the convergence graphs for F_6 to F_8 by solving them in $n = 10$ dimensions, the second panel presents the convergence graphs of the same problems to solve them in $n = 30$ dimensions, and the third panel offers the convergence graphs for F_6 to F_8 benchmark functions as solved with $n = 50$ decision variables. Overall, the convergence graphs in the aforementioned figures demonstrate that MSIA has tackled most of the test problems promisingly as compared to their competitors in its 25-time independent runs of executions within same given resources and parameter settings. In each figure in each panel, the upward curves demonstrate the evolution in the maximum objective function values of benchmark function, the middle curves depict the evolution in the average objective function value, and the downward curves show the evolution in the

TABLE 2: The experimental results of (a) MSIA, (b) ACO, and (c) FA for CEC'13 benchmark functions in 10 dimensions [54].

Problem	Optimum	Best	Average	Standard deviation	Maximum	Algorithms
F_1	-1400	-1.40000000e+003	-1.28813052e+003	3.83901313e+002	2.67500831e+003	a
		-1.40000000e+003	-1.15749008e+003	1.07195386e+003	4.55466927e+003	b
		-1.40000000e+003	-1.39443074e+003	1.05911384e+002	1.33204120e+003	c
F_2	-1300	-1.30000000e+003	1.00558869e+002	3.34248688e+003	1.01984633e+003	a
		-1.31053585e+003	1.20311183e+002	3.71650087e+003	1.50229253e+003	b
		-1.30240122e+003	1.00650874e+001	3.34011105e+002	1.03514064e+003	c
F_3	-1200	-1.20000000e+003	1.84411229e+003	6.23354197e+003	9.99110936e+004	a
		-1.31510766e+002	2.31193243e+004	6.50115943e+003	2.48403247e+005	b
		-1.29995459e+003	1.18540350e+003	1.61050797e+003	7.63499302e+004	c
F_4	-1100	-1.10000000e+003	1.12780473e+003	6.26400664e-003	1.16009458e+003	a
		4.11467285e+004	1.11467285e+004	2.47506343e+5	4.11467285e+004	b
		-1.10000000e+003	1.65098334e+003	1.45403194e+003	1.86066330e+004	c
F_5	-1000	-1.00000000e+003	1.65169743e+003	6.32203970e+003	1.14679794e+005	a
		-1.00000000e+003	-8.90435476e+002	5.25039922e+002	3.99696948e+003	b
		-1.00000000e+003	-9.99017949e+002	1.99880005e+001	-4.21267907e+002	c
F_6	-900	-1.40000000e+003	1.74778253e+003	5.98501382e+003	1.07202661e+005	a
		-8.94435996e+002	-8.72043240e+002	9.43596267e+001	-4.29479950e+001	b
		-8.99971565e+002	-8.99359770e+002	6.00877221e+000	-7.22793586e+002	c
F_7	-800	-1.40000000e+003	1.74757754e+003	6.32949555e+003	1.06572975e+005	a
		-7.99256478e+002	-7.53886898e+002	6.62008126e+001	-4.39710456e+002	b
		-8.00000000e+002	-7.99490674e+002	6.06214778e+000	-6.93335202e+002	c
F_8	-700	-1.40000000e+003	1.77042499e+003	6.21341629e+003	1.10342739e+005	a
		-1.40000000e+003	-1.17255767e+003	1.00591243e+003	5.03660790e+003	b
		-1.40000000e+003	-1.39333836e+003	1.37246794e+002	2.48848299e+003	c
F_9	-600	-1.40000000e+003	1.63539741e+003	5.79958004e+003	1.02497124e+005	a
		-6.79760366e+002	-6.79685333e+002	3.41686898e-002	-6.79222141e+002	b
		-6.79892570e+002	-6.79846262e+002	4.32771150e-002	-6.79577360e+002	c
F_{10}	-500	-1.40000000e+003	1.81847280e+003	6.63718254e+003	1.02569292e+005	a
		-5.92651099e+002	-5.91792871e+002	3.09488316e-001	-5.88815678e+002	b
		-5.99358333e+002	-5.99287888e+002	3.50839046e-001	-5.91626684e+002	c
F_{11}	-400	-1.40000000e+003	1.82639194e+003	6.40390942e+003	9.39110015e+004	a
		-4.99597235e+002	-4.18769343e+002	2.03331451e+002	3.02821266e+002	b
		-4.99953263e+002	-4.99163416e+002	1.17780480e+001	-1.66277831e+002	c
F_{12}	-300	-1.40000000e+003	2.04024682e+003	6.43914792e+003	1.08141375e+005	a
		-3.74840231e+002	-3.59247046e+002	1.98618374e+001	-2.62805243e+002	b
		-4.00000000e+002	-3.99487961e+002	4.70209205e+000	-3.18882349e+002	c
F_{13}	-200	-1.40000000e+003	1.61912036e+003	5.97109732e+00	9.13305274e+004	a
		-2.81939856e+002	-2.60638895e+002	2.09195180e+001	-1.65501622e+002	b
		-2.96020164e+002	-2.95516805e+002	4.98604369e+000	-1.99035788e+002	c
F_{14}	-100	-1.41000000e+003	-1.38285100e+003	1.26343726e+002	2.37763536e+003	a
		-1.80661472e+002	-1.53623927e+002	2.15069087e+001	-9.27553100e+000	b
		-1.99005041e+002	-1.98527872e+002	4.31448924e+000	-1.04821070e+002	c

TABLE 3: The experimental results of (a) MSIA, (b) ACO, and (c) FA by solving the CEC'13 benchmark functions in 10 dimensions [54].

