Knowledge-Based Systems 165 (2019) 374–406

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Improved sine cosine algorithm with crossover scheme for global optimization

Shubham Gupta *, Kusum Deep

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India

HIGHLIGHTS

- A new method called ISCA is proposed for global optimization problems.
- The ISCA improves the SCA using crossover and personal best memory of agents.
- The classical, CEC 2014 and CEC 2017 benchmarks are used to examine ISCA.
- The ISCA is used for engineering problems and image thresholding problem.
- Comparisons illustrate the improvement on the performance of ISCA.

ARTICLE INFO

Article history:
Received 2 September 2018
Received in revised form 4 December 2018
Accepted 6 December 2018
Available online 17 December 2018

Keywords:
Optimization
Population based algorithms
Sine cosine algorithm
Engineering optimization problems
Multilevel thresholding

ABSTRACT

Sine Cosine Algorithm is a recently developed algorithm based on the characteristics of sine and cosine trigonometric functions, to solve global optimization problems. This paper introduces a novel improved version of sine cosine algorithm, which enhances the exploitation ability of solutions and reduces the overflow of diversity present in the search equations of classical SCA. The proposed algorithm is named as ISCA. The key feature in the proposed algorithm is the hybridization of exploitation skills of crossover with personal best state of individual solutions and integration of self-learning and global search mechanisms. To evaluate these skills in ISCA, a classical set of well-known benchmark problems, standard IEEE CEC 2014 benchmark test and a recent set of benchmark problems, IEEE CEC 2017 have been taken. Several performance metrics (such as convergence, statistical test, performance index), employed on ISCA, ensure the robustness and efficiency of the algorithm. In the paper, the proposed algorithm ISCA is also used to solve five well-known engineering optimization problems. At the end of the paper, the proposed algorithm is also used for multilevel thresholding in image segmentation. The numerical experiments and analysis demonstrate that the proposed algorithm (ISCA) can be highly effective in solving real-life optimization problems.

1. Introduction

Optimization is the process of selecting or determining the best solution from the available set of possible solutions. Nowadays optimization is unavoidable in engineering, science, finance and various other fields. The general form of single objective optimization problem can be defined as:

$$\boldsymbol{min/max} \quad f(x), \qquad x = (x_1, x_2, \ldots, x_d) \in R^d \tag{1}$$

$$\boldsymbol{s.t.} \quad g_j(x) \le 0 \qquad j = 1, 2, \ldots, J. \tag{2}$$

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

$$l_i \le x_i \le u_i \qquad i = 1, 2, \ldots, d \tag{4}$$

where, $f, g_j(j = 1, 2, \ldots, J)$ and $h_k(k = 1, 2, \ldots, K)$ are real valued functions and are known as objective function, inequality constraints and equality constraints. These functions may be linear or non-linear. $x \in R^d$ is $d-$ dimensional decision vector, l_i and u_i are the lower and upper limits for the ith component of a decision vector. J and K represents the total number of inequality and equality constraints.

In the literature, a large number of traditional techniques and exact methods are available which have been applied to solve the optimization problems of various fields such as multiproduct economic production quantity problem [1], feature selection [2], machine learning [3], supply chain [4–10], selective maintenance scheduling [11], Single machine scheduling [12], vehicle routing problem [13], and many others [14,15]. But, these traditional methods cannot be applied everywhere. The real-life optimization problems where the continuity, differentiability, convexity of the objective function and/or constraints are absent, the traditional

* Corresponding author.
E-mail addresses: sgupta@ma.iitr.ac.in (S. Gupta), kusumfma@iitr.ac.in (K. Deep).

https://doi.org/10.1016/j.knosys.2018.12.008

techniques fail to start the process of determination of optima. Nature Inspired Optimization techniques works as a boon in all such cases because in these techniques any information regarding the problem is not needed. From some past decades, the Nature Inspired Optimization has received enormous attention. Generally, these techniques mimic the intelligent behavior of various phenomenon which occurs in nature. These techniques treated the problem as a black box and try to explore all the search regions of the search space which are promising. Based on the number of search agents, nature inspired optimization techniques/algorithms can be grouped into two categories — single solution based algorithms and population-based algorithms. In single solution based algorithms, the search process is followed by a single search agent. Hill Climbing [16], Simulated Annealing [17] are some algorithms which are single solution based. In population based algorithms, multiple search agents proceed the search process. The advantage of population-based algorithms is the sharing of useful information between the search agents and this increases the exploration ability of search agents as compared to single solution based algorithms. Also, the multiple agents help each other to avoid the local solutions. Genetic Algorithm (GA) [18], Particle Swarm Optimization (PSO) [19], Differential Evolution (DE) [20], Ant Colony Optimization (ACO) [21], Artificial Bee Colony (ABC) algorithm [22], Grey Wolf Optimizer (GWO) [23], Ant Lion Optimizer (ALO) [24], Moth-flame Optimization (MFO) [25], Whale Optimization Algorithm (WOA) [26] Invasive weed optimization algorithm [27], Flower Pollination Algorithm [28,29] are some examples of population based algorithms.

Sine cosine algorithm (SCA) [30] is a recently developed population-based approach to solve global optimization problems. SCA uses the characteristics of sine and cosine trigonometric functions in the search process. In [30], SCA has shown its efficiency in terms of exploration and exploitation. Due to its exploration ability it has been applied to solve many real-life applications. Hafez et al. [31] have used classical SCA for the feature selection problem. SCA was used to train a feed-forward neural network [32]. In [33], SCA was also used for Handwritten Arabic Manuscript Image Binarization. In [34], the application of SCA is demonstrated for training artificial neural network in the problem of load forecasting. In [35], SCA is used for data clustering problem. In [36], the parameters of support vector regression have been optimized by SCA. In [37], the performance of SCA has been used for the optimization of a vehicle engine connecting rod. In [38], the loading margin stability is optimized using SCA to improve the power system security. In [39], the solution of optimal power flow problem is obtained using SCA. In [40], unified power quality conditioner allocation in the distribution system is addressed by SCA. In [41], SCA is used to optimize the complementary metal-oxide semiconductor analog circuits.

Although the literature shows that the SCA has enough ability to explore the search space but like other algorithms, it faces some difficulties like local optima stagnation, slow convergence and skipping of true solutions while solving the real-life problems. To alienate these issues from classical SCA, some attempts have been done in the literature. For example — In [42], an improved version of SCA is proposed based on orthogonal parallel information for global optimization tasks. In [43], the levy flight strategy is integrated into SCA to avoid the stagnation at local solutions. In [44], SCA has been hybridized with opposition based learning to train the feed-forward neural network. In [45], an improved version of SCA, by incorporating the opposition based learning, is introduced to solve the global optimization problems. In [46], an improved version of SCA is used for unit commitment problem. In [47], a novel weighted update mechanism is used to improve the performance of SCA. In [48], the search equations of classical SCA are modified to enhance the search ability in promising directions and to avoid the problem of stagnation at local solutions. In [49], the modified SCA has been combined with an extreme learning machine for pathological brain detection. In [50] SCA and Water Wave Optimization algorithm is hybridized to propose an efficient optimizer. In [51], the SCA has been hybridized with DE and employed on structural damage detection problems. In [52], the SCA and DE are hybridized and applied for visual tracking. In [53], SCA is hybridized with multi-orthogonal search strategy for manufacturing optimization problems. In [54], hybrid Q-learning sine–cosine-based strategy is proposed for addressing the combinatorial test problems. The multiobjective version of SCA is used to develop a novel forecasting system for wind speed forecasting [55] and for engineering design problems [56].

Although, there are many attempts have been done to improve the performance of SCA, but from the analysis of their results on benchmark test problems, it can be observed that in some cases SCA still suffers from the problem of overflow of diversity and stagnation at local optima. Therefore in the present work, the search mechanism of SCA has been improved by modifying its search equations, so that the above mentioned issues of SCA can be avoided.

The aim of the present paper is to introduce the improved version of classical SCA which is more efficient to search for optima during the intermediate iterations. Therefore, in the present work, an enhanced version of SCA is proposed which is named as Improved Sine Cosine Algorithm (ISCA) in the paper. ISCA, uses the personal best memory in the search equation in place global state of the population, to decide the search area. In the search equation of ISCA global search direction is also integrated to utilize the useful information of a current best solution of the problem. To maintain the best features of a solution, and to explore the search regions around the best memory of search agents, a crossover is performed between the current solution and its personal best memory. To prevent the overflow of diversity, a greedy selection has been applied between the previous and current population of solutions. The proposed modifications in the search strategy of SCA reduces the difficulties of classical SCA. The proposed algorithm considers the personal best as well as global information to explore the search space as in PSO [19], but in the proposed algorithm the crossover operator allows the search around the individual's best memory and the sine cosine functions maintain the diversity and exploitation near the current solution using algorithm parameters. The proposed algorithm can also be called as memetic algorithm as the crossover operator has been merged with the search strategy of SCA. Crossover is an evolutionary algorithms-based strategy to evolve the search agents. The term 'memetic algorithm' was first presented by Moscato in [57] with local improvement strategy to search the solution. The memetic algorithm provides a local search in establishing the exploitation of the search space in an algorithm. Memetic algorithms are hybrid methods which are based on population-based search strategy [58,59] and neighborhood-based local search framework [60].

To prove the efficacy and reliability of the proposed algorithm on real-life optimization problems, first, it has been tested on classical benchmark set as well as on standard IEEE CEC 2014 [61] and latest CEC 2017 [62] benchmark set of problems. Secondly, the proposed algorithm has been used to solve benchmark engineering optimization problems. In the last, the proposed algorithm is also applied for multilevel thresholding problem. Comparison with classical SCA indicates that the proposed algorithm is much better than classical SCA in terms of accuracy and efficacy.

In the literature, various evolutionary algorithms have been used for optimal thresholding. For example, Ayala et al. [63] have proposed a novel beta DE approach for thresholding. Maitra and Chatterjee [64] in 2008, have used a cooperative comprehensive learning based PSO for image segmentation. Ali et al. [65] have

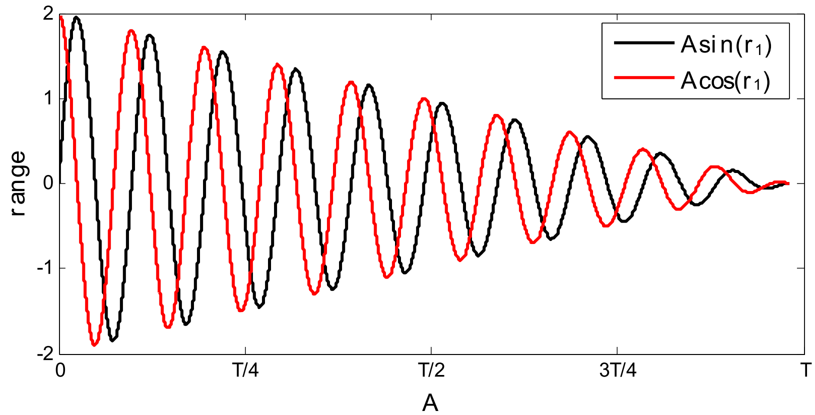

Fig. 1. The impact of sine and cosine functions with random vector A.

determined the optimal thresholds using synergetic DE. Sarkar et al. [66] have applied thresholding for colored images based on DE and minimum cross entropy. Hammouche et al. [67] have studied the effects of various meta-heuristic techniques on image thresholding and concluded the better performance ability of DE and rapid convergence of PSO.

In some previous recent years, various new and efficient approaches have been applied to determine the optimal thresholds for image segmentation. Aziz et al. [68] have applied the Whale Optimization Algorithm (WOA) [26] and Moth-flame Optimization [25] for multiple thresholding. In [69], Cuckoo Search (CS) and Wind Driven Optimization have been studied on image thresholding. In [70], the determination of multiple thresholds has been done using Grey Wolf Optimizer. In [71], Elephant Herding Optimization Algorithm is used to determine the optimal thresholds for image segmentation. In [72], an improved version of Bacterial Foraging Optimization is used for multilevel thresholding. In [73], Modified Firefly Algorithm is used for color image multilevel thresholding.

The remaining of the paper is organized as follows — Section 2 provides a brief description of Sine Cosine Algorithm. In Section 3, the improved version of Sine Cosine Algorithm named as ISCA is discussed in detail. In Section 4, evaluation and analysis of the proposed algorithm have been done based on 23 classical, 30 standard IEEE CEC 2014 and 30 latest IEEE CEC 2017 benchmark problems. In Section 5, the applications of ISCA is discussed on five engineering optimization problems. In Section 6, the proposed algorithm is applied to solve multilevel thresholding and finally, Section 7 concludes the work of the paper and suggests some future ideas.

2. Sine Cosine Algorithm (SCA)

The Sine Cosine Algorithm (SCA) is a recently developed meta-heuristic algorithm based on the mathematical characteristics of sine and cosine trigonometric functions. This algorithm was designed by Mirjalili in 2015 [30]. Like other population-based meta-heuristic optimization algorithms, SCA also starts with a set of randomly distributed solutions, then each solution updates their position with the help of following equations —

$$x_{i,t+1} = x_{i,t} + A\sin(r_1)\,|Cx_{Best} - x_{i,t}| \tag{5}$$

$$x_{i,t+1} = x_{i,t} + A\cos(r_1)\,|Cx_{Best} - x_{i,t}| \tag{6}$$

The above two equations are used in SCA in a following manner

$$x_{i,t+1} = \begin{cases} x_{i,t} + A\sin(r_1)\left|Cx_{Best} - x_{i,t}\right| & if\ r < 0.5 \\ x_{i,t} + A\cos(r_1)\left|Cx_{Best} - x_{i,t}\right| & otherwise \end{cases} \tag{7}$$

where $x_{i,t}$ and $x_{i,t+1}$ represents the ith solution vector at tth and $(t+1)^{th}$ iteration respectively. x_{Best} is the fittest solution in the solution set. r is a uniformly distributed random number in the interval (0, 1). r_1 is a vector that decides the direction of moment of current solution which can be either towards the x_{Best} or outside x_{Best}. The vector C provides a weight to x_{Best} which emphasizes the exploration ($C > 1$) and exploitation ($C < 1$). The vector C also helps in avoiding the premature convergence at the end of iterations. The vector r helps in transition from sine to cosine functions and vice versa. The effect of random number A on sine and cosine function is shown in Fig. 1.

The parameter A is a random vector which decides the area of the search space around the current solution. This region of search space may lie inside x_{Best} and $x_{i,t}$ or outside them. The parameter A also helps in exploration and exploitation of a search space as well as in maintaining a suitable balance between them. In the first half of the total number of iterations, coefficient A contributes in the exploration of a search space while in the second half of the total number of iterations, it is devoted to the exploitation of an available search space. Mathematically the vector A can be defined as follows —

$$A = 2 - 2\left(\frac{t}{T}\right) \tag{8}$$

where T represents the maximum number of iterations which is predefined as a termination criteria for SCA. The steps of Sine Cosine Algorithm are pointed in Algorithm 1.

Although the sine cosine algorithm is very efficient to explore the search space but in many cases, it suffers from some major difficulties like skipping of true solutions and local optima stagnation and therefore an improvement is required in the search strategy of SCA. The present work attempts to improve the efficiency of classical SCA.