Problem	Optimum	Best	Average	Standard deviation	Maximum	Algorithms
F_{15}		$-1.05787441e+003$	$-4.27478554e+002$	$5.32520087e+002$	$4.51919784e+003$	a
		$1.29020039e+003$	$1.43497411e+003$	$1.52524081e+002$	$2.42391695e+003$	b
		$-6.97608460e+001$	$-5.56706985e+001$	$9.83861302e+001$	$1.59094345e+003$	c
F_{16}		$-1.41000000e+003$	$-1.38038036e+003$	$1.51705325e+002$	$3.82948621e+003$	a
		$1.37302013e+003$	$1.81027837e+003$	$1.99350254e+001$	$1.82017859e+003$	b
		$2.60622099e+003$	$7.38929816e+004$	$8.28530915e+005$	$1.68940889e+007$	c
F_{17}		$-1.07775351e+003$	$-7.87135563e+002$	$5.41473038e+002$	$4.68688680e+003$	a
		$2.01161840e+002$	$2.01275691e+002$	$3.26119574e-001$	$2.03212837e+002$	b
		$-1.19999683e+003$	$1.34713183e+007$	$2.61331526e+008$	$6.96041088e+009$	c
F_{18}		$-1.00000000e+001$	$2.97380129e+006$	$5.27084138e+006$	$7.70958077e+007$	a
		$3.38294199e+002$	$3.48198354e+002$	$2.52233087e+001$	$4.52178904e+002$	b
		$-1.10000000e+003$	$-8.53652774e+002$	$1.41482529e+003$	$1.45950316e+004$	c
F_{19}		$-7.71635593e+002$	$-5.58888645e+002$	$2.76514199e+002$	$4.56991524e+003$	a
		$4.38988097e+002$	$4.56925962e+002$	$3.70758564e+001$	$7.12211302e+002$	b
		$-1.00000000e+003$	$-9.98683610e+002$	$2.96836196e+001$	$-1.60926967e+002$	c
F_{20}		$-1.41000000e+003$	$-1.38210398e+003$	$9.76743330e+001$	$5.09148005e+002$	a
		$5.02238254e+002$	$5.93002146e+002$	$4.51529039e+002$	$3.20047624e+003$	b
		$-8.99935443e+002$	$-8.99208513e+002$	$5.49981592e+000$	$-7.64788173e+002$	c
F_{21}		$-7.34688317e+002$	$-6.55445684e+002$	$4.26933534e+002$	$4.19502288e+003$	a
		$6.03841405e+002$	$6.04103425e+002$	$1.21304155e-001$	$6.04659232e+002$	b
		$-8.00000000e+002$	$-7.99515370e+002$	$5.41047502e+000$	$-6.89237686e+002$	c
F_{22}		$-1.41000000e+003$	$-1.37613278e+003$	$1.47628379e+002$	$2.70449872e+003$	a
		$1.10019387e+003$	$1.11529246e+003$	$6.39077139e+001$	$1.94034991e+003$	b
		$-6.79882628e+002$	$-6.79814730e+002$	$2.70757318e-002$	$-6.79706209e+002$	c
F_{23}		$-7.79223446e+002$	$-5.87923497e+002$	$5.29115496e+002$	$6.36559812e+003$	a
		$2.71131924e+003$	$2.80638128e+003$	$7.47686019e+001$	$3.35948750e+003$	b
		$-5.99730286e+002$	$-5.99653895e+002$	$4.48506744e-001$	$-5.90984947e+002$	c
F_{24}		$-1.41000000e+003$	$-1.38722065e+003$	$1.35985509e+002$	$3.96291901e+003$	a
		$2.81293009e+003$	$3.03118853e+003$	$5.83771758e+001$	$3.20391460e+003$	b
		$-4.99955742e+002$	$-4.99294862e+002$	$1.08319919e+001$	$-2.62831236e+002$	c
F_{25}		$-1.00003268e+003$	$-6.28496611e+002$	$5.20374802e+002$	$8.18200464e+003$	a
		$1.22181090e+003$	$1.22284625e+003$	$1.61905418e+000$	$1.23616915e+003$	b
		$-4.00000000e+002$	$-3.99480200e+002$	$4.13754005e+000$	$-3.08463035e+002$	c
F_{26}		$-1.41000000e+003$	$-1.37818962e+003$	$1.46117712e+002$	$2.51374993e+003$	a
		$1.32080243e+003$	$1.32256986e+003$	$1.04244644e+000$	$1.33641433e+003$	b
		$-2.97015123e+002$	$-2.96424650e+002$	$4.86448955e+000$	$-1.94232316e+002$	c
F_{27}		$-7.55127444e+002$	$-5.31346642e+002$	$5.59678487e+002$	$6.24369518e+003$	a
		$-1.40000000e+003$	$-1.21823184e+003$	$6.99597007e+002$	$2.29375556e+003$	b
		$-1.97015123e+002$	$-1.96411417e+002$	$4.60497004e+000$	$-7.09293980e+001$	c
F_{28}		$-1.41000000e+003$	$-1.37461266e+003$	$1.46565437e+002$	$2.11257128e+003$	a
		$7.79045518e+006$	$1.92214598e+007$	$1.24536123e+007$	$5.64788737e+007$	b
		$-8.46930600e+001$	$-6.97385530e+001$	$1.11122780e+002$	$1.69479719e+003$	c

TABLE 4: The experimental results of (a) MSIA, (b) ACO, and (c) FA by solving the CEC'13 in 30 dimensions [54].