3. The proposed ISCA algorithm:

In this section, the proposed algorithm ISCA has been described in detail. The stepwise description and analysis of proposed ISCA is as follows:

3.1. Motivation for improvement in SCA

Although, the sine cosine algorithm explores the search space very efficiently but like other population-based algorithms, it sometimes faces the overflow of diversity (exploration). The overflow of diversity skips the true solutions of the problem if the suitable balance between exploration and exploitation is not present in the algorithm. In the search equations of SCA, the solution is

Algorithm 1. Classical Sine Cosine Algorithm (SCA)

1. Generate the uniformly distributed, initial set of random solutions within the search space
2. Compute the fitness of each solution vector
3. Initialize the parameters T (maximum number of iterations) and coefficient A.
4. Select the fittest solution x_{Best} from the set of solutions
5. Initialize the iteration count $t = 0$
6. **while** $t < T$
7. Update each solution vector with the help of equation (7) and (8)
8. Compute the fitness of each updated solution vector
9. Update the best solution x_{Best}
10. Update the coefficient A
11. $t = t + 1$
12. **end of while**
13. Return the fittest solution x_{Best}

updated around the current state of a solution and the area of a search space is decided by the coefficient A. In the prior generation of an algorithm, the solutions are reallocated far from the current state as the coefficient A supports the exploration of the search space. The coefficient C also contributes to the exploration during the search process. Therefore, during the search process, in each generation, the solution losses its own features and always reallocate to its new location. The overflow of diversity (exploration) and loss of features of solutions skips the true solution and these skipped solutions may have a chance to provide better positions in the next generation by exploiting the regions around them.

Therefore, to alleviate the above mentioned issues from classical SCA, some modifications have been done in the proposed algorithm ISCA to proceed the search. The modifications which have been proposed in ISCA are pointed as follows:

i. In the search equation of classical SCA, the fittest position x_{Best} is completely replaced by the current best (x_{pBest}) position in order to prevent from the stagnation in local solutions. The search process in the direction of personal best helps in exploring the promising regions around the personal best solutions. This strategy also helps when the elite solution stuck in local optima and fails to guide the search.
ii. An available optimum direction (i.e. along the current best solution within the population of solutions) is integrated into the search equation of classical SCA.
iii. To integrate the personal best features of solutions, a crossover is performed between the updated position and its personal best position (x_{pBest}) obtained so far. This helps in preventing from the skipping of true solutions during the search.
iv. To reduce the overflow of diversity and to provide a greedy direction of search, greedy selection has been applied between the updated state and previous state of solutions.

These modifications in the search process of classical SCA are adopted to enhance the exploitation of search space with the personal best memory of solutions and to provide a direction along the fittest solution of the population. A greedy selection and crossover mechanism prevent the solution set from the overflow of diversity and keeps the best feature of the solution set in the population of solutions respectively. The crossover and the greedy selection mechanisms, used in ISCA are described as follows:

Crossover

Through the crossover operator, the two parent solution are hybridized to produce the offspring consisting the features of both the parent solutions. The crossover which is used in the present work is defined as follows:

$$u_i = \begin{cases} v_i^j & if\ r_j \leq CR \\ x_i^j & otherwise \end{cases} \tag{9}$$

where CR is the crossover rate. r_j is the uniformly distributed random number between 0 and 1 corresponding to the dimension j. v_i and x_i are two different parent solutions to be crossed. In the present work, the crossover probability CR is taken as 0.3.

Greedy selection

Greedy selection is the process which decides that the updated vector u_i will survive in the next generation or not and this is done by comparing the fitness of updated and current solution vector. The selection operator is described as follows:

$$x_{i,t+1} = \begin{cases} u_{i,t} & if \quad f\left(u_{i,t}\right) \leq f(x_{i,t}) \\ x_{i,t} & if \quad f\left(u_{i,t}\right) > f(x_{i,t}) \end{cases} \tag{10}$$

where f is the objective function. The above selection process is described for the minimization problem. The introduced inequality "$\leq$" in Eq. (10) helps to navigate the population of solutions from the flat search areas of fitness landscape and reduces the possibility of solutions becoming stagnated.

3.2. Framework of Improved Sine Cosine Algorithm (ISCA)

Like the classical SCA, the proposed ISCA also starts with the same set of uniformly distributed solutions (called population) within the search space to analyze the effectiveness of the search process in ISCA. After initializing the population of solutions, the search process for optima of the problem starts. The modified search equation introduced in ISCA is as follows:

$$v_{i,t+1} = \begin{cases} x_{i,t} + \overbrace{A\sin(r_1)\left|Cx_{i,pBest} - x_{i,t}\right|}^{\text{Cognitive Component}} + \overbrace{r_2(x_{Best} - x_{i,t})}^{\text{Social Component}} & if\ r < 0.5 \\ x_{i,t} + \underbrace{A\cos(r_1)\left|Cx_{i,pBest} - x_{i,t}\right|}_{\text{Cognitive Component}} + \underbrace{r_2(x_{Best} - x_{i,t})}_{\text{Social Component}} & otherwise \end{cases} \tag{11}$$

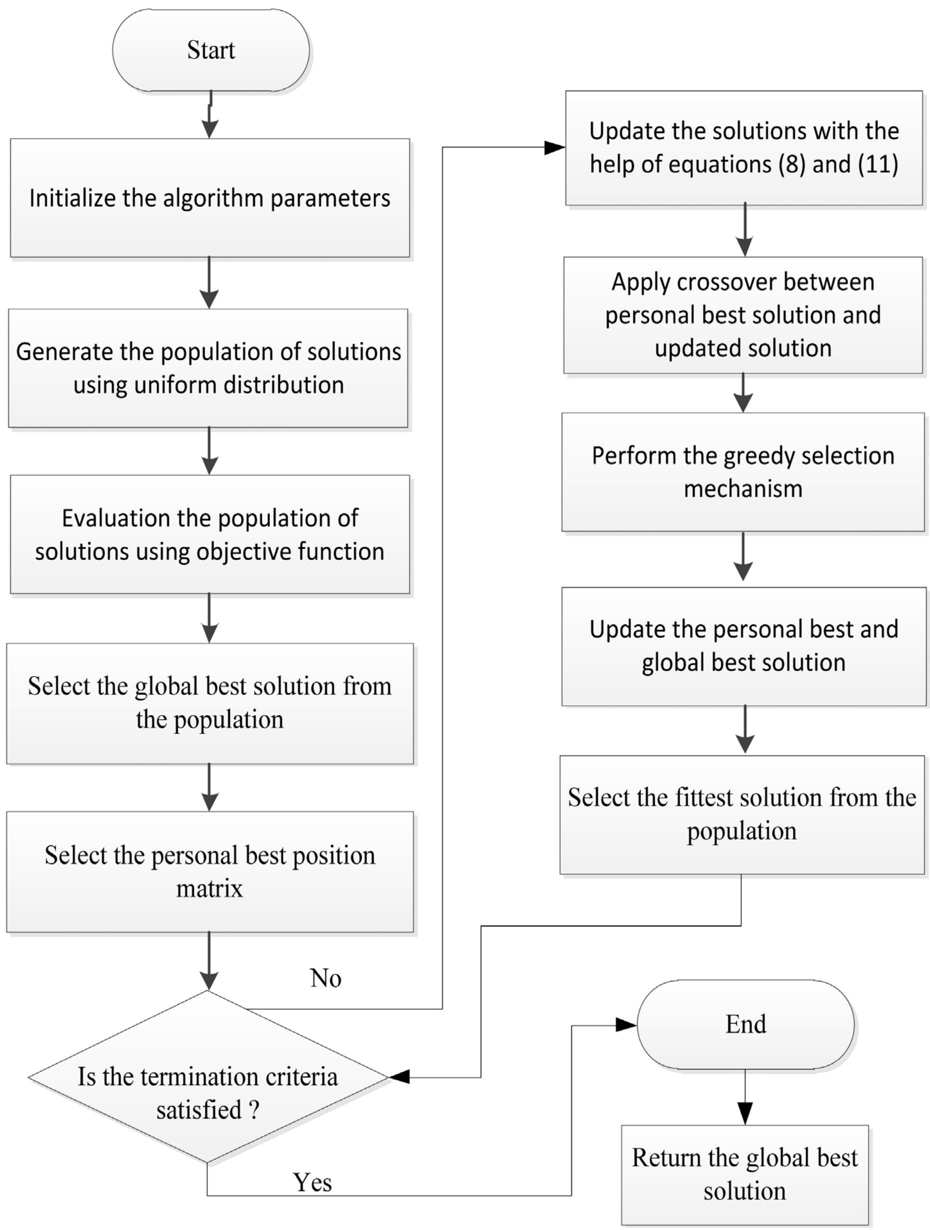

Fig. 2. Flow chart of the proposed improved sine cosine algorithm (ISCA).

where, $x_{i,t}$ is the solution at iteration t, $x_{i,pBest}$ is the current best position obtained so far of the solution x_i, x_{Best} is the best position within the population of solutions, r_2 is a uniformly distributed random number between 0 and 1. All the other parameters A, r_1, C and r are same as in classical SCA.

In Eq. (11) the second term on the right-hand side contributes the cognitive component in the search process and the third term contributes to the social component. The benefit of addressing these two components in the search process is to perform the local and global search during the search process. The cognitive and social components provides an efficient and promising direction to the current solution by combining the directions along the solution's best and population's best state.

The solution updated with the help of Eq. (11) may have a chance to diverge from the current state of a solution when the search area provided by the coefficient A is very large. Therefore in order to deal such situation and to integrate the own best features of a solution, the updated solution $v_{i,t+1}$ is crossed with the current best solution $x_{i,pBest}$ of solution x_i. This crossover is performed with Eq. (9). After the crossover mechanism, the greedy selection is performed between the current best solution $x_{i,pBest}$ and obtained solution u_i, which keeps the balance between exploration and exploitation in the search process. All the above steps are clearly summarized in Algorithm 2 and the flowchart is also presented in Fig. 2.

4. Validation of proposed ISCA

The proposed improved Sine Cosine Algorithm (ISCA) can be considered more efficient optimizer as compared to classical SCA in terms of exploiting the promising regions of a search space and in keeping the personal best features within the population of solutions. To validate the strength and advantages of integrated strategies in proposed ISCA, three set of benchmark problems — classical set of 23 well-known benchmark test problems, standard IEEE CEC 2014 benchmark set and a latest set of benchmarks, IEEE CEC 2017 of 30 problems with dimension 10 and 30 have been taken.

Algorithm 2. Improved Sine Cosine Algorithm (ISCA)

1. Generate the uniformly distributed, initial set of random solutions within the search space
2. Compute the fitness of each solution vector
3. Initialize the parameters T, A and CR
4. Select the fittest solution x_{Best} from the population of solutions
5. Store the current positions as personal best position matrix $[x_{i,pBest}]_{i=1}^{N}$
6. Initialize the iteration count $t = 0$
7. while $t < T$
8. for each individual solution
9. Update the position with the help of equation (8) and (11)
10. Apply the crossover operator between personal best and updated solution as described in equation (9)
11. Compute the fitness of updated solution vector
12. Perform the greedy selection mechanism defined in equation (10)
13. Update the personal best solution $x_{p,Best}$
14. Update the best solution x_{Best}
15. end of for
16. Update the parameter A using equation (8)
17. $t = t + 1$
18. end of while
19. Return the fittest solution x_{Best}

4.1. Benchmark problem set I:

In the first set of benchmarks, the classical set is used to examine the performance of the proposed algorithm (ISCA). In the classical set, the benchmark test problems are categorized in three groups — (*i*) Unimodal, (*ii*) Multimodal, and (*iii*) fixed dimension multimodal problems. Out of 23 test problems, first 13 problems are scalable while remaining are non-scalable. These test problems are described in detail in Table 1. 2-D shapes of some selected benchmark test problems are shown in Fig. 3 which are plotted with the help of code provided in [23]. This problem set has been used to evaluate the performance of various algorithms in the literature [23–25,74,75].

4.1.1. Experimental setup

To implement the proposed algorithm ISCA on test problems and to study the comparative performance of ISCA with classical SCA, the fair selection of parameter is done to avoid the biasedness. For the classical set of problems, the population size of solutions is taken as 30, as in [17] for both the algorithms and the termination criteria is fixed as 1000 iterations for all the problems. As the proposed algorithm does not adds extra function evaluation, therefore the selection of 1000 iterations is fair, i.e. both the algorithms use the same number of function evaluations (30,000). The obtained results on these problems are presented in Tables 2 to 4.

4.1.2. Analysis on benchmark problem set I

4.1.2.1. Unimodal test problems. In the unimodal problems, only one global optimum is present and no other local minima are present in these test problems. Generally, these unimodal test problems evaluate the capability of exploitation and convergence rate. In the test set, problems F1 to F7 are unimodal and the results of proposed ISCA on these problems is presented in Table 2. From the table, it can be analyzed easily that in all the unimodal test problems, the proposed algorithm ISCA outperforms classical SCA in all the statistical measures like best, mean, median, mean, worst and standard deviation of objective function values. Therefore, the performance of the proposed algorithm on unimodal test problems demonstrates its efficacy in terms of exploiting the search regions and convergence rate.

4.1.2.2. Multimodal test problems. In the test set, functions F8 to F13 are multimodal with a large number of local optima. These multimodal test problems generally used to evaluate the exploration ability of search algorithms because a large number of local optima increases the probability of stagnation. The results on these test problems are reported in Table 3. In all the tests problems the proposed algorithm (ISCA) beats the classical SCA. In the problems F9 and F11 the proposed algorithm is able to locate the optima of the problem while in other problems the results are very impressive and provide a better approximation to the optima as compared to the classical SCA.

4.1.2.3. Fixed-dimension multimodal problems. The problems F14 to F23 are multimodal problems but with low dimensions. These test problems have less number of local optima as the dimension is very less as compared to the considered multimodal problems (F8–F13). These test problems examine the balance between exploration and exploitation ability of search algorithms. From Table 4 it can be observed that the proposed algorithm, ISCA outperforms classical SCA in all the test problems (F14 to F23) except the problems F15. In problem F15 the results are very competitive as compared to classical SCA. In all the multimodal test problems, ISCA able to allocate the optima of the problems with a very less error.

Thus in terms of exploiting the promising regions of a search space and in exploring the search regions that are not discovered earlier but are very promising, the proposed ISCA is very effective as compared to classical SCA and it provides a far better solution as compared to classical SCA. By analyzing the Tables 2–4, it can be concluded that the proposed algorithm ISCA have proved the efficacy and efficiency in determining the global optima.

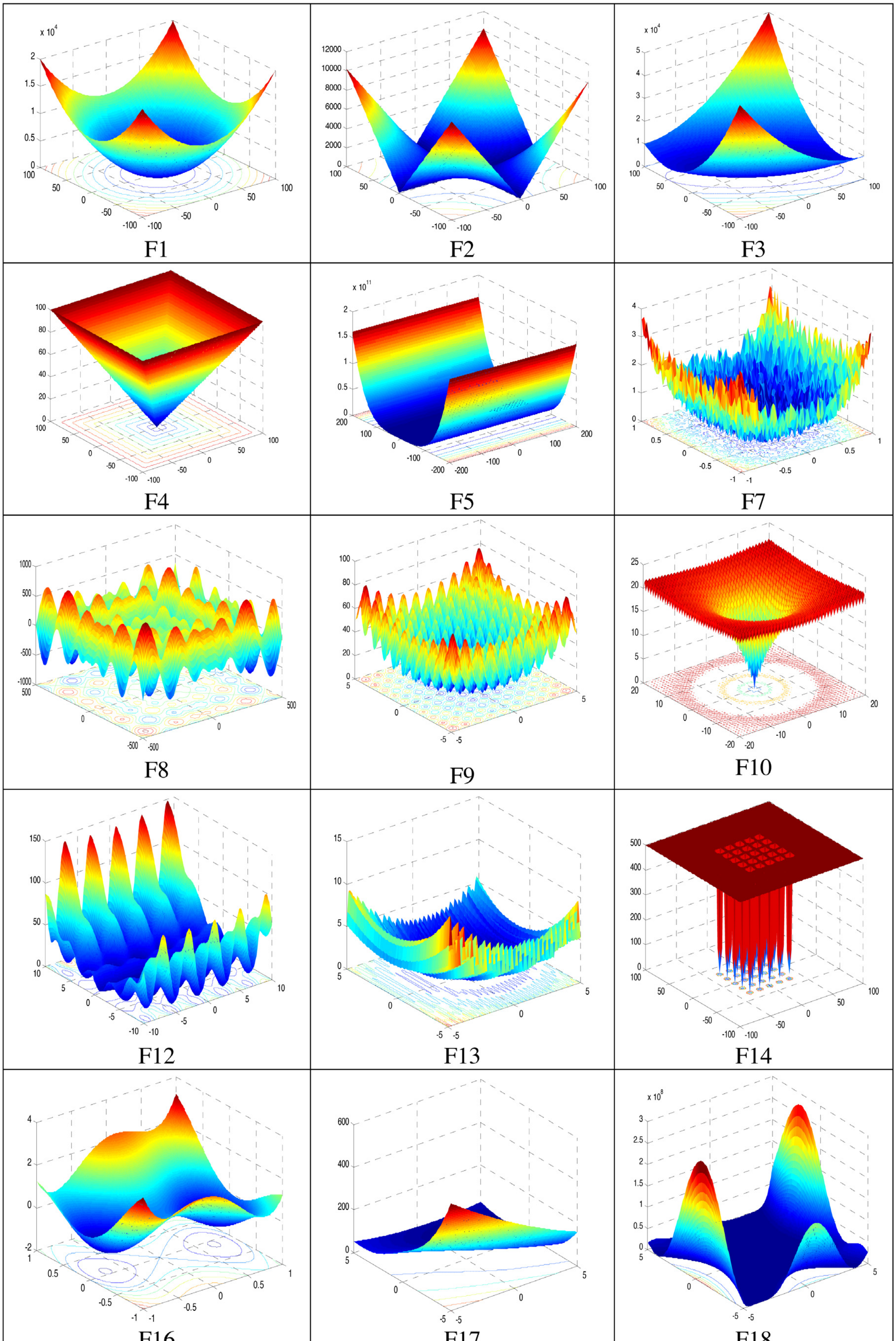

Fig. 3. 2-D shape of some selected benchmarks.