Problem	Best	Average	Standard deviation	Maximum	Algorithms
F_1	$-1.00000000e+001$	$-1.34847601e+004$	$4.08563270e+003$	$7.14781379e+004$	a
	$-1.40000000e+003$	$-1.31762841e+003$	$6.24857337e+002$	$5.92866180e+003$	b
	$-1.40000000e+003$	$-1.39850118e+003$	$4.75975593e+001$	$4.92018509e+002$	c
F_2	$-1.00000000e+001$	$1.34847601e+004$	$4.08563270e+003$	$7.14781379e+004$	a
	$5.11053585e+006$	$1.70311183e+007$	$3.71650087e+006$	$3.50229253e+007$	b
	$-1.22998756e+003$	$2.30092533e+004$	$3.40438747e+005$	$1.23150007e+007$	c
F_3	$-1.00000000e+001$	$1.29269959e+004$	$4.09476108e+003$	$7.75773844e+004$	a
	$-1.19999489e+003$	$1.12802551e+009$	$7.55617940e+009$	$8.33553469e+010$	b
	$-1.19999029e+003$	$4.24092267e+006$	$1.35141321e+008$	$6.10608129e+009$	c
F_4	$-1.00000000e+001$	$1.26689291e+004$	$4.32681461e+003$	$6.93749063e+004$	a
	$4.62070602e+004$	$5.85905504e+004$	$4.32263932e-009$	$7.01655272e+004$	b
	$-1.10000000e+003$	$-1.02665335e+003$	$7.40976376e+002$	$1.24856684e+004$	c
F_5	$-1.00000000e+001$	$1.38374537e+004$	$3.94289518e+003$	$6.73494023e+004$	a
	$-1.00000000e+003$	$-9.69829937e+002$	$1.97609298e+002$	$1.70842420e+003$	b
	$-1.00000000e+003$	$-9.99693929e+002$	$1.00936173e+001$	$-5.07286413e+002$	c
F_6	$-1.00000000e+001$	$1.04357828e+004$	$3.97031629e+003$	$7.52279597e+004$	a
	$-8.98211405e+002$	$-8.89173945e+002$	$4.91621529e+001$	$-1.52687789e+002$	b
	$-8.99992095e+002$	$-8.99652152e+002$	$3.12308019e+000$	$-7.44410826e+002$	c
F_7	$-1.00000000e+001$	$1.44303050e+004$	$3.09999749e+003$	$7.62766568e+004$	a
	$-7.99999570e+002$	$-7.79560063e+002$	$5.40992509e+001$	$-4.30716664e+002$	b
	$-8.00000000e+002$	$-7.99803559e+002$	$3.39121331e+000$	$-6.65980780e+002$	c
F_8	$-1.00000000e+001$	$1.33612319e+004$	$3.27656564e+003$	$7.80617165e+004$	a
	$-1.40000000e+003$	$-1.33502027e+003$	$4.82036178e+002$	$6.63206765e+003$	b
	$-6.79904212e+002$	$-6.79884280e+002$	$3.38983850e-002$	$-6.79611291e+002$	c
F_9	$-1.00000000e+001$	$1.14785811e+004$	$3.53367480e+003$	$6.56185803e+004$	a
	$-6.79910546e+002$	$-6.79663617e+002$	$3.52873906e-002$	$-6.79101411e+002$	b
	$-5.99163443e+002$	$-5.99141005e+002$	$2.49386486e-001$	$-5.92168717e+002$	c
F_{10}	$-1.00000000e+001$	$1.37390386e+004$	$3.72354347e+003$	$6.09673611e+004$	a
	$-5.91820960e+002$	$-5.91449793e+002$	$3.70460630e-001$	$-5.89231846e+002$	b
	$-4.99977814e+002$	$-4.99721994e+002$	$7.57832193e+000$	$-1.38027458e+002$	c
F_{11}	$-1.00000000e+001$	$1.34848754e+004$	$3.63155978e+003$	$8.36727298e+004$	a
	$-4.99475498e+002$	$-4.79337941e+002$	$8.14000795e+001$	$4.29514179e+002$	b
	$-3.99005041e+002$	$-3.98830318e+002$	$2.95481402e+000$	$-2.85890624e+002$	c
F_{12}	$-1.00000000e+001$	$1.36075496e+004$	$3.82776022e+003$	$8.29620099e+004$	a
	$-3.75129327e+002$	$-3.67133358e+002$	$1.37839005e+001$	$-2.36996315e+002$	b
	$-2.96020169e+002$	$-2.95865600e+002$	$2.41752229e+000$	$-2.13057107e+002$	c
F_{13}	$-1.00000000e+001$	$1.23359046e+004$	$4.24524249e+003$	$7.07813201e+004$	a
	$-2.79504176e+002$	$-2.73450147e+002$	$1.59304444e+001$	$-1.28836347e+002$	b
	$-1.97107646e+002$	$-1.96895621e+002$	$2.36431415e+000$	$-1.01814893e+002$	c
F_{14}	$-1.00000000e+001$	$1.31945555e+004$	$4.83006304e+003$	$4.55239693e+004$	a
	$-1.77607239e+002$	$-1.72159776e+002$	$1.66448427e+001$	$-1.90275590e+001$	b
	$-8.82953840e+001$	$-8.37815952e+001$	$5.75227562e+001$	$1.70271789e+003$	c

TABLE 5: The experimental results of (a) MSIA, (b) ACO, and (c) FA by solving the CEC'13 in 30 dimensions [54].

Problem	Best	Average	Standard deviation	Maximum	Algorithms
F_{15}	$-1.00000000e+001$	$1.17678517e+004$	$4.19260801e+003$	$6.99119354e+004$	a
	$1.43222909e+003$	$1.54101244e+003$	$9.23279003e+001$	$2.13679767e+003$	b
	$3.91260701e+002$	$4.02599646e+002$	$5.30758844e+001$	$1.60528931e+003$	c
F_{16}	$-1.00000000e+001$	$1.36000342e+004$	$4.41649078e+003$	$7.47814557e+004$	a
	$1.42113678e+003$	$1.64027890e+003$	$8.49407985e+001$	$2.26938740e+003$	b
	$2.00459133e+002$	$2.00514033e+002$	$9.39412344e-002$	$2.01269803e+002$	c
F_{17}	$-1.00000000e+001$	$1.05879624e+004$	$4.09410778e+003$	$7.21491129e+004$	a
	$2.00689911e+002$	$2.00962796e+002$	$2.15379730e-001$	$2.02372908e+002$	b
	$3.10965091e+002$	$3.11851445e+002$	$4.17063288e+000$	$4.50075796e+002$	c
F_{18}	$-1.00000000e+001$	$1.14961501e+004$	$4.67131822e+003$	$7.21145444e+004$	a
	$3.29079929e+002$	$3.41977697e+002$	$2.44899705e+001$	$7.20851725e+002$	b
	$4.10780091e+002$	$4.12501042e+002$	$5.36744967e+000$	$5.98143774e+002$	c
F_{19}	$-1.00000000e+001$	$1.28019919e+004$	$4.19423733e+003$	$6.22023871e+004$	a
	$4.33379354e+002$	$4.43469268e+002$	$2.42019177e+001$	$7.20032703e+002$	b
	$5.00402182e+002$	$5.00819260e+002$	$4.11858268e+000$	$7.02659326e+002$	c
F_{20}	$-1.00000000e+001$	$1.27712941e+004$	$4.14302012e+003$	$7.09567962e+004$	a
	$5.01645112e+002$	$5.66417564e+002$	$6.60883380e+002$	$9.19019400e+003$	b
	$6.01283828e+002$	$6.01309059e+002$	$4.33502708e-002$	$6.04538503e+002$	c
F_{21}	$-1.00000000e+001$	$1.21557177e+004$	$4.43525864e+003$	$8.19245032e+004$	a
	$6.03858981e+002$	$6.03913568e+002$	$1.02479998e-001$	$6.04932488e+002$	b
	$9.00000000e+002$	$9.00531224e+002$	$4.24046971e+000$	$1.29892286e+003$	c
F_{22}	$-1.00000000e+001$	$1.35325013e+004$	$4.37606868e+003$	$7.89091921e+004$	a
	$1.10019387e+003$	$1.10667394e+003$	$5.01088911e+001$	$1.65340183e+003$	b
	$8.23273710e+002$	$8.28211025e+002$	$6.35219641e+001$	$2.71638792e+003$	c
F_{23}	$-1.00000000e+001$	$1.25997912e+004$	$3.66188342e+003$	$8.42953291e+004$	a
	$2.37530690e+003$	$2.59453594e+003$	$7.39652314e+001$	$3.32431330e+003$	b
	$9.68507849e+002$	$9.75416587e+002$	$2.07440950e+001$	$2.93823230e+003$	c
F_{24}	$-1.00000000e+001$	$1.27023976e+004$	$3.34330239e+003$	$5.65528731e+004$	a
	$2.45776158e+003$	$2.47866550e+003$	$8.14277485e+001$	$3.73865930e+003$	b
	$1.12532990e+003$	$1.12559980e+003$	$7.80349602e-001$	$1.21437056e+003$	c
F_{25}	$-1.00000000e+001$	$1.32972583e+004$	$4.04146240e+003$	$5.93783435e+004$	a
	$1.22116690e+003$	$1.22322675e+003$	$1.24341107e+000$	$1.23724897e+003$	b
	$1.20781404e+003$	$1.20810635e+003$	$7.04658281e-001$	$1.31771006e+003$	c
F_{26}	$-1.00000000e+001$	$1.42197213e+004$	$4.42812498e+003$	$8.82240668e+004$	a
	$1.31935053e+003$	$1.32228110e+003$	$1.02488069e+000$	$1.33308473e+003$	b
	$1.30397984e+003$	$1.30430397e+003$	$7.93146115e-002$	$1.40422272e+003$	c
F_{27}	$-1.00000000e+001$	$1.35961931e+004$	$4.55021230e+003$	$6.87856577e+004$	a
	$-1.40000000e+003$	$-1.32567902e+003$	$5.89965900e+002$	$4.90025229e+003$	b
	$1.60000000e+003$	$1.60044806e+003$	$6.63397786e+000$	$1.99305962e+003$	c
F_{28}	$-1.00000000e+001$	$1.27913603e+004$	$3.67355028e+003$	$8.33154374e+004$	a
	$2.46276416e+006$	$9.67035007e+006$	$3.45689813e+006$	$1.89713016e+007$	b
	$1.50000000e+003$	$1.50123431e+003$	$2.08826897e+001$	$2.49319245e+003$	c

TABLE 6: The experimental results of (a) MSIA, (b) ACO, and (c) FA for CEC'13 benchmark functions in 50 dimensions [54].