Table 1
Classical benchmark test problems.

Test problems	Dimension	Range of search space	F_{min}
Unimodal benchmark problems			
$F1(x) = \sum_{i=1}^{d} x_i^2$	30	[−100, 100]	0
$F2\,(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	30	[−10, 10]	0
$F3\,(x) = \sum_{i=1}^{d} (\sum_{j-1}^{i} x_i)^2$	30	[−100, 100]	0
$F4\,(x) = max_i\{\lvert x_i \rvert, 1 \le i \le d\}$	30	[−100, 100]	0
$F5\,(x) = \sum_{i=1}^{d-1} [100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
$F6\,(x) = \sum_{i=1}^{d} ([x_i + 0.5])^2$	30	[−100, 100]	0
$F7\,(x) = \sum_{i=1}^{d} i \cdot x_i^4 + rand[0, 1)$	30	[−1.28, 1.28]	0
Multimodal benchmark problems			
$F8\,(x) = \sum_{i=1}^{d} -x_i sin(\sqrt{\lvert x_i \rvert})$	30	[−500, 500]	−418.9829 × d
$F9\,(x) = \sum_{i=1}^{d} [x_i^2 - 10cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$F10\,(x) = \sum_{i=1}^{d} -20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)) + 20 + e$	30	[−32, 32]	0
$F11\,(x) = \frac{1}{4} \times 10^{-3} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(x_i/\sqrt{i}\right) + 1$	30	[−600, 600]	0
$F12\,(x) = \frac{\pi}{d}\{10sin\,(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$ $y_i = \frac{x_i + 5}{4}$ $u\,(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m & if\ x_i > a \\ k\,(-x_i - a)^m & if\ x_i < -a \\ 0 & otherwise \end{cases}$	30	[−50, 50]	0
$F13\,(x) = 0.1 \times \left\{\sin^2(3\pi x_1) + \sum_{i=1}^{d} (x_i - 1)^2 \left[1 + \sin^2(1 + 3\pi x_i)\right] + (x_d - 1)^2 \left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{d} u(x_i, 5, 100, 4)$	30	[−50, 50]	0
Fixed-dimension multimodal benchmark problems			
$F14\,(x) = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - x_{ij})^6}\right]^{-1}$	2	[−65, 65]	1
$F15\,(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5, 5]	0.0003
$F16\,(x) = 4x_1^2 - 2.1x_1^4 + 0.33x_1^6 10 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$F17\,(x) = 10 + 10 \times \left(1 - \frac{0.125}{\pi}\right) cos\,(x_1) + \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2$	2	[−5, 5]	0.398
$F18\,(x) = \left[1 + (1 + x_1 + x_2)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2, 2]	3
$F19\,(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
$F20\,(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
$F21\,(x) = -\sum_{i=1}^{5} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.1532
$F22\,(x) = -\sum_{i=1}^{7} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.4028
$F23\,(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363

Table 2
Best, Mean, Median, Worst and Standard Deviation (STD) of objective function values obtained in 30 runs on unimodal test problems by classical SCA and proposed ISCA.

Test function	Algorithm	Best	Mean	Median	Worst	STD
F1	SCA	4.74E−06	3.83E−02	2.43E−03	4.99E−01	1.05E−01
	ISCA	**1.04E−61**	**2.01E−58**	**1.37E−59**	**5.10E−57**	**9.26E−58**
F2	SCA	3.13E−09	1.89E−05	4.02E−06	2.95E−04	5.46E−05
	ISCA	**3.82E−38**	**1.87E−36**	**7.21E−37**	**1.22E−35**	**2.52E−36**
F3	SCA	6.20E+00	3.66E+03	2.72E+03	1.02E+04	3.12E+03
	ISCA	**2.97E−09**	**2.52E−04**	**2.75E−06**	**6.09E−03**	**1.11E−03**
F4	SCA	5.91E+00	2.19E+01	2.10E+01	4.20E+01	9.76E+00
	ISCA	**3.78E−17**	**2.07E−15**	**8.80E−16**	**9.93E−15**	**2.64E−15**
F5	SCA	28.30	318.40	39.62	3078.97	750.82
	ISCA	**25.51**	**26.05**	**26.07**	**27.08**	**0.41**
F6	SCA	3.64E+00	4.63E+00	4.59E+00	6.72E+00	6.08E−01
	ISCA	**1.48E−04**	**1.56E−01**	**3.06E−04**	**7.56E−01**	**2.09E−01**
F7	SCA	2.01E−03	3.47E−02	2.67E−02	1.48E−01	3.13E−02
	ISCA	**3.26E−04**	**1.13E−03**	**1.00E−03**	**3.85E−03**	**7.11E−04**

Table 3
Best, Mean, Median, Worst and Standard Deviation (STD) of objective function values obtained in 30 runs on multimodal test problems by classical SCA and proposed ISCA.

Test function	Algorithm	Best	Mean	Median	Worst	STD
F8	SCA	−4432.70	−3797.39	−3801.49	−3215.76	**294.68**
	ISCA	**−8513.65**	**−7665.59**	**−7680.60**	**−6233.21**	545.64
F9	SCA	8.56E−05	1.04E+01	1.05E+00	4.17E+01	1.36E+01
	ISCA	**0**	**0**	**0**	**0**	**0**
F10	SCA	2.18E−04	1.72E+01	2.02E+01	2.03E+01	6.76E+00
	ISCA	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**4.01E−31**
F11	SCA	9.69E−06	2.77E−01	2.21E−01	8.53E−01	2.63E−01
	ISCA	**0**	**0**	**0**	**0**	**0**
F12	SCA	4.39E−01	1.53E+00	9.94E−01	6.51E+00	1.49E+00
	ISCA	**1.09E−05**	**1.00E−02**	**6.82E−03**	**3.35E−02**	**8.89E−03**
F13	SCA	2.02E+00	6.72E+02	2.74E+00	1.98E+04	3.61E+03
	ISCA	**1.81E−04**	**1.69E−01**	**1.78E−01**	**4.07E−01**	**9.08E−02**

Table 4
Best, Mean, Median, Worst and Standard Deviation (STD) of objective function values obtained in 30 runs on fixed dimensional multimodal test problems by classical SCA and proposed ISCA.

Test function	Algorithm	Best	Mean	Median	Worst	STD
F14	SCA	0.998	1.660	0.998	2.982	0.951
	ISCA	**0.998**	**0.998**	**0.998**	**0.998**	**0**
F15	SCA	3.27E−04	**9.42E−04**	8.34E−04	**1.49E−03**	**3.36E−04**
	ISCA	**3.08E−04**	1.12E−03	**3.68E−04**	2.04E−02	3.64E−03
F16	SCA	−1.03160	−1.03159	−1.03160	−1.03150	0.00003
	ISCA	**−1.03160**	**−1.03160**	**−1.03160**	**−1.03160**	**0**
F17	SCA	0.39792	0.39871	0.39866	0.40104	0.00065
	ISCA	**0.39789**	**0.39789**	**0.39789**	**0.39789**	**0**
F18	SCA	3.00000	3.00001	3.00000	3.00020	0.00004
	ISCA	**3.00000**	**3.00000**	**3.00000**	**3.00000**	**0**
F19	SCA	−3.86180	−3.85501	−3.85435	−3.85240	0.00239
	ISCA	**−3.86280**	**−3.86280**	**−3.86280**	**−3.86280**	**0**
F20	SCA	−3.18830	−2.80511	−3.01170	−1.15570	0.49150
	ISCA	**−3.32200**	**−3.29028**	**−3.32200**	**−3.20310**	**0.05347**
F21	SCA	−7.71530	−2.73001	−0.88160	−0.35065	2.24021
	ISCA	**−10.15290**	**−10.15092**	**−10.15195**	**−10.12990**	**0.00415**
F22	SCA	−8.21600	−3.57250	−4.57765	−0.52113	2.23841
	ISCA	**−10.40280**	**−10.40182**	**−10.40210**	**−10.39860**	**0.00090**
F23	SCA	−7.58470	−4.28349	−4.76285	−0.94492	1.65663
	ISCA	**−10.53630**	**−10.53563**	**−10.53585**	**−10.53400**	**0.00059**

4.1.3. Convergence analysis

In this section, to ensure the better convergence rate in proposed ISCA, convergence curves have been plotted corresponding to the median value of the objective function obtained in 30 runs for different problems. In the figures, the horizontal axis represents the iterations and the vertical axis denotes the fitness of the objective function. In Figs. 4 to 7, the convergence curves are plotted for the unimodal as well as for multimodal test functions. In most of the test problems, the convergence is far better in proposed ISCA than classical SCA. Also, it is evident from the convergence curve that in most of the test problems ISCA has achieved more efficient and accurate solution as compared to classical SCA.

4.1.4. Comparison of CPU execution time between classical SCA and ISCA

In this section, the CPU execution time has been compared between classical SCA and proposed algorithm (ISCA). The CPU execution time of classical SCA and ISCA in obtaining the solutions of classical test problems is reported in Table 5. In the table mean value of execution time obtained in 30 runs is presented for all the problems. From the table, it can be observed that the proposed algorithm is slightly more time consuming than the classical SCA.

Table 5
Comparison of CPU execution time between classical SCA and proposed algorithm (ISCA).

Test problem	SCA	ISCA	Test problem	SCA	ISCA
F1	0.304	0.307	F13	1.011	1.066
F2	0.301	0.367	F14	1.966	2.110
F3	1.676	1.666	F15	0.261	0.291
F4	0.330	0.375	F16	0.161	0.170
F5	0.384	0.428	F17	0.145	0.158
F6	0.356	0.379	F18	0.139	0.151
F7	0.545	0.579	F19	0.359	0.389
F8	0.439	0.431	F20	0.381	0.408
F9	0.412	0.419	F21	0.571	0.479
F10	0.482	0.474	F22	0.542	0.576
F11	0.449	0.497	F23	0.698	0.776
F12	1.029	1.030			

4.1.5. Statistical analysis of results

According to Derrac et al. [76] the statistical analysis of the results is essential to compare the given algorithms. Therefore in this section, a non-parametric test has been applied between classical SCA and proposed ISCA to examine the improvement in ISCA. A non-parametric statistical test is chosen because it does not require the information regarding the distribution of the data set and in these tests the measure of central tendency is median. To evaluate the performance of any search algorithm, the median value is comparatively better statistical measure. The test has been performed at a 5% level of significance and the obtained results are

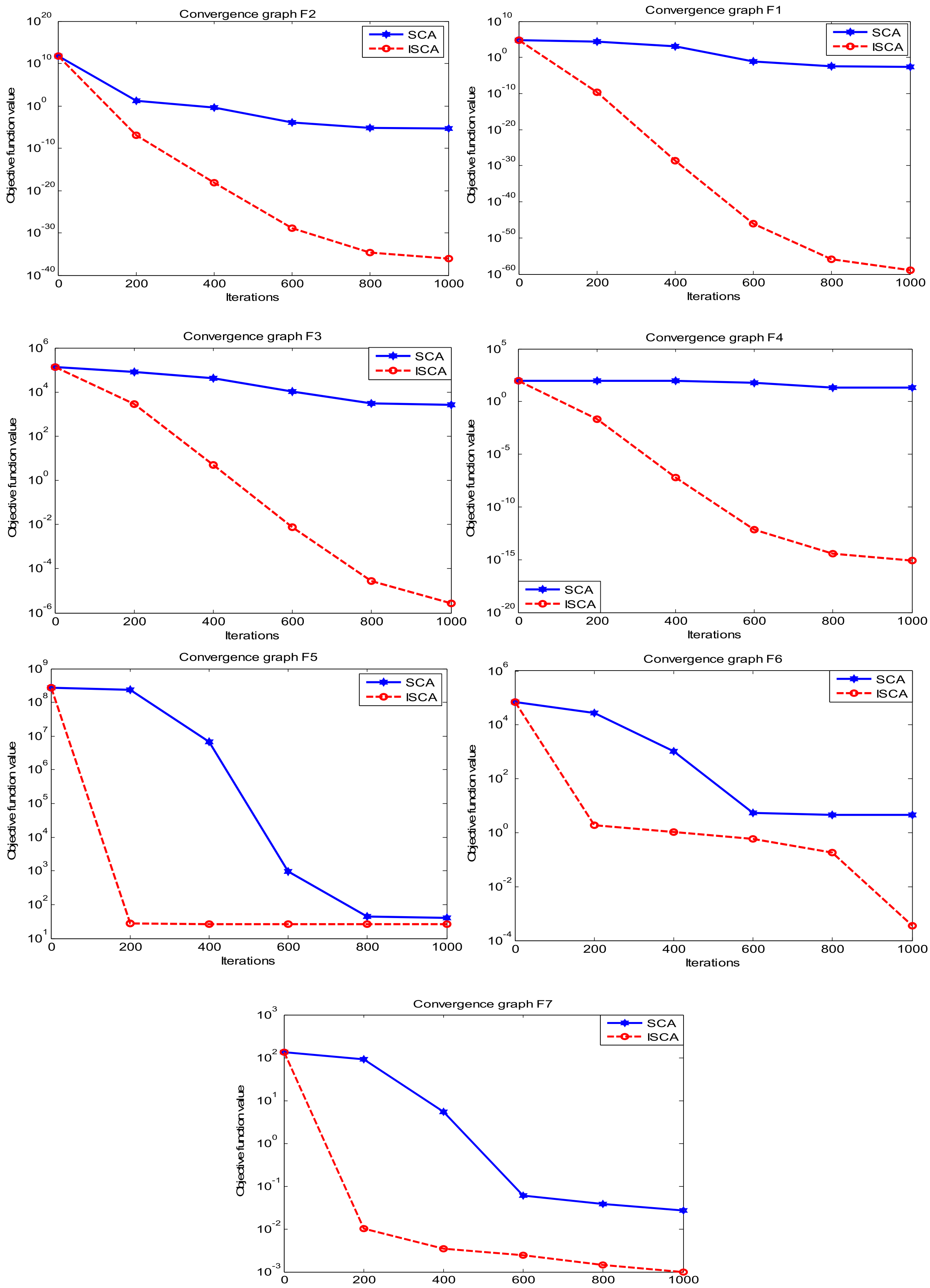

Fig. 4. Convergence corves for unimodal test problems.

reported in Table 6. In the table, p-values, obtained by applying the Wilcoxon test are also reported. In the table, symbol '+' is used to represent that proposed ISCA is significantly better than classical SCA, while '−' symbol is used to represent the better performance

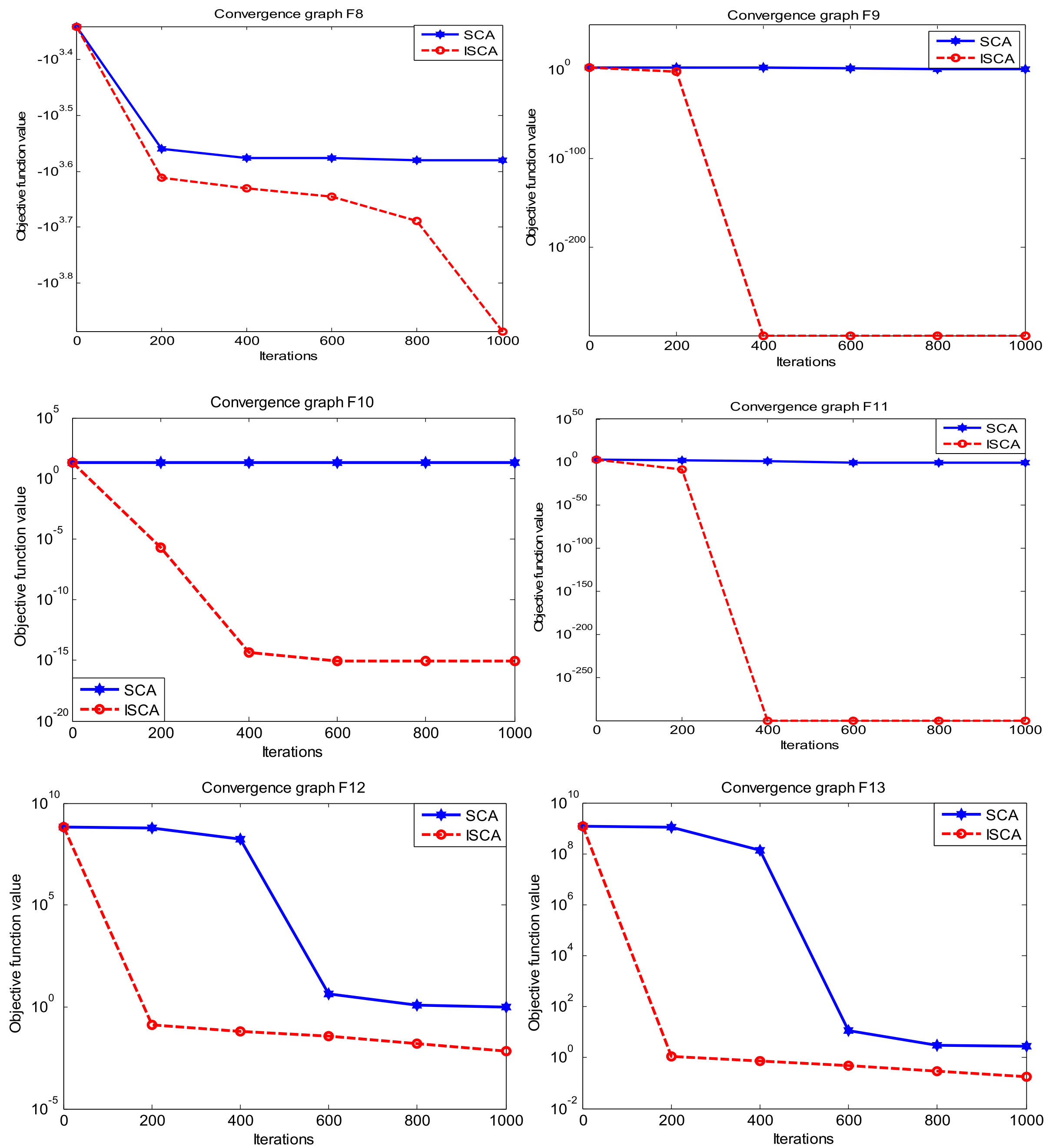

Fig. 5. Convergence corves for multimodal test problems.

of classical SCA as compared to ISCA. '=' is used when both the algorithms are statistically same.

4.1.6. *Diversity analysis*

Since the proposed algorithm ISCA focuses on the reduction of high exploration present in the classical SCA. Therefore, the diversity analysis between the search agents is very important to observe the significance of proposed strategies in ISCA. The diversity plot in terms of average distance between the solutions in each generation is plotted in Figs. 8 to 10. The Euclidean distance $\|.\|$ between two solutions $X = (x_1, x_2, \ldots, x_d)$ and $Y = (y_1, y_2, \ldots, y_d)$ is calculated as follows —

$$||X - Y||_2 = \sqrt{\sum_{j=1}^{d}(x_j - y_j)^2}$$

From the figures, it can be observed that the distance between solutions in ISCA is less than the classical SCA which shows that the search agents are utilizing the best information which is available in the population of agents. The figures also illustrate that the proposed algorithm is able to alleviate from the problem of high exploration during the search process.

4.1.7. *Performance Index (PI) analysis*

To compare the algorithms, classical SCA and proposed ISCA, by giving weights to the error value, success rate and CPU execution time, Performance Index (PI) can be calculated [77]. The relative performance of an algorithm using *PI* can be obtained as follows —

$$PI = \frac{1}{N_P}\sum_{P=1}^{N_P} u_1\alpha_P + u_2\beta_P + u_3\delta_P$$

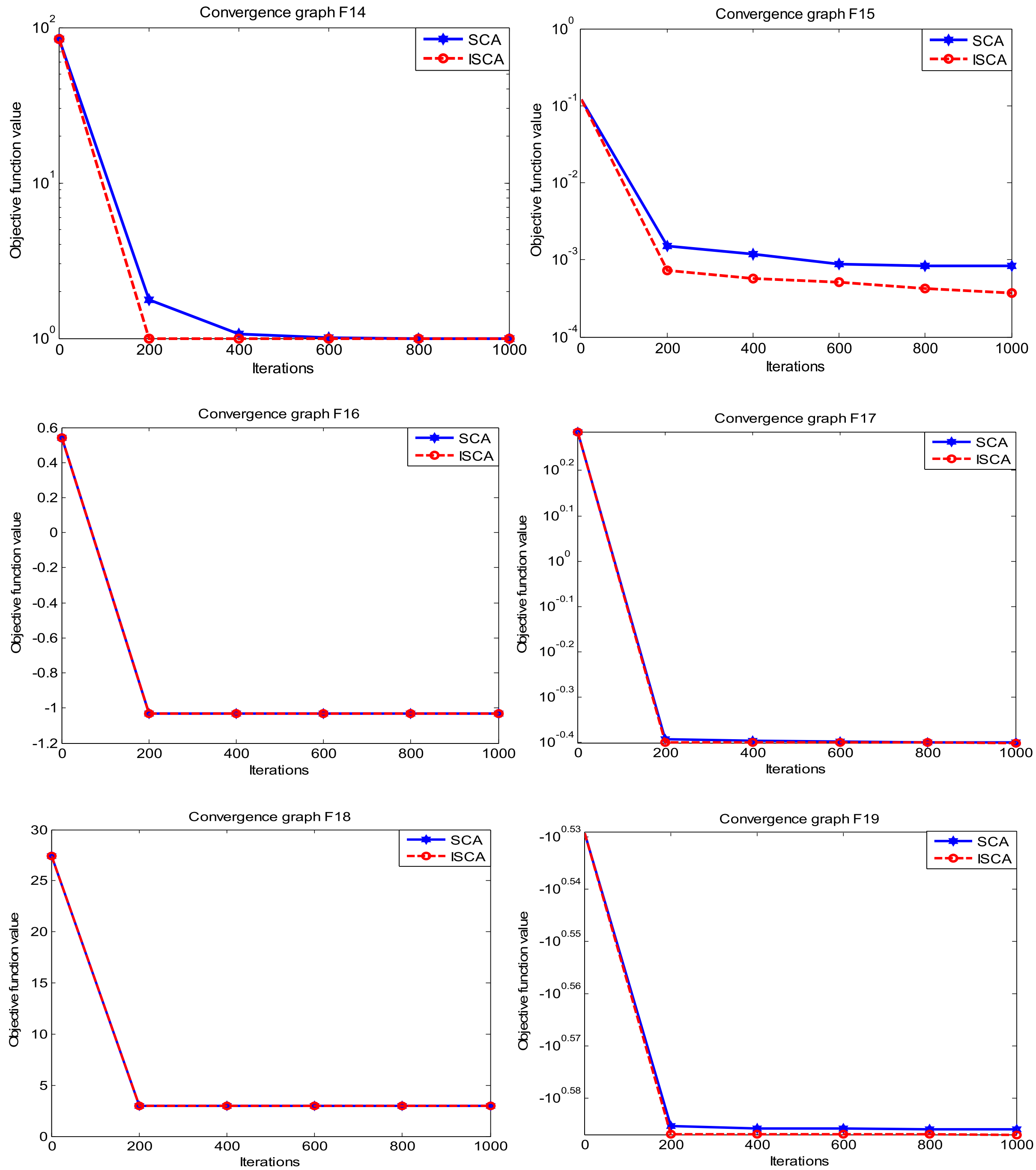

Fig. 6. Convergence corves for fixed dimensional multimodal test problems.

where,

$$\alpha_P = \frac{ME^P}{AE^P} \qquad \beta_P = \frac{SR^P}{TR^P} \qquad \delta_P = \frac{MT^P}{AT^P}$$

N_P — Total number of considered problems
ME^P — Minimum of average error achieved by all the algorithms to obtain a solution of the problem P
AE^P — Average error achieved by an algorithm to obtain a solution of the problem P
SR^P — Number of successful runs while solving the problem$P$
TR^P — Total number of runs for the problem P
MT^P — Minimum of average execution time used by all the algorithms to obtain a solution of the problem P
AT^P — Average execution time used by an algorithm to obtain a solution of the problem P.

u_1, u_2 and u_3 are non-negative weights associated with error term α_P, success rate β_P and percentage of computational time δ_P such that $u_1+u_2+u_3 = 1$. In our study, three different cases to analyze the proposed algorithm, are considered:

Case A: $u_1 = w$, $u_2 = \frac{(1-w)}{2}$ and $u_3 = \frac{(1-w)}{2}$, $0 \le w \le 1$
Case B: $u_1 = \frac{(1-w)}{2}$, $u_2 = w$ and $u_3 = \frac{(1-w)}{2}$, $0 \le w \le 1$
Case C: $u_1 = \frac{(1-w)}{2}$, $u_2 = \frac{(1-w)}{2}$ and $u_3 = w$, $0 \le w \le 1$

The PI graphs corresponding to each case are presented in Fig. 11. In these figures, weights u_1, u_2 and u_3 in terms of w are depicted on horizontal axis and the corresponding PI is shown on vertical axis.

In case A, success rate and execution time are equally weighted. It can be analyzed from the Fig. 11(a) that PI of ISCA is much higher than classical SCA. In case B, an equal weight is assigned to the

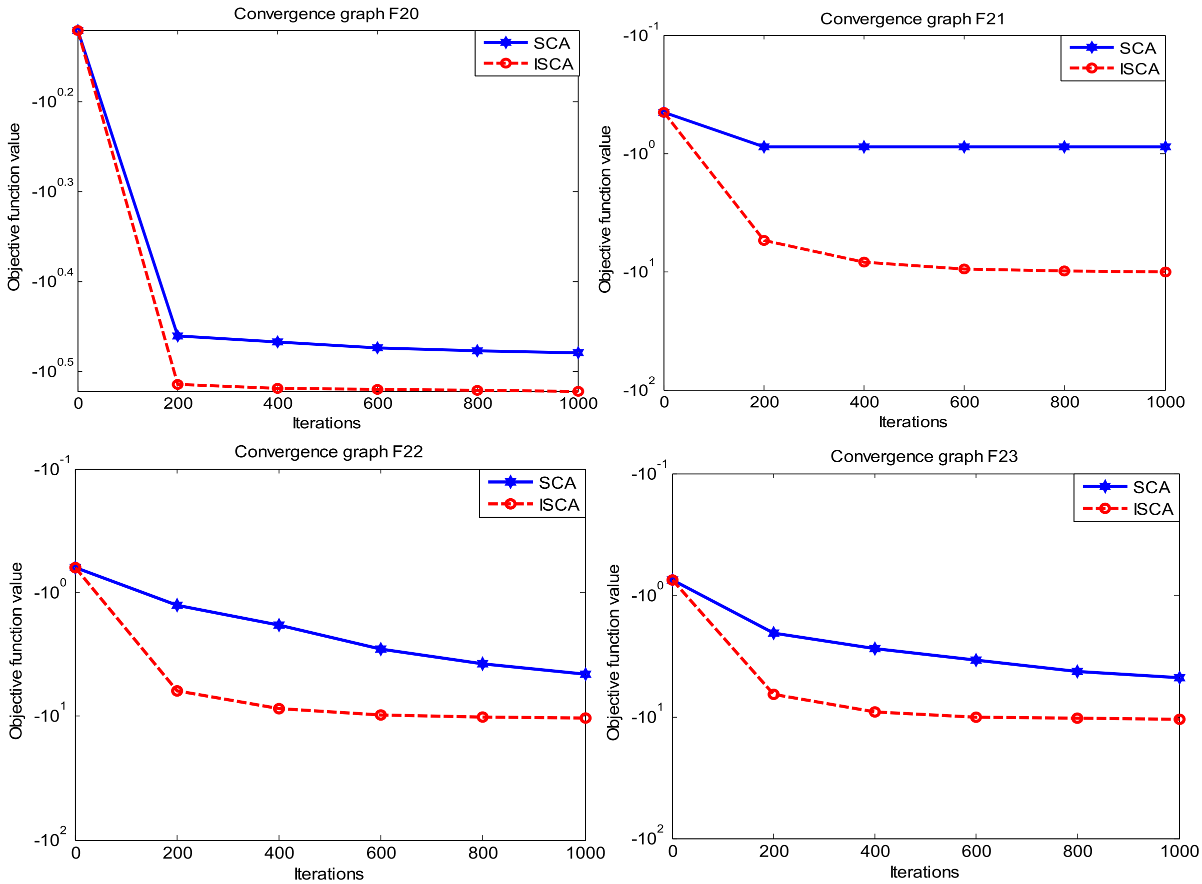

Fig. 7. Convergence corves for fixed dimensional multimodal test problems.

Table 6
Results of Wilcoxon signed rank test on classical set of benchmarks at 5% level of significance.

Test problem	*p*-value	Decision	Test problem	*p*-value	Decision
F1	1.7344E−06	+	F13	1.7344E−06	+
F2	1.7344E−06	+	F14	1.7344E−06	+
F3	1.7344E−06	+	F15	2.2248E−04	+
F4	1.7344E−06	+	F16	1.7344E−06	+
F5	1.7344E−06	+	F17	1.7344E−06	+
F6	1.7344E−06	+	F18	1.7344E−06	+
F7	1.7344E−06	+	F19	1.7344E−06	+
F8	1.7344E−06	+	F20	1.7344E−06	+
F9	1.7344E−06	+	F21	1.7344E−06	+
F10	1.7344E−06	+	F22	1.7344E−06	+
F11	1.7344E−06	+	F23	1.7344E−06	+
F12	1.7344E−06	+			

execution time and error term while in Case C, the success rate and error term are equally weighted. From the Fig. 11(b) and 11(c), it can be analyzed that PI of the proposed algorithm ISCA is higher than classical SCA except in case C when the user's consideration is only time not the quality of the solution. Overall, in terms of error, a number of successful runs and the elapsed time in finding the solution, the proposed ISCA outperforms classical SCA and it can be recommended over classical SCA when the users' requirement are less error, less execution time and successful runs simultaneously.