Problem	Best	Average	Standard deviation	Maximum	Algorithms
F_1	$-1.40000000e+003$	$3.47354192e+003$	$8.27740523e+003$	$1.05496319e+005$	a
	$-1.40000000e+003$	$-1.35428971e+003$	$4.70646934e+002$	$6.25649642e+003$	b
	$-1.40000000e+003$	$-1.39910071e+003$	$3.68737654e+001$	$4.92018509e+002$	c
F_2	$-1.41000000e+003$	$1.84206047e+003$	$6.15229975e+003$	$1.00498786e+005$	a
	$1.58838476e+006$	$5.58502270e+006$	$1.12878507e+006$	$6.68406976e+006$	b
	$-1.29205225e+003$	$1.41883572e+004$	$3.38098501e+005$	$1.58121852e+007$	c
F_3	$-1.41000000e+003$	$1.81535300e+003$	$6.46557329e+003$	$1.05226948e+005$	a
	$-1.19999370e+003$	$4.84702158e+008$	$4.52354730e+009$	$4.13826791e+010$	b
	$-1.19990718e+003$	$3.82746531e+006$	$1.45131998e+008$	$7.77082609e+009$	c
F_4	$-1.41000000e+003$	$1.72163802e+003$	$6.14342458e+003$	$1.09589953e+005$	a
	$3.70514417e+004$	$4.91623674e+004$	$8.55923168e+003$	$6.31998448e+004$	b
	$-1.10000000e+003$	$-1.05559019e+003$	$6.69675768e+002$	$1.87691603e+004$	c
F_5	$-1.41000000e+003$	$1.71817869e+003$	$5.85107760e+003$	$9.79306370e+004$	a
	$-1.00000000e+003$	$-9.85375133e+002$	$1.03364586e+002$	$3.21895510e+001$	b
	$-1.00000000e+003$	$-9.99689256e+002$	$1.52449018e+001$	$3.11424940e+001$	c
F_6	$-1.41000000e+003$	$1.99558869e+003$	$6.34248688e+003$	$1.01974633e+005$	a
	$-8.99378954e+002$	$-8.91979873e+002$	$4.34885611e+001$	$-2.80863653e+002$	b
	$-8.90187577e+002$	$-8.90104659e+002$	$3.02193637e+000$	$-7.29051446e+002$	c
F_7	$-1.41000000e+003$	$1.54987687e+003$	$5.83338980e+003$	$8.44106665e+004$	a
	$-8.00000000e+002$	$-7.88309504e+002$	$4.62373905e+001$	$-5.73943541e+002$	b
	$-8.00000000e+002$	$-7.99909427e+002$	$2.05901988e+000$	$-7.04307186e+002$	c
F_8	$-1.41000000e+003$	$1.83411229e+003$	$6.23354197e+003$	$9.99010936e+004$	a
	$-6.79738489e+002$	$-6.79704207e+002$	$3.88206288e-002$	$-6.79388983e+002$	b
	$-6.79943515e+002$	$-6.79889283e+002$	$2.01173471e-002$	$-6.79477819e+002$	c
F_9	$-1.41000000e+003$	$1.75086618e+003$	$5.50835728e+003$	$9.66542362e+004$	a
	$-5.94391734e+002$	$-5.93477219e+002$	$1.86712018e-001$	$-5.88456073e+002$	b
	$-5.99898297e+002$	$-5.99881850e+002$	$1.20806497e-001$	$-5.92019062e+002$	c
F_{10}	$-1.41000000e+003$	$1.81780473e+003$	$6.26400664e+003$	$1.11999458e+005$	a
	$-4.99574735e+002$	$-4.87583068e+002$	$7.67688548e+001$	$3.58471374e+002$	b
	$-4.99967994e+002$	$-4.99789635e+002$	$6.69373644e+000$	$-8.19969150e+001$	c
F_{11}	$-1.41000000e+003$	$2.00976090e+003$	$6.69847722e+003$	$1.06053345e+005$	a
	$-3.79849409e+002$	$-3.69899276e+002$	$1.22865847e+001$	$-2.22256810e+002$	b
	$-1.40000000e+003$	$-1.39875811e+003$	$5.40784967e+001$	$1.82944608e+003$	c
F_{12}	$-1.41000000e+003$	$1.64169743e+003$	$6.32203970e+003$	$1.14669794e+005$	a
	$-2.79898809e+002$	$-2.70056788e+002$	$1.15344465e+001$	$-1.24564014e+002$	b
	$-1.28034976e+003$	$1.27153310e+004$	$1.10178983e+005$	$3.51107348e+006$	c
F_{13}	$-1.41000000e+003$	$1.63715788e+003$	$6.16915673e+003$	$1.00990637e+005$	a
	$-1.79369746e+002$	$-1.69530534e+002$	$1.36496165e+001$	$-4.88058640e+001$	b
	$-1.40000000e+003$	$-1.39910071e+003$	$3.68737654e+001$	$4.92018509e+002$	c
F_{14}	$-1.41000000e+003$	$1.73778253e+003$	$5.98501382e+003$	$1.07192661e+005$	a
	$1.28527965e+003$	$1.52502262e+003$	$1.00139634e+002$	$1.88113886e+003$	b
	$-1.29205225e+003$	$1.41883572e+004$	$3.38098501e+005$	$1.58121852e+007$	c

TABLE 7: The experimental results gathered by (a) MSIA, (b) ACO, and (c) FA by solving the CEC'13 in 50 dimensions [54].