4.1.8. Comparison with other search algorithms

In this section, the performance of the proposed improved sine cosine algorithm has been compared with other optimization algorithms on the classical set of benchmark problems. The results in terms of objective function values from various other techniques are reported in Table 7. In all the algorithms population size is taken as 30 and the maximum number of iterations are fixed as 1000 to avoid the biasedness. In the table, the results of ISCA has been compared with Particle Swarm Optimization (PSO) [19], Whale Optimization Algorithm (WOA) [26], Moth-flame Optimization (MFO) [25], Salp Swarm Algorithm (SSA) [78], modified-SCA (m-SCA) [49], OBSCA [45] and classical SCA [30] based on the average objective function value. From the table, it can be observed that the proposed ISCA is very competitive in obtaining the solution as compared to other algorithms.

4.2. Benchmark problem set II

In the second set of benchmarks, test problems of IEEE CEC 2014 have been considered. In the standard IEEE CEC 2014 benchmark set, the problems are more difficult than the classical set of benchmarks. In these problems the problems are categorized into four groups — (*i*) Unimodal, (*ii*) Multimodal (*iii*) hybrid, and (*iv*) composite problems. The dimensions of these test problems are fixed as 10 and 30 in the present work. These problems can be found in detail in [61]. For the IEEE CEC 2014 benchmark problems, the termination criteria is fixed as $10^4 \times d$ function evaluations (d represent the dimension of the problem) as per the guidelines of IEEE CEC 2014. The obtained results on these test problems corresponding to the dimension 10 and 30 are presented in Tables 8 and 9.

4.2.1. Analysis of results and discussion

Unimodal problems

In the IEEE CEC 2014 benchmark set, the problems F1 to F3 are unimodal. The obtained results on these problems which are

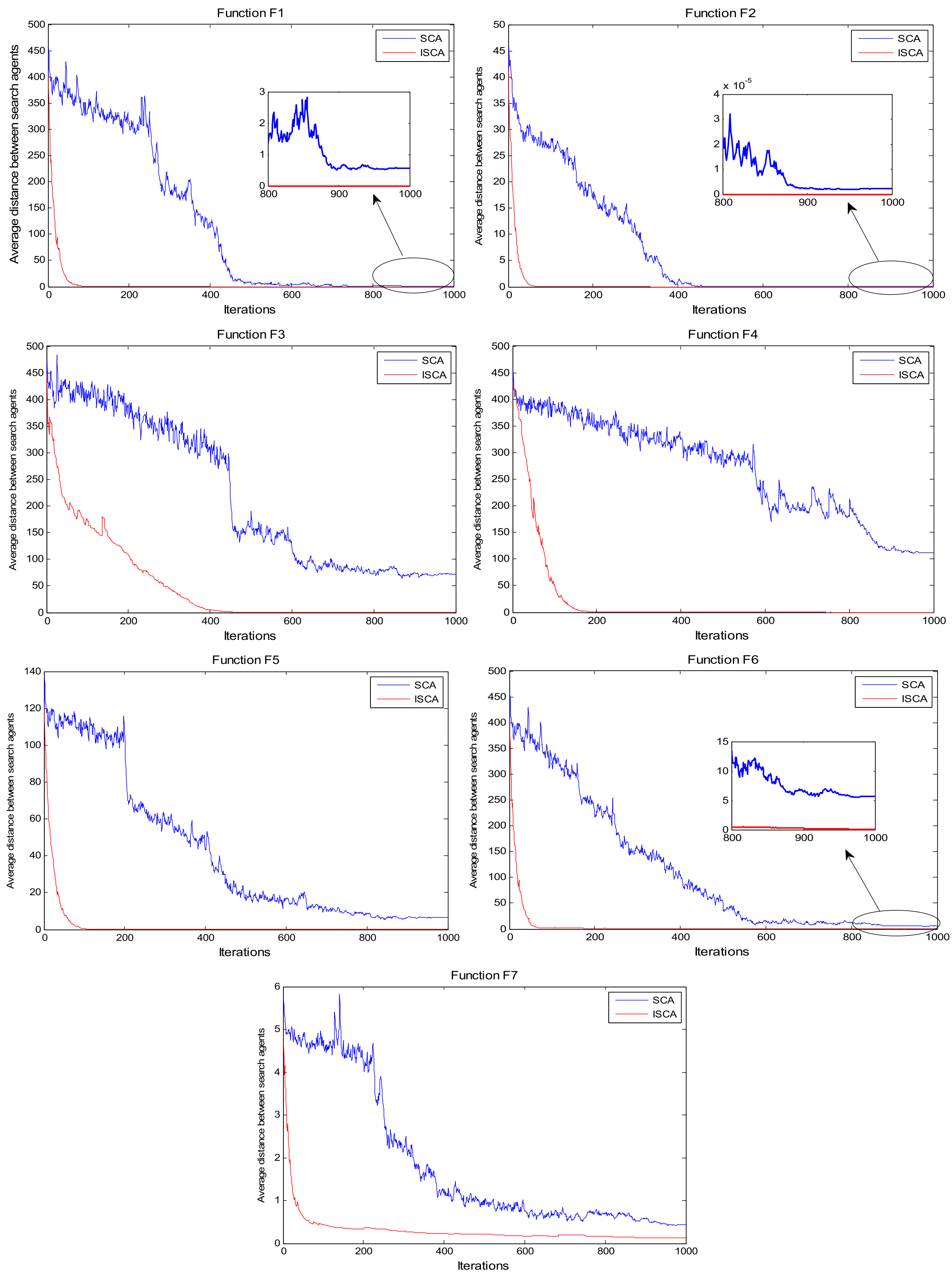

Fig. 8. Diversity plot for unimodal test problems (F1–F7).

presented in Tables 8 and 9 ensures the capability of proposed ISCA in exploiting the search regions of promising search areas. In both the dimension 10 and 30, ISCA outperforms classical SCA in all the statistics. Thus the incorporation of best memory, crossover and greedy selection proves their efficacy in exploiting the search regions of the problem.

Multimodal problems

Problems F4 to F16 in IEEE CEC 2014 set, are multimodal problems. The results on these problems ensure the ability of enhanced

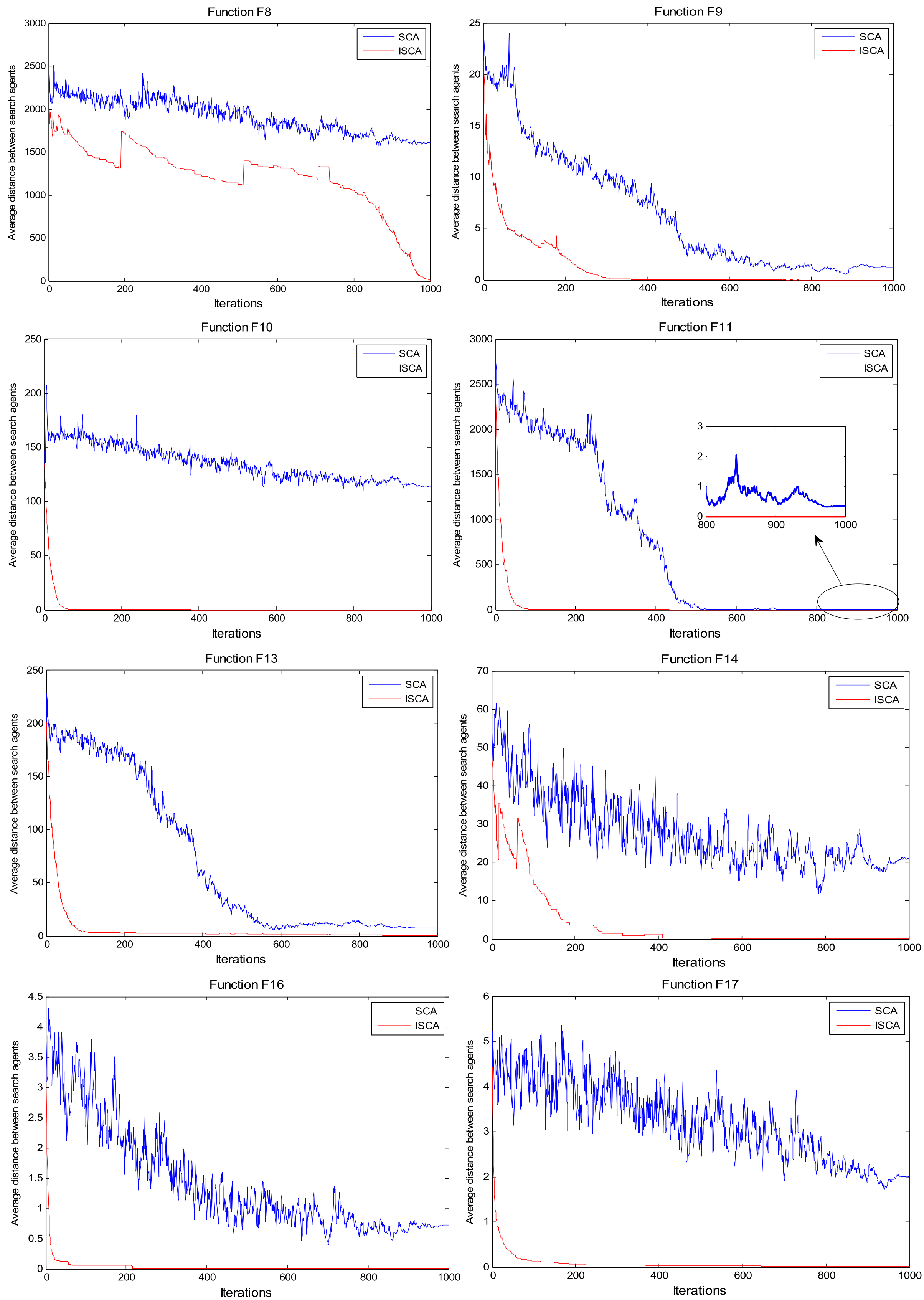

Fig. 9. Diversity plot for multimodal test problems (F8–F17).

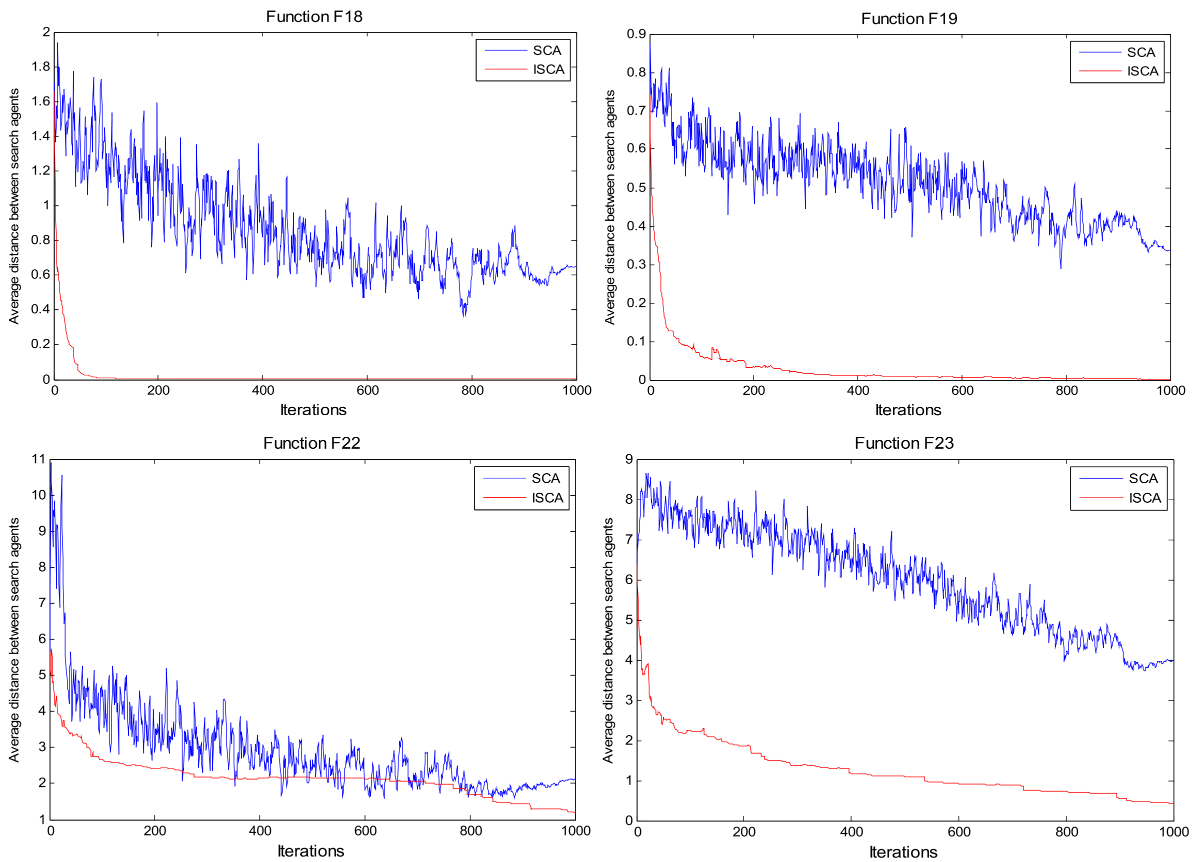

Fig. 10. Diversity plot for multimodal test problems (F18–F23).

Table 7
Performance comparison of proposed ISCA with other algorithms on classical set of benchmark problems.

Test problem	SCA	PSO	WOA	MFO	SSA	m-SCA	OBSCA	ISCA
F1	3.83E−02	2.12E+01	1.40E−148	2.33E+03	1.24E−08	8.25E−02	1.88E−18	2.01E−58
F2	1.89E−05	2.44E+00	9.90E−102	3.83E+01	1.54E+00	2.13E−05	3.66E−17	1.87E−36
F3	3.66E+03	9.55E+01	2.45E+04	1.75E+04	2.58E+02	3.84E+03	1.35E+01	2.52E−04
F4	2.19E+01	8.69E+00	4.17E+01	6.82E+01	8.73E+00	2.11E+01	7.07E−01	2.07E−15
F5	3.18E+02	5.02E+03	2.72E+01	5.34E+06	1.38E+02	6.01E+03	2.83E+01	2.61E+01
F6	4.63E+00	3.10E+01	4.06E−02	2.00E+03	1.39E−08	4.64E+00	4.60E+00	1.56E−01
F7	3.47E−02	1.42E−02	1.72E−03	3.99E+00	1.04E−01	6.95E−02	2.15E−03	1.13E−03
F8	−3.80E+03	−3.99E+03	−1.15E+04	−8.67E+03	−7.49E+03	−3.95E+03	−3.76E+03	−7.67E+03
F9	1.04E+01	2.48E+01	0.00E+00	1.75E+02	5.79E+01	2.07E+01	5.26E−11	0.00E+00
F10	1.72E+01	5.19E+00	4.80E−15	1.75E+01	2.04E+00	1.64E+01	1.27E+01	8.88E−16
F11	2.77E−01	1.25E+00	0.00E+00	2.11E+01	8.62E−03	4.13E−01	4.34E−11	0.00E+00
F12	1.53E+00	5.76E+00	3.01E−03	7.33E−01	5.58E+00	1.58E+03	4.94E−01	1.00E−02
F13	6.72E+02	9.46E+00	2.82E−01	4.40E−01	2.58E+00	4.45E+03	2.46E+00	1.69E−01
F14	1.66E+00	1.30E+00	3.35E+00	2.77E+00	1.06E+00	1.40E+00	2.47E+00	9.98E−01
F15	9.40E−04	5.90E−03	4.50E−04	1.50E−03	2.11E−03	9.84E−04	9.91E−04	1.12E−03
F16	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00
F17	3.99E−01	3.99E−01	3.98E−01	3.98E−01	3.98E−01	3.99E−01	3.99E−01	3.98E−01
F18	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F19	−3.86E+00	3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00
F20	−2.81E+00	−3.27E+00	−3.29E+00	−3.23E+00	−3.22E+00	−3.11E+00	−3.09E+00	−3.29E+00
F21	−2.73E+00	−8.90E+00	−9.64E+00	−7.23E+00	−7.73E+00	−5.18E+00	−9.80E+00	−1.02E+01
F22	−3.57E+00	−9.64E+00	−8.44E+00	−8.09E+00	−9.72E+00	−5.10E+00	−1.02E+01	−1.04E+01
F23	−4.28E+00	−8.14E+00	−8.84E+00	−8.96E+00	−9.64E+00	−5.20E+00	−1.03E+01	−1.05E+01

exploration in proposed ISCA. In terms of minimum, median, maximum and average error the proposed algorithm outperforms the classical SCA in all the problems corresponding to dimension 10 and 30. In the problems F5, F11 and F16 corresponding to dimension 30 the standard deviation is slightly higher in ISCA as compared to classical SCA. The integration of personal best strategy in place of global best and integration of global best direction with random steps confirms the ability of exploration in ISCA. Thus the proposed algorithm can be considered more efficient than classical SCA in terms of exploration strength.

Hybrid and composite problems

The hybrid and composite problems evaluate the ability of exploration along with the strength of maintaining a suitable balance between exploration and exploitation. In the IEEE CEC 2014 benchmark set, F16 to F22 are hybrid functions and F23 to F30 are composite functions. From the results presented in Tables 8 and 9 on these problems, it can be observed that the proposed algorithm

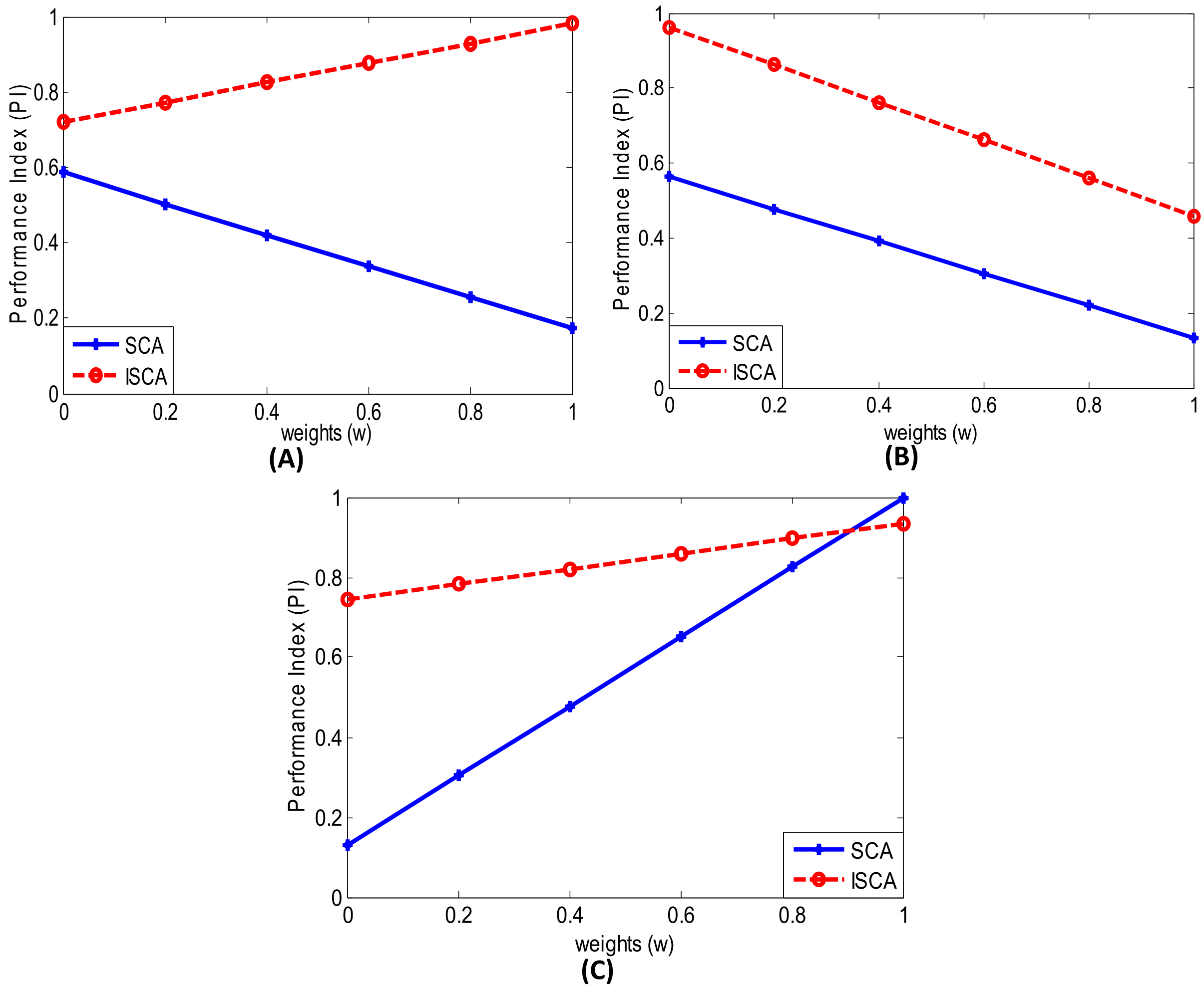

Fig. 11. Performance Index (PI) corresponding to the case (A), (B) and (C).

has shown its enhanced ability of global search with skipping of local solutions that cause the premature convergence.

Therefore on analyzing the performance on standard IEEE CEC 2014 benchmark test suite, it can be concluded that the proposed ISCA is more reliable and efficient optimizer as compared to the classical SCA.

4.2.2. Statistical analysis of the results on CEC 2014

To ensure the improvement in ISCA, Wilcoxon signed rank test has been applied with the same setting as in the classical set of benchmark test problems and the obtained statistical conclusions are presented in Tables 10 and 11 with same symbols as described in Section 4.1.5. From the significant improvement proved by a statistical test, it can be concluded that the proposed algorithm ISCA improves the search ability of solutions in terms of escaping from local solutions and enhancing the global search ability during the search process.

4.2.3. Comparison with other search algorithms

In this section, the performance of the proposed improved sine cosine algorithm has been compared with other optimization algorithms on 30-dimensional standard IEEE CEC 2014 of benchmark test set. The results from various other techniques are reported in Table 12. In all the algorithms population size is taken as 30 and $10^4 \times dimension of the problem$, function evaluations as per the guidelines of CEC 2014 are fixed for each algorithm to avoid the biasedness. In the table the results of ISCA has been compared with Particle Swarm Optimization (PSO) [19], Whale Optimization Algorithm (WOA) [26], Moth-flame Optimization (MFO) [25], Salp Swarm Algorithm (SSA) [78], modified-SCA (m-SCA) [49], OB-SCA [45] and classical SCA [30] based on the error value in average fitness. From the table, it can be observed that the proposed ISCA is very competitive in obtaining the solution as compared to other algorithms.

4.3. Benchmark set III: IEEE CEC 2017

In this section, the latest benchmark set of IEEE CEC 2017 [62] is used to examine the performance of the proposed strategy in ISCA. In this benchmark set, the problems are categorized into four groups similar to CEC 2014 benchmarks as — (i) Unimodal (F1–F3), (ii) Multimodal (F4–F10) (iii) hybrid (F11–F20), and (iv) composite problems (F21–F30). The dimensions of these test problems are fixed as 10 and 30 in the present work. These problems can be found in detail in [62]. For the IEEE CEC 2017 benchmark problems, the termination criteria is fixed as $10^4 \times d$ function evaluations as per the guidelines of IEEE CEC 2017. The obtained results on these test problems corresponding to the dimension 10 and 30 are presented in Tables 13 and 14. Since the problem F2 has been deleted from the benchmark set due to the unstable behavior, the results are not reported for problem F2. In the table, the results are presented in the format of CEC 2017. From the obtained results it can be seen that in all the problems corresponding to the dimensions 10 and 30 the proposed algorithm outperforms classical SCA in most of the test statistics.

Table 8
Comparison of results between classical SCA and proposed ISCA on 10 dimensional IEEE CEC 2014 benchmark test problems.

Test function	Algorithm	Best	Mean	Median	Worst	STD
F1	SCA	5.459E+06	2.059E+07	1.537E+07	8.095E+07	1.417E+07
	ISCA	**2.362E+04**	**3.469E+05**	**2.142E+05**	**3.483E+06**	**5.262E+05**
F2	SCA	3.541E+08	3.013E+09	2.249E+09	8.897E+09	2.150E+09
	ISCA	**1.346E+03**	**6.017E+03**	**3.386E+03**	**1.550E+04**	**4.696E+03**
F3	SCA	1.965E+03	2.121E+04	1.521E+04	7.258E+04	1.794E+04
	ISCA	**1.447E+01**	**2.761E+02**	**1.200E+02**	**1.090E+03**	**2.834E+02**
F4	SCA	6.395E+01	3.875E+02	2.418E+02	1.548E+03	3.310E+02
	ISCA	**6.555E−01**	**2.757E+01**	**3.502E+01**	**3.522E+01**	**1.295E+01**
F5	SCA	2.020E+01	2.041E+01	2.039E+01	2.065E+01	**1.032E−01**
	ISCA	**2.311E−01**	**1.982E+01**	**2.021E+01**	**2.034E+01**	2.799E+00
F6	SCA	6.053E+00	9.759E+00	1.001E+01	1.146E+01	1.220E+00
	ISCA	**1.518E−01**	**1.042E+00**	**7.873E−01**	**3.531E+00**	**7.814E−01**
F7	SCA	6.657E+00	6.615E+01	7.577E+01	1.188E+02	3.186E+01
	ISCA	**1.174E−01**	**4.211E−01**	**3.632E−01**	**1.198E+00**	**1.976E−01**
F8	SCA	4.273E+01	7.405E+01	7.105E+01	1.089E+02	1.583E+01
	ISCA	**2.939E−03**	**1.148E+00**	**1.007E+00**	**3.982E+00**	**8.587E−01**
F9	SCA	2.603E+01	7.949E+01	8.361E+01	1.181E+02	2.111E+01
	ISCA	**2.991E+00**	**7.406E+00**	**6.444E+00**	**1.403E+01**	**2.743E+00**
F10	SCA	6.907E+02	1.317E+03	1.327E+03	1.762E+03	2.208E+02
	ISCA	**4.786E+00**	**4.119E+01**	**2.302E+01**	**1.561E+02**	**4.502E+01**
F11	SCA	1.197E+03	1.641E+03	1.719E+03	2.021E+03	2.516E+02
	ISCA	**1.283E+00**	**1.873E+02**	**1.435E+02**	**7.087E+02**	**1.648E+02**
F12	SCA	8.000E−01	1.529E+00	1.447E+00	2.648E+00	4.368E−01
	ISCA	**1.366E−01**	**3.520E−01**	**3.450E−01**	**6.027E−01**	**1.223E−01**
F13	SCA	5.736E−01	2.804E+00	2.848E+00	4.811E+00	1.245E+00
	ISCA	**8.081E−02**	**1.697E−01**	**1.642E−01**	**3.121E−01**	**4.603E−02**
F14	SCA	6.997E−01	1.447E+01	1.480E+01	2.767E+01	7.299E+00
	ISCA	**6.698E−02**	**1.868E−01**	**1.903E−01**	**6.066E−01**	**8.025E−02**
F15	SCA	6.488E+00	1.111E+04	7.919E+03	4.674E+04	1.230E+04
	ISCA	**6.302E−01**	**1.341E+00**	**1.248E+00**	**2.289E+00**	**4.163E−01**
F16	SCA	2.916E+00	3.903E+00	3.982E+00	4.368E+00	**3.164E−01**
	ISCA	**3.537E−01**	**1.802E+00**	**1.808E+00**	**2.834E+00**	5.730E−01
F17	SCA	3.824E+03	3.436E+05	1.709E+05	1.884E+06	3.965E+05
	ISCA	**1.666E+02**	**2.684E+03**	**1.882E+03**	**7.947E+03**	**2.346E+03**
F18	SCA	8.457E+03	2.810E+06	5.832E+05	3.874E+07	6.492E+06
	ISCA	**4.775E+01**	**4.924E+03**	**3.124E+03**	**1.619E+04**	**5.025E+03**
F19	SCA	6.497E+00	1.125E+01	1.105E+01	2.267E+01	3.549E+00
	ISCA	**1.153E+00**	**1.592E+00**	**1.570E+00**	**2.779E+00**	**2.363E−01**
F20	SCA	3.781E+02	1.610E+05	5.155E+04	1.102E+06	2.421E+05
	ISCA	**7.170E+00**	**3.816E+02**	**3.759E+01**	**4.370E+03**	**9.610E+02**
F21	SCA	1.888E+03	1.227E+05	5.032E+04	8.773E+05	1.898E+05
	ISCA	**5.586E+01**	**1.406E+03**	**4.046E+02**	**5.182E+03**	**1.697E+03**
F22	SCA	5.395E+01	2.265E+02	2.323E+02	5.035E+02	1.115E+02
	ISCA	**1.273E+00**	**1.693E+01**	**2.156E+01**	**4.019E+01**	**9.660E+00**
F23	SCA	3.400E+02	4.319E+02	4.301E+02	5.361E+02	4.622E+01
	ISCA	**3.295E+02**	**3.295E+02**	**3.295E+02**	**3.296E+02**	**2.292E−02**
F24	SCA	1.476E+02	2.015E+02	2.028E+02	2.374E+02	2.430E+01
	ISCA	**1.074E+02**	**1.167E+02**	**1.134E+02**	**2.000E+02**	**1.349E+01**
F25	SCA	1.696E+02	2.064E+02	2.074E+02	2.220E+02	**7.826E+00**
	ISCA	**1.206E+02**	**1.833E+02**	**1.988E+02**	**2.022E+02**	2.836E+01
F26	SCA	1.005E+02	1.025E+02	1.024E+02	1.038E+02	8.564E−01
	ISCA	**1.001E+02**	**1.002E+02**	**1.002E+02**	**1.003E+02**	**3.932E−02**
F27	SCA	1.231E+02	4.449E+02	4.608E+02	6.110E+02	**1.160E+02**
	ISCA	**2.376E+00**	**2.788E+02**	**3.450E+02**	**4.013E+02**	1.663E+02
F28	SCA	4.139E+02	5.485E+02	5.487E+02	7.389E+02	7.743E+01
	ISCA	**2.036E+02**	**4.157E+02**	**3.777E+02**	**5.069E+02**	**6.106E+01**
F29	SCA	5.593E+03	1.518E+05	4.721E+04	**1.832E+06**	**3.139E+05**
	ISCA	**2.501E+02**	**1.092E+05**	**4.012E+02**	2.101E+06	4.417E+05
F30	SCA	1.290E+03	8.130E+03	4.876E+03	4.709E+04	9.535E+03
	ISCA	**4.839E+02**	**6.709E+02**	**5.935E+02**	**2.083E+03**	**2.721E+02**

Table 9
Comparison of results between classical SCA and proposed ISCA on 30 dimensional IEEE CEC 2014 benchmark test problems.