Problem	Best	Average	Standard deviation	Maximum	Algorithms
F_{15}	$-1.41000000e+003$	$1.66138664e+003$	$5.80597454e+003$	$1.04737568e+005$	a
	$1.44383602e+003$	$1.68507771e+003$	$1.20240236e+002$	$2.31595933e+003$	b
	$-1.19990718e+003$	$3.82746531e+006$	$1.45131998e+008$	$7.77082609e+009$	c
F_{16}	$-1.41000000e+003$	$1.73757754e+003$	$6.32949555e+003$	$1.06562975e+005$	a
	$2.00748304e+002$	$2.00994974e+002$	$1.32241987e-001$	$2.01380946e+002$	b
	$-1.10000000e+003$	$-1.05559019e+003$	$6.69675768e+002$	$1.87691603e+004$	c
F_{17}	$-1.41000000e+003$	$1.65456868e+003$	$5.58184222e+003$	$9.12960968e+004$	a
	$3.28546329e+002$	$3.39782429e+002$	$1.46231581e+001$	$5.23035189e+002$	b
	$-1.00000000e+003$	$-9.99689256e+002$	$1.52449018e+001$	$3.11424940e+001$	c
F_{18}	$-1.41000000e+003$	$1.76042499e+003$	$6.21341629e+003$	$1.10332739e+005$	a
	$4.23505035e+002$	$4.38285056e+002$	$1.90225565e+001$	$6.95578654e+002$	b
	$-8.90187577e+002$	$-8.90104659e+002$	$3.02193637e+000$	$-7.29051446e+002$	c
F_{19}	$-1.41000000e+003$	$1.85396888e+003$	$6.17159111e+003$	$1.03570243e+005$	a
	$5.01667133e+002$	$5.80533254e+002$	$9.81067479e+002$	$1.58505896e+004$	b
	$-8.00000000e+002$	$-7.99909427e+002$	$2.05901988e+000$	$-7.04307186e+002$	c
F_{20}	$-1.41000000e+003$	$1.62539741e+003$	$5.79958004e+003$	$1.02487124e+005$	a
	$6.03679642e+002$	$6.03785422e+002$	$9.66042719e-002$	$6.04406379e+002$	b
	$-1.40000000e+003$	$-1.39910071e+003$	$3.68737654e+001$	$4.92018509e+002$	c
F_{21}	$-1.41000000e+003$	$1.64131527e+003$	$6.14062123e+003$	$1.05186050e+005$	a
	$1.10019387e+003$	$1.10324808e+003$	$2.94743305e+001$	$1.49762916e+003$	b
	$-1.29205225e+003$	$1.41883572e+004$	$3.38098501e+005$	$1.58121852e+007$	c
F_{22}	$-1.41000000e+003$	$1.80847280e+003$	$6.63718254e+003$	$1.02559292e+005$	a
	$2.37478548e+003$	$2.54293073e+003$	$9.12159912e+001$	$3.00312732e+003$	b
	$-1.19990718e+003$	$3.82746531e+006$	$1.45131998e+008$	$7.77082609e+009$	c
F_{23}	$-1.41000000e+003$	$1.69081569e+003$	$5.78391372e+003$	$9.43656508e+004$	a
	$2.28938148e+003$	$2.62333094e+003$	$8.80245575e+001$	$3.44774393e+003$	b
	$-1.10000000e+003$	$-1.05559019e+003$	$6.69675768e+002$	$1.87691603e+004$	c
F_{24}	$-1.41000000e+003$	$1.81639194e+003$	$6.40390942e+003$	$9.39010015e+004$	a
	$1.22091464e+003$	$1.22195198e+003$	$1.01433920e+000$	$1.23395762e+003$	b
	$-1.00000000e+003$	$-9.99689256e+002$	$1.52449018e+001$	$3.11424940e+001$	c
F_{25}	$-1.41000000e+003$	$1.86194201e+003$	$6.54569682e+003$	$9.52988225e+004$	a
	$1.31791601e+003$	$1.31863180e+003$	$9.99567055e-001$	$1.34051854e+003$	b
	$-8.90187577e+002$	$-8.90104659e+002$	$3.02193637e+000$	$-7.29051446e+002$	c
F_{26}	$-1.41000000e+003$	$2.03024682e+003$	$6.43914792e+003$	$1.08131375e+005$	a
	$-1.40000000e+003$	$-1.35739006e+003$	$4.27324154e+002$	$4.72562605e+003$	b
	$-8.00000000e+002$	$-7.99909427e+002$	$2.05901988e+000$	$-7.04307186e+002$	c
F_{27}	$-1.41000000e+003$	$1.77597903e+003$	$6.06248056e+003$	$8.06134908e+004$	a
	$-1.40000000e+003$	$-1.35296994e+003$	$5.19958849e+002$	$8.55895569e+003$	b
	$-6.79943515e+002$	$-6.79889283e+002$	$2.01173471e-002$	$-6.79477819e+002$	c
F_{28}	$-1.41000000e+003$	$1.60912036e+003$	$5.97109732e+003$	$9.13205274e+004$	a
	$2.19090607e+006$	$1.14588392e+007$	$1.17502885e+007$	$4.93585298e+007$	b
	$-5.99898297e+002$	$-5.99881850e+002$	$1.20806497e-001$	$-5.92019062e+002$	c