Test function	Algorithm	Best	Mean	Median	Worst	STD
F1	SCA	1.586E+08	6.275E+08	6.354E+08	1.845E+09	3.005E+08
	ISCA	**5.807E+06**	**1.427E+07**	**1.317E+07**	**3.714E+07**	**5.937E+06**
F2	SCA	1.780E+10	5.010E+10	4.803E+10	1.270E+11	2.402E+10
	ISCA	**9.697E+06**	**3.131E+08**	**2.576E+08**	**7.441E+08**	**2.073E+08**
F3	SCA	3.155E+04	1.079E+05	1.003E+05	2.865E+05	6.483E+04
	ISCA	**4.419E+02**	**2.629E+03**	**2.061E+03**	**6.568E+03**	**1.682E+03**
F4	SCA	9.362E+02	9.413E+03	7.112E+03	2.335E+04	7.025E+03
	ISCA	**9.641E+01**	**1.466E+02**	**1.492E+02**	**2.207E+02**	**2.550E+01**
F5	SCA	2.077E+01	2.098E+01	2.098E+01	2.111E+01	**5.704E−02**
	ISCA	**2.058E+01**	**2.086E+01**	**2.088E+01**	**2.098E+01**	9.131E−02
F6	SCA	3.219E+01	4.025E+01	4.071E+01	4.454E+01	2.739E+00
	ISCA	**3.922E+00**	**8.333E+00**	**8.067E+00**	**1.375E+01**	**1.826E+00**
F7	SCA	1.383E+02	6.639E+02	7.208E+02	1.056E+03	2.626E+02
	ISCA	**1.248E+00**	**3.731E+00**	**3.317E+00**	**9.813E+00**	**1.859E+00**
F8	SCA	2.393E+02	3.877E+02	4.061E+02	4.892E+02	8.038E+01
	ISCA	**1.678E+01**	**2.937E+01**	**2.932E+01**	**4.460E+01**	**6.929E+00**
F9	SCA	2.519E+02	4.528E+02	5.072E+02	5.926E+02	1.106E+02
	ISCA	**3.036E+01**	**6.115E+01**	**6.008E+01**	**1.015E+02**	**1.526E+01**
F10	SCA	5.716E+03	7.115E+03	7.197E+03	7.921E+03	5.040E+02
	ISCA	**1.696E+02**	**6.770E+02**	**6.631E+02**	**1.441E+03**	**2.522E+02**
F11	SCA	6.438E+03	7.634E+03	7.696E+03	8.645E+03	**4.992E+02**
	ISCA	**9.705E+02**	**2.427E+03**	**2.363E+03**	**3.736E+03**	6.261E+02
F12	SCA	1.911E+00	2.945E+00	2.909E+00	3.569E+00	4.072E−01
	ISCA	**8.848E−01**	**1.605E+00**	**1.559E+00**	**2.606E+00**	**3.519E−01**
F13	SCA	3.239E+00	7.604E+00	8.139E+00	1.068E+01	2.014E+00
	ISCA	**2.545E−01**	**3.454E−01**	**3.428E−01**	**5.400E−01**	**6.063E−02**
F14	SCA	4.346E+01	2.512E+02	2.592E+02	4.086E+02	9.466E+01
	ISCA	**1.962E−01**	**6.459E−01**	**7.432E−01**	**9.177E−01**	**2.178E−01**
F15	SCA	1.711E+03	5.709E+06	6.027E+06	1.208E+07	4.367E+06
	ISCA	**5.876E+00**	**1.327E+01**	**1.349E+01**	**1.862E+01**	**2.877E+00**
F16	SCA	1.267E+01	1.346E+01	1.352E+01	1.387E+01	**2.711E−01**
	ISCA	**8.831E+00**	**1.068E+01**	**1.079E+01**	**1.157E+01**	5.420E−01
F17	SCA	2.864E+06	4.382E+07	4.386E+07	9.857E+07	2.598E+07
	ISCA	**4.582E+04**	**4.538E+05**	**4.002E+05**	**2.233E+06**	**4.076E+05**
F18	SCA	1.042E+08	1.931E+09	1.891E+09	4.996E+09	1.039E+09
	ISCA	**7.449E+02**	**5.833E+03**	**2.479E+03**	**8.107E+04**	**1.197E+04**
F19	SCA	9.053E+01	4.406E+02	4.638E+02	8.194E+02	1.760E+02
	ISCA	**7.805E+00**	**1.185E+01**	**1.181E+01**	**1.637E+01**	**2.084E+00**
F20	SCA	1.539E+04	5.276E+05	2.505E+05	3.091E+06	6.879E+05
	ISCA	**3.154E+02**	**1.729E+03**	**1.331E+03**	**5.823E+03**	**1.401E+03**
F21	SCA	8.689E+05	1.456E+07	1.156E+07	4.463E+07	1.042E+07
	ISCA	**2.970E+04**	**1.579E+05**	**1.223E+05**	**6.854E+05**	**1.274E+05**
F22	SCA	8.854E+02	1.776E+03	1.781E+03	2.635E+03	3.934E+02
	ISCA	**4.060E+01**	**2.206E+02**	**1.928E+02**	**4.281E+02**	**8.724E+01**
F23	SCA	3.786E+02	1.075E+03	1.209E+03	1.807E+03	3.988E+02
	ISCA	**3.156E+02**	**3.177E+02**	**3.173E+02**	**3.207E+02**	**1.278E+00**
F24	SCA	2.019E+02	4.580E+02	5.127E+02	5.523E+02	1.107E+02
	ISCA	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**1.680E−03**
F25	SCA	2.210E+02	2.939E+02	2.950E+02	3.566E+02	3.228E+01
	ISCA	**2.000E+02**	**2.064E+02**	**2.067E+02**	**2.101E+02**	**2.236E+00**
F26	SCA	1.031E+02	1.075E+02	1.078E+02	**1.104E+02**	**1.622E+00**
	ISCA	**1.002E+02**	**1.062E+02**	**1.003E+02**	2.000E+02	2.368E+01
F27	SCA	5.604E+02	1.068E+03	1.111E+03	1.243E+03	1.826E+02
	ISCA	**4.033E+02**	**4.736E+02**	**4.706E+02**	**6.243E+02**	**6.618E+01**
F28	SCA	2.271E+03	3.453E+03	3.380E+03	4.709E+03	5.709E+02
	ISCA	**6.712E+02**	**8.718E+02**	**8.692E+02**	**1.027E+03**	**5.716E+01**
F29	SCA	9.935E+06	6.848E+07	7.313E+07	1.154E+08	2.427E+07
	ISCA	**6.074E+03**	**2.254E+05**	**1.667E+04**	**1.066E+07**	**1.491E+06**
F30	SCA	4.129E+05	1.299E+06	1.243E+06	4.116E+06	6.792E+05
	ISCA	**3.974E+03**	**8.049E+03**	**7.827E+03**	**1.389E+04**	**2.631E+03**

Table 10
Statistical decision based on Wilcoxon Signed rank test at 5% level of significance on 10 dimensional problems of IEEE CEC 2014.

Test function	p-value	Decision	Test function	p-value	Decision
F1	5.145E−10	+	F16	5.145E−10	+
F2	5.145E−10	+	F17	5.145E−10	+
F3	5.145E−10	+	F18	5.462E−10	+
F4	5.145E−10	+	F19	5.145E−10	+
F5	6.528E−10	+	F20	5.145E−10	+
F6	5.145E−10	+	F21	5.462E−10	+
F7	5.145E−10	+	F22	5.145E−10	+
F8	5.145E−10	+	F23	5.145E−10	+
F9	5.145E−10	+	F24	5.145E−10	+
F10	5.145E−10	+	F25	7.433E−08	+
F11	5.145E−10	+	F26	5.145E−10	+
F12	5.145E−10	+	F27	3.289E−05	+
F13	5.145E−10	+	F28	1.486E−09	+
F14	5.145E−10	+	F29	1.520E−06	+
F15	5.145E−10	+	F30	5.145E−10	+

Table 11
Statistical decision based on Wilcoxon Signed rank test at 5% level of significance on 30 dimensional problems of IEEE CEC 2014.

Test function	p-value	Decision	Test function	p-value	Decision
F1	5.145E−10	+	F16	5.145E−10	+
F2	5.145E−10	+	F17	5.145E−10	+
F3	5.145E−10	+	F18	5.145E−10	+
F4	5.145E−10	+	F19	5.145E−10	+
F5	2.531E−07	+	F20	5.145E−10	+
F6	5.145E−10	+	F21	5.145E−10	+
F7	5.145E−10	+	F22	5.145E−10	+
F8	5.145E−10	+	F23	5.145E−10	+
F9	5.145E−10	+	F24	5.145E−10	+
F10	5.145E−10	+	F25	5.145E−10	+
F11	5.145E−10	+	F26	1.520E−06	+
F12	5.462E−10	+	F27	5.462E−10	+
F13	5.145E−10	+	F28	5.145E−10	+
F14	5.145E−10	+	F29	5.145E−10	+
F15	5.145E−10	+	F30	5.145E−10	+

To ensure the significant improvement in results and to prove that the improved results are not just by chance, Wilcoxon signed rank test is applied between classical SCA and proposed algorithm (ISCA). The obtained statistical conclusions with p-values are presented in Table 15 for 30-dimensional problems. From the table, it can be ensure that the proposed algorithm outperforms classical SCA in achieving the optima of the test problems.

4.3.1. Comparison with other search algorithms

In this section, the performance of the proposed algorithm has been compared with other optimization algorithms on 30 dimensional standard IEEE CEC 2017 benchmark test set. The results from various other techniques are reported in Table 16. In all the algorithms population size is taken as 30 and $10^4 \times d$, function evaluations are fixed as per the guidelines of CEC 2017. The same parameter setting is taken to avoid the biasedness in results. In the table the results of ISCA has been compared with Particle Swarm Optimization (PSO) [19], Whale Optimization Algorithm (WOA) [26], Moth-flame Optimization (MFO) [25], Salp Swarm Algorithm (SSA) [78], modified-SCA (m-SCA) [49], OBSCA [45] and classical SCA [30] based on the error value in average fitness. From the table, it can be observed that the proposed ISCA is very competitive than other algorithms.

From the experimental results and their analysis in terms of various metrics on three set of benchmarks — classical, CEC 2014 and CEC 2017 demonstrate that the proposed strategy is able to improve the search ability of agents in classical SCA. In terms of error value in objective fitness, the proposed algorithm outperforms classical SCA and compete some other variants of SCA and other algorithms. The achieved error in various benchmarks having diverse difficulty levels (unimodal, multimodal, hybrid and composite functions) demonstrate the better ability of proposed algorithm in terms of convergence rate, local optima avoidance and sustaining an appropriate balance between exploration and exploitation. In terms of the number of iterations, the proposed algorithm also shows better accuracy in obtaining solutions to the problem than classical SCA. Although, the results on CEC 2014

Table 12
Performance comparison of proposed ISCA with other meta-heuristic algorithms with 30 dimensional CEC2014 problems.

Problem	SCA	PSO	WOA	MFO	SSA	m-SCA	OBSCA	ISCA
F1	6.27E+08	3.27E+07	2.98E+07	7.73E+07	1.72E+06	2.31E+08	4.37E+08	1.43E+07
F2	5.01E+10	1.93E+09	5.28E+06	1.25E+10	1.06E+04	1.65E+10	2.03E+10	3.13E+08
F3	1.08E+05	7.12E+03	3.23E+04	9.86E+04	1.04E+03	3.77E+04	4.77E+04	2.63E+03
F4	9.41E+03	2.07E+02	1.82E+02	9.38E+02	9.00E+01	1.03E+03	1.33E+03	1.47E+02
F5	2.10E+01	2.09E+01	2.03E+01	2.03E+01	2.01E+01	2.09E+01	2.09E+01	2.09E+01
F6	4.02E+01	2.03E+01	3.54E+01	2.40E+01	1.90E+01	3.34E+01	2.99E+01	8.33E+00
F7	6.64E+02	2.22E+01	1.02E+00	1.06E+02	1.40E−02	1.18E+02	1.54E+02	3.73E+00
F8	3.88E+02	1.92E+02	1.90E+02	1.49E+02	1.02E+02	2.36E+02	2.46E+02	2.94E+01
F9	4.53E+02	1.87E+02	2.35E+02	2.14E+02	1.21E+02	2.74E+02	2.94E+02	6.12E+01
F10	7.11E+03	5.57E+03	3.81E+03	3.47E+03	3.49E+03	5.99E+03	4.11E+03	6.77E+02
F11	7.63E+03	6.22E+03	4.84E+03	4.13E+03	3.69E+03	7.01E+03	5.55E+03	2.43E+03
F12	2.94E+00	2.50E+00	1.79E+00	4.63E−01	4.91E−01	2.46E+00	1.71E+00	1.61E+00
F13	7.60E+00	5.52E−01	5.18E−01	2.01E+00	4.91E−01	3.04E+00	2.96E+00	3.45E−01
F14	2.51E+02	2.90E+00	2.68E−01	2.67E+01	2.99E−01	4.59E+01	4.54E+01	6.46E−01
F15	5.71E+06	3.35E+01	7.72E+01	1.94E+05	8.78E+00	1.97E+03	1.45E+04	1.33E+01
F16	1.35E+01	1.19E+01	1.27E+01	1.28E+01	1.17E+01	1.28E+01	1.25E+01	1.07E+01
F17	4.38E+07	8.63E+05	3.46E+06	4.31E+06	1.02E+05	5.89E+06	9.53E+06	4.54E+05
F18	1.93E+09	2.07E+07	6.24E+03	8.02E+06	6.64E+03	1.45E+08	1.66E+08	5.83E+03
F19	4.41E+02	2.28E+01	4.99E+01	8.91E+01	1.74E+01	9.24E+01	1.07E+02	1.18E+01
F20	5.28E+05	7.72E+02	2.54E+04	7.25E+04	3.69E+02	1.05E+04	1.71E+04	1.73E+03
F21	1.46E+07	2.55E+05	1.13E+06	1.85E+06	6.80E+04	1.33E+06	1.28E+06	1.58E+05
F22	1.78E+03	4.85E+02	8.37E+02	7.98E+02	3.78E+02	7.67E+02	7.71E+02	2.21E+02
F23	1.07E+03	3.32E+02	3.33E+02	3.86E+02	3.15E+02	3.65E+02	3.59E+02	3.18E+02
F24	4.58E+02	2.54E+02	2.06E+02	2.93E+02	2.41E+02	2.00E+02	2.00E+02	2.00E+02
F25	2.94E+02	2.09E+02	2.16E+02	2.13E+02	2.11E+02	2.27E+02	2.24E+02	2.06E+02
F26	1.07E+02	1.01E+02	1.00E+02	1.07E+02	1.01E+02	1.02E+02	1.03E+02	1.06E+02
F27	1.07E+03	4.31E+02	1.04E+03	9.31E+02	6.96E+02	7.81E+02	4.95E+02	4.74E+02
F28	3.45E+03	1.67E+03	2.26E+03	1.16E+03	1.05E+03	2.01E+03	1.42E+03	8.72E+02
F29	6.85E+07	3.85E+05	4.30E+06	2.55E+06	1.73E+06	1.08E+07	6.73E+06	2.25E+05
F30	1.30E+06	2.87E+04	8.89E+04	2.96E+04	8.34E+03	2.42E+05	2.52E+05	8.05E+03

Table 13
Comparison of results between classical SCA and proposed ISCA on 10 dimensional IEEE CEC 2017 benchmark test problems.