minimum objective function value of F_6 to F_8 benchmark functions. The convergence graphs are exhibited by the proposed algorithms to solve each benchmark function in different dimensions including $n = 10, 30$, and 50 as visualized in Figure 5. The first panel represents the convergence graphs for F_9 to F_{11} benchmark functions as solved with $n = 10$ decision variables by the proposed algorithm, the second panel is for the same problems with $n = 30$ decision variables, and third panel is for the F_9 to F_{11} benchmark functions as solved in $n = 50$

dimensions. In Figure 5, for each panel, the upward curves demonstrate the evolution in the maximum objective function values of benchmark function, the middle curves depict the evolution in the average objective function value, and the downward curves show the evolution in the minimum objective function value of F_9 to F_{11} benchmark functions. These figures clearly demonstrate that overall MSIA has tackled most of the test problems efficiently as compared to its competitors in 25-time independent runs of simulations within allowed resources.

(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 3: Continued.

FIGURE 3: The evolution in maximum, average, and minimum function values displayed by the proposed MSIA solving F_2 to F_5 benchmark functions [54]. The first panel is with respect to $n = 10$, the second panel is with respect to $n = 30$, and third panel includes the convergence graphs for $n = 50$ decision variables.

FIGURE 4: The evolution in maximum, average, and minimum function values displayed by the proposed MSIA solving the benchmark functions from F_6 to F_8 [54]. The first panel shows the convergence graph with respect to $n = 10$, the second panel is with respect to $n = 30$, and the third panel is with respect to $n = 50$ decision variables.

FIGURE 5: The evolution in maximum, average, and minimum function values displayed by the proposed MSIA solving F_9 to F_{11} benchmark functions [54]. The first panel is with respect to $n = 10$, the second panel is with respect to $n = 30$, and the third panel is with respect to $n = 50$ decision variables.

4. Conclusions

In the last two decades, evolutionary algorithms based on swarm intelligence were developed to solve various test suites of benchmark functions and real-world problems. Swarm-intelligence-based approaches perform their search process with mutual interactions. They are mainly inspired from the collective and cooperative movement of ant colonies, bird flocking, hawks hunting, animal herding, bacterial growth, fish schooling, and microbial intelligence. In the firefly algorithm, fireflies comply with the brightness of the individual firefly in the exploration process to effectively search the decision space of the given problem. ACO is another algorithm of the most important SI-based algorithms inspired by the foraging behavior of real ant colonies. In this paper, we have proposed a multiswarm-intelligence-based algorithm (MSIA) by employing the ant colony optimization (ACO) and firefly algorithm (FA) as constituent algorithms for population evolution. The proposed algorithm is tested upon using the recently designed benchmark functions for the special session of 2013 IEEE Congress on Evolutionary Computation. The experimental results provided by MSIA are promising in terms of proximity and diversity.

In the future, we intend to investigate the performance of the suggested algorithm by using the latest IEEE-CEC series benchmark functions and real-world problems.

Data Availability

The data used in this article are freely available upon citing the article. The MATLAB code of the benchmark functions and related data are available at <https://www3.ntu.edu.sg/home/epsnugan/>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors thankfully acknowledge the support of the Qatar National Library, Qatar, and Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Education City, Qatar Foundation, Doha, Qatar.

References

- [1] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, Princeton, NJ, USA, 2006.
- [2] S.-C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms*, Prentice-Hall, Upper Saddle River, NJ, USA, 1993.
- [3] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, NY, USA, 1968.
- [4] R. E. Miller, *Optimization: Foundations and Applications*, John Wiley & Sons, London, UK, 1999.
- [5] L. Decastro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1–36, 2007.
- [6] S. M. Elsayed, R. A. Sarker, and D. L. Essam, *A Genetic Algorithm for Solving the CEC'2013 Competition Problems on Real-Parameter Optimization*, IEEE Congress on Evolutionary Computation, London, UK, 2013.
- [7] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York, NY, USA, 1996.
- [8] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer Publishing Company, Berlin, Germany, 2nd edition, 2015.
- [9] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of populations for global numerical optimization," *OPSEARCH*, vol. 46, no. 2, pp. 184–213, 2009.
- [10] S. Akyol and B. Alatas, "Plant intelligence based metaheuristic optimization algorithms," *Artificial Intelligence Review*, vol. 47, no. 4, pp. 417–462, 2017.
- [11] B. I. S. Muhammad Sulaiman, Abdellah Salhi, and O. B. Kirikchi, "A plant propagation algorithm for constrained engineering optimisation problems," *Mathematical Problems in Engineering*, vol. 10, 2014.
- [12] S. Hong, D. Han, K. Cho, J. S. Shin, and J. Noh, "Physics-based full-body soccer motion control for dribbling and shooting," *ACM Trans. Graph.*, vol. 38, no. 4, 2019.
- [13] N. Siddique and H. Adeli, "Physics-based search and optimization: inspirations from nature," *Expert Systems*, vol. 33, no. 6, pp. 607–623, 2016.
- [14] F. Johari, A. Zain, N. Mustafa, and A. Udin, "Firefly algorithm for optimization problem," *Applied Mechanics and Materials*, vol. 421, 2013.
- [15] L. Li and F. Liu, *Group search optimization for applications in structural design*, Springer, New York, NY, USA, 2011.
- [16] X.-S. Yang, *A new metaheuristic bat-inspired algorithm*, Springer, Berlin, Germany, 2010.
- [17] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*, Springer Science & Business Media, Berlin, Germany, 2010.
- [18] J. H. Jeong and C. W. Ahn, "Automatic evolutionary music composition based on multi-objective genetic algorithm," 2015.
- [19] F. T. Silva, M. X. Silva, and J. C. Belchior, "A new genetic algorithm approach applied to atomic and molecular cluster studies," *Frontiers in Chemistry*, vol. 7, p. 707, 2019.
- [20] L. While and G. Kendall, *Scheduling the English football league with a multi-objective evolutionary algorithm*, Springer International Publishing, Berlin, Germany, 2014.
- [21] H. Mühlenbein and T. Mahnig, *Mathematical Analysis of Evolutionary Algorithms*, Springer US, Boston, MA, USA, 2002.
- [22] C. Blum, "Ant colony optimization: introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353–373, 2005.
- [23] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," 1995.
- [24] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2–3, pp. 235–306, 2002.
- [25] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, *The Bees Algorithm*, Cardiff University, London, UK, 2005.
- [26] J. R. Rohan, C. N. Prasad, and P. Sadiq, "An Introduction to the Collective Behaviour of Swarm Intelligence," 2017.
- [27] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," 1998.

- [28] X.-S. Yang, "Swarm intelligence based Algorithms: a critical analysis," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17–28, 2014.
- [29] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: methodologies, architectures, and reviews," 2007.
- [30] W. Khan, *Hybrid Multiobjective Evolutionary Algorithm Based on Decomposition*, Department of Mathematical Sciences, University of Essex, Colchester, UK, 2012.
- [31] W. K. Mashwani, "Hybrid multiobjective evolutionary algorithms: a survey of the state-of-the-art," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 374–392, 2011.
- [32] W. K. Mashwani, "MOEA/D with DE and PSO: MOEA/D-DE+PSO," 2011.
- [33] W. K. Mashwani, "Comprehensive survey of the hybrid evolutionary algorithms," *International Journal of Applied Evolutionary Computation (IJAEC)*, vol. 4, no. 2, pp. 1–19, 2013.
- [34] W. K. Mashwani and A. Salhi, "A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation," *Applied Soft Computing*, vol. 12, no. 9, pp. 2765–2780, 2012.
- [35] W. K. Mashwani, "Multiobjective memetic algorithm based on decomposition," *Applied Soft Computing*, vol. 21, pp. 221–243, 2014.
- [36] X. Qian, X. Wang, Y. Su, and L. He, "An effective hybrid evolutionary algorithm for solving the numerical optimization problems," *Journal of Physics: Conference Series*, vol. 1004, 2018.
- [37] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.
- [38] F. H. M. T. Back and H. Schwefel, *A Survey of Evolution Strategies*, Morgan Kaufman, Mateo, CA, USA, 1997.
- [39] J. R. Koza, "Genetic programming: on the programming of computers by means of natural selection," 1992.
- [40] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [41] J. Koza, *Genetic Programming II Autonomous Discovery of Reusable Programs*, MIT Press, New York, NY, USA, 1995.
- [42] R. Storn and K. V. Price, "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," 1995.
- [43] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [44] J. Kim, T. Sharma, B. Kumar et al., "Research article inter-cluster Ant colony optimization algorithm for wireless sensor network in dense environment," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.
- [45] L. Perez Caceres, M. Lopez-Ibanez, and T. Stutzle, "Ant colony optimization on a limited budget of evaluations," *Swarm Intelligence*, vol. 9, 2015.
- [46] X.-S. Yang and S. Deb, *Cuckoo Search via Levy Flights*, IEEE, Berlin, Germany, 2009.
- [47] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [48] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics*, P. Dario, G. Sandini, and P. Aebischer, Eds., pp. 703–712, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
- [49] H. Ahmed and J. Glasgow, *Swarm Intelligence: Concepts, Models and Applications*, School of Computing, Queens University, London, UK, 2012.
- [50] A. ur Rehman, A. Islam, and S. B. Belhaouari, "Multi-Cluster jumping particle swarm optimization for fast convergence," *IEEE Access*, vol. 8, 2020.
- [51] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [52] Y. Feng, G.-G. Wang, and X.-Z. Gao, "A novel hybrid cuckoo search algorithm with global harmony search for 0-1 knapsack problems," *International Journal of Computational Intelligence Systems*, vol. 9, no. 6, 2013.
- [53] R. A. Khanum, M. A. Jan, W. K. Mashwani, N. M. Tairan, H. U. Khan, and H. Shah, "On the hybridization of global and local search methods," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 3, pp. 3451–3464, 2018.
- [54] J. Liang, B. Qu, P. Suganthan, and A. Hernández-Díaz, *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, 2013.
- [55] M. P. J. Awad, *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization*, Nanyang Technological University, Singapore, 2016.
- [56] W. Fang, X. Li, M. Zhang, and M. Hu, "Nature-inspired algorithms for real-world optimization problems," *Journal of Applied Mathematics*, vol. 2015, 2015.
- [57] W. K. Mashwani, "An introduction to the collective behaviour of swarm intelligence," 2017.