Test problem	Algorithm	Minimum	Median	Mean	Maximum	STD
F1	SCA	2.874E+08	3.815E+09	3.579E+09	1.146E+10	2.396E+09
	ISCA	**4.848E+03**	**2.425E+04**	**1.577E+04**	**2.374E+05**	**3.523E+04**
F2	SCA	NA	NA	NA	NA	NA
	ISCA	NA	NA	NA	NA	NA
F3	SCA	4.197E+02	1.111E+04	9.549E+03	3.937E+04	9.220E+03
	ISCA	**2.285E−02**	**2.117E+01**	**1.049E+01**	**8.012E+01**	**2.231E+01**
F4	SCA	5.075E+01	2.721E+02	2.525E+02	9.467E+02	1.898E+02
	ISCA	**4.964E+00**	**7.276E+00**	**7.434E+00**	**7.742E+00**	**4.706E−01**
F5	SCA	3.368E+01	7.886E+01	8.337E+01	1.157E+02	1.913E+01
	ISCA	**2.003E+00**	**8.363E+00**	**7.979E+00**	**2.140E+01**	**3.960E+00**
F6	SCA	1.535E+01	4.596E+01	4.787E+01	7.655E+01	1.449E+01
	ISCA	**3.238E−02**	**1.209E−01**	**6.138E−02**	**1.100E+00**	**1.813E−01**
F7	SCA	6.860E+01	1.916E+02	1.782E+02	4.437E+02	9.948E+01
	ISCA	**1.393E+01**	**2.029E+01**	**1.980E+01**	**3.167E+01**	**4.265E+00**
F8	SCA	2.785E+01	8.242E+01	8.322E+01	1.358E+02	2.648E+01
	ISCA	**3.272E+00**	**7.054E+00**	**6.972E+00**	**1.276E+01**	**2.309E+00**
F9	SCA	5.858E+01	1.441E+03	1.465E+03	3.461E+03	9.893E+02
	ISCA	**2.023E−02**	**3.602E−01**	**4.804E−01**	**9.233E−01**	**2.423E−01**
F10	SCA	7.538E+02	1.546E+03	1.550E+03	1.987E+03	2.502E+02
	ISCA	**1.083E+00**	**2.174E+02**	**2.482E+02**	**4.833E+02**	**1.415E+02**
F11	SCA	4.651E+01	1.130E+03	9.736E+02	3.881E+03	8.667E+02
	ISCA	**1.987E+00**	**7.277E+00**	**7.229E+00**	**1.476E+01**	**2.980E+00**
F12	SCA	5.697E+06	2.156E+08	1.681E+08	6.561E+08	1.698E+08
	ISCA	**4.520E+03**	**9.202E+04**	**2.406E+04**	**1.371E+06**	**2.604E+05**
F13	SCA	2.181E+03	2.008E+06	5.884E+05	1.523E+07	3.243E+06
	ISCA	**8.928E+01**	**1.180E+03**	**4.016E+02**	**8.614E+03**	**1.762E+03**
F14	SCA	8.348E+01	4.808E+03	2.178E+03	3.590E+04	7.121E+03
	ISCA	**1.522E+01**	**4.669E+01**	**3.778E+01**	**2.235E+02**	**3.511E+01**
F15	SCA	2.932E+02	1.159E+04	6.949E+03	6.501E+04	1.373E+04
	ISCA	**5.281E+00**	**5.100E+01**	**2.188E+01**	**1.232E+03**	**1.716E+02**
F16	SCA	7.023E+01	4.099E+02	4.082E+02	7.783E+02	1.864E+02
	ISCA	**2.303E+00**	**8.210E+00**	**4.924E+00**	**4.220E+01**	**8.116E+00**
F17	SCA	6.175E+01	2.376E+02	2.374E+02	5.669E+02	1.121E+02
	ISCA	**3.088E+00**	**1.994E+01**	**2.230E+01**	**4.875E+01**	**9.660E+00**
F18	SCA	3.299E+04	4.076E+06	1.885E+06	2.945E+07	5.502E+06
	ISCA	**2.128E+02**	**7.867E+03**	**5.138E+03**	**2.572E+04**	**7.498E+03**
F19	SCA	6.155E+02	1.032E+05	3.733E+04	6.583E+05	1.566E+05
	ISCA	**4.595E+00**	**3.945E+01**	**9.563E+00**	**1.371E+03**	**1.906E+02**
F20	SCA	7.843E+01	2.468E+02	2.545E+02	4.634E+02	9.094E+01
	ISCA	**1.763E+00**	**1.995E+01**	**2.163E+01**	**3.106E+01**	**7.406E+00**
F21	SCA	1.091E+02	2.188E+02	2.290E+02	3.225E+02	6.683E+01
	ISCA	**1.003E+02**	**1.817E+02**	**2.073E+02**	**2.180E+02**	**4.765E+01**
F22	SCA	2.104E+02	8.091E+02	8.066E+02	1.576E+03	3.454E+02
	ISCA	**1.007E+02**	**1.032E+02**	**1.030E+02**	**1.064E+02**	**1.301E+00**
F23	SCA	3.416E+02	3.913E+02	3.894E+02	4.547E+02	2.162E+01
	ISCA	**3.043E+02**	**3.116E+02**	**3.110E+02**	**3.224E+02**	**4.631E+00**
F24	SCA	1.561E+02	3.971E+02	4.209E+02	4.626E+02	7.380E+01
	ISCA	**1.005E+02**	**3.334E+02**	**3.428E+02**	**3.548E+02**	**4.769E+01**
F25	SCA	4.391E+02	8.395E+02	7.714E+02	1.427E+03	2.582E+02
	ISCA	**3.980E+02**	**4.178E+02**	**4.029E+02**	**4.477E+02**	**2.164E+01**
F26	SCA	4.979E+02	1.082E+03	9.640E+02	2.290E+03	4.278E+02
	ISCA	**2.078E+02**	**3.289E+02**	**3.002E+02**	**1.223E+03**	**1.852E+02**
F27	SCA	4.062E+02	4.330E+02	4.265E+02	5.177E+02	2.437E+01
	ISCA	**3.890E+02**	**3.925E+02**	**3.908E+02**	**4.408E+02**	**7.217E+00**
F28	SCA	4.291E+02	6.677E+02	6.597E+02	8.597E+02	**1.059E+02**
	ISCA	**3.002E+02**	**4.607E+02**	**3.898E+02**	**6.118E+02**	1.137E+02
F29	SCA	3.074E+02	5.013E+02	4.856E+02	8.346E+02	1.066E+02
	ISCA	**2.361E+02**	**2.539E+02**	**2.523E+02**	**2.980E+02**	**1.288E+01**
F30	SCA	1.587E+05	5.328E+06	3.935E+06	1.833E+07	3.741E+06
	ISCA	**1.257E+03**	**2.017E+05**	**6.941E+03**	**9.981E+05**	**3.771E+05**

and CEC 2017 are taken by adopting the termination criteria as function evaluations but since both the algorithms classical SCA and proposed ISCA evaluates each search agent once in each iteration, therefore the number of maximum iterations remains same in both the algorithms. Thus, when fixing the same number of iterations, the proposed algorithm outperformance classical SCA and compete other algorithms. In terms of obtaining the feasible solutions, all the compared algorithm and proposed algorithm ISCA performance equal as the considered test problems are unconstrained in nature and to deal the bound constraints during the search process, the components of a position are fixed at boundary if they exceed. Overall, it can be concluded from the experimental analysis that the proposed algorithm ISCA enhances the search efficiency of agents during the search and maintains the balance between exploration as exploitation in order to provide the better result of the problems as compared to the classical SCA and other algorithms.

4.4. Computational complexity of ISCA

Computational complexity of any meta-heuristic algorithm is very crucial task to determine the run time of an algorithm. Generally, the computational complexity, depends on the structure of the

Table 14
Comparison of results between classical SCA and proposed ISCA on 30 dimensional IEEE CEC 2017 benchmark test problems.

Test problem	Algorithm	Minimum	Median	Mean	Maximum	STD
F1	SCA	1.137E+10	4.705E+10	3.375E+10	1.033E+11	2.858E+10
	ISCA	**7.238E+05**	**3.232E+08**	**1.545E+08**	**1.839E+09**	**4.181E+08**
F2	SCA	NA	NA	NA	NA	NA
	ISCA	NA	NA	NA	NA	NA
F3	SCA	2.800E+04	1.636E+05	1.767E+05	3.230E+05	7.588E+04
	ISCA	**1.151E+03**	**5.244E+03**	**4.953E+03**	**1.464E+04**	**2.646E+03**
F4	SCA	1.360E+03	1.605E+04	1.606E+04	4.916E+04	1.051E+04
	ISCA	**7.246E+01**	**1.233E+02**	**1.221E+02**	**1.688E+02**	**2.245E+01**
F5	SCA	2.810E+02	5.021E+02	5.342E+02	6.976E+02	1.162E+02
	ISCA	**3.731E+01**	**6.881E+01**	**6.625E+01**	**1.135E+02**	**1.856E+01**
F6	SCA	5.812E+01	1.047E+02	1.115E+02	1.422E+02	2.208E+01
	ISCA	**2.457E−01**	**1.353E+00**	**1.116E+00**	**3.293E+00**	**8.558E−01**
F7	SCA	4.095E+02	1.975E+03	2.388E+03	3.047E+03	8.609E+02
	ISCA	**6.932E+01**	**1.139E+02**	**1.095E+02**	**2.473E+02**	**2.679E+01**
F8	SCA	2.560E+02	4.765E+02	5.135E+02	6.176E+02	1.054E+02
	ISCA	**3.173E+01**	**6.107E+01**	**6.231E+01**	**9.079E+01**	**1.172E+01**
F9	SCA	4.703E+03	2.324E+04	2.525E+04	3.743E+04	9.059E+03
	ISCA	**7.685E+00**	**1.441E+02**	**1.318E+02**	**5.303E+02**	**1.132E+02**
F10	SCA	6.853E+03	7.873E+03	7.975E+03	8.675E+03	**4.559E+02**
	ISCA	**1.786E+03**	**2.908E+03**	**2.911E+03**	**4.672E+03**	6.133E+02
F11	SCA	1.731E+03	1.609E+04	1.759E+04	3.072E+04	8.007E+03
	ISCA	**8.335E+01**	**1.430E+02**	**1.365E+02**	**2.201E+02**	**3.209E+01**
F12	SCA	1.679E+09	1.176E+10	1.199E+10	2.021E+10	3.857E+09
	ISCA	**6.423E+05**	**1.101E+07**	**7.632E+06**	**4.276E+07**	**1.048E+07**
F13	SCA	3.715E+08	7.983E+09	7.804E+09	2.379E+10	4.363E+09
	ISCA	**1.866E+04**	**8.881E+04**	**8.323E+04**	**2.118E+05**	**4.863E+04**
F14	SCA	1.422E+05	4.284E+06	3.506E+06	1.793E+07	3.913E+06
	ISCA	**7.743E+02**	**1.834E+04**	**1.079E+04**	**8.267E+04**	**1.981E+04**
F15	SCA	2.493E+06	1.134E+09	8.135E+08	3.865E+09	9.946E+08
	ISCA	**2.827E+03**	**1.654E+04**	**1.337E+04**	**5.795E+04**	**1.004E+04**
F16	SCA	2.172E+03	3.827E+03	3.870E+03	5.292E+03	6.333E+02
	ISCA	**1.884E+02**	**4.920E+02**	**4.899E+02**	**9.244E+02**	**1.840E+02**
F17	SCA	4.760E+02	2.156E+03	2.101E+03	4.151E+03	6.741E+02
	ISCA	**5.356E+01**	**1.547E+02**	**1.509E+02**	**3.067E+02**	**6.401E+01**
F18	SCA	2.344E+06	5.601E+07	4.647E+07	1.627E+08	4.082E+07
	ISCA	**2.910E+04**	**2.838E+05**	**1.959E+05**	**1.163E+06**	**2.593E+05**
F19	SCA	4.041E+07	1.381E+09	1.349E+09	3.870E+09	9.968E+08
	ISCA	**2.155E+03**	**5.458E+04**	**1.030E+04**	**7.050E+05**	**1.146E+05**
F20	SCA	6.707E+02	1.161E+03	1.169E+03	1.606E+03	2.253E+02
	ISCA	**6.014E+01**	**2.120E+02**	**2.212E+02**	**3.746E+02**	**7.749E+01**
F21	SCA	4.457E+02	6.897E+02	7.021E+02	8.583E+02	9.113E+01
	ISCA	**2.326E+02**	**2.603E+02**	**2.596E+02**	**2.953E+02**	**1.605E+01**
F22	SCA	1.918E+03	8.065E+03	8.251E+03	9.154E+03	**1.162E+03**
	ISCA	**1.152E+02**	**9.874E+02**	**2.140E+02**	**4.041E+03**	1.236E+03
F23	SCA	7.388E+02	9.815E+02	9.670E+02	1.178E+03	1.023E+02
	ISCA	**3.752E+02**	**4.201E+02**	**4.190E+02**	**4.590E+02**	**1.834E+01**
F24	SCA	8.724E+02	1.056E+03	1.035E+03	1.377E+03	1.111E+02
	ISCA	**4.459E+02**	**4.897E+02**	**4.905E+02**	**5.467E+02**	**2.291E+01**
F25	SCA	8.420E+02	1.112E+04	1.149E+04	2.565E+04	5.135E+03
	ISCA	**3.974E+02**	**4.277E+02**	**4.248E+02**	**4.772E+02**	**1.442E+01**
F26	SCA	6.113E+03	8.577E+03	8.590E+03	1.273E+04	1.180E+03
	ISCA	**8.122E+02**	**1.645E+03**	**1.638E+03**	**2.090E+03**	**2.147E+02**
F27	SCA	7.457E+02	1.142E+03	1.115E+03	1.569E+03	2.119E+02
	ISCA	**4.937E+02**	**5.158E+02**	**5.145E+02**	**5.437E+02**	**1.070E+01**
F28	SCA	1.173E+03	6.353E+03	6.833E+03	8.836E+03	1.982E+03
	ISCA	**4.208E+02**	**5.137E+02**	**5.135E+02**	**5.971E+02**	**4.364E+01**
F29	SCA	1.440E+03	4.166E+03	3.975E+03	7.460E+03	1.364E+03
	ISCA	**5.144E+02**	**6.582E+02**	**6.630E+02**	**8.661E+02**	**8.309E+01**
F30	SCA	2.911E+07	1.363E+09	1.372E+09	3.546E+09	7.943E+08
	ISCA	**6.641E+04**	**6.257E+05**	**5.446E+05**	**2.012E+06**	**4.176E+05**

algorithm. Therefore the complexity of proposed algorithm ISCA will also depend on population size of search agents, dimension of the considered problem, and the termination criteria — maximum number of iterations. In this section, the computational complexity of proposed ISCA is calculated in terms of $O-$notation i.e. the worst time complexity is calculated from its pseudo code presented in Algorithm 2.

$$\begin{aligned} O(\text{ISCA}) &= O(\text{positions update}) + O(\text{crossover}) \\ &\quad + O(\text{greedy selection}) \\ &\quad + O(\text{selection of best search agent}) \\ &= O(T(ND + N)) = O(\text{TND} + \text{TN}) \end{aligned}$$

Similarly,

$$O(SCA) = O(TND + TN)$$

where T represents the maximum number of iterations, N is the number of search agents in the population and D represents the dimension size of the problem. From the calculated time complexities, it can be observed that in terms of worst time complexity both the algorithm classical SCA and proposed ISCA are same.

Table 15
Statistical decisions based on Wilcoxon Signed rank test at 5% level of significance on 30 dimensional problems of IEEE CEC 2017.

Test function	p-value	Decision	Test function	p-value	Decision
F1	5.145E−10	+	F16	5.145E−10	+
F2	NA	NA	F17	5.145E−10	+
F3	5.145E−10	+	F18	5.145E−10	+
F4	5.145E−10	+	F19	5.145E−10	+
F5	5.145E−10	+	F20	5.145E−10	+
F6	5.145E−10	+	F21	5.145E−10	+
F7	5.145E−10	+	F22	5.145E−10	+
F8	5.145E−10	+	F23	5.145E−10	+
F9	5.145E−10	+	F24	5.145E−10	+
F10	5.145E−10	+	F25	5.145E−10	+
F11	5.145E−10	+	F26	5.145E−10	+
F12	5.145E−10	+	F27	5.145E−10	+
F13	5.145E−10	+	F28	5.145E−10	+
F14	5.145E−10	+	F29	5.145E−10	+
F15	5.145E−10	+	F30	5.145E−10	+

5. Applications of ISCA on engineering test problems

Validation of the proposed algorithm ISCA on an extensive set of benchmark test problems ensures the significant improvement in the search ability of solutions. The performance of ISCA on the benchmark test problems examines the reliability and efficiency of the proposed algorithm. Moreover, in this section, proposed ISCA is used to determine the solutions of engineering optimization problems. The two set of engineering test applications have been taken in this section. First contains the unconstrained problems where only bound constraint are present and the second set is of constrained application problems. To deal with the constraints of optimization problems, a simple constraint handling technique based on constraint violation is used. In this approach, the solutions are compared based on the constraint violation and Deb's feasibility rules [79] are used to compare the two solutions. Deb's feasibility rules consist of three rules:

1. A solution which is feasible is selected between feasible and infeasible solutions.
2. A solution with better fitness is selected between two feasible solutions.
3. A solution with less constraint violation is selected between two infeasible solutions.

The constraint violation $viol_{\hat{x}}$, for a general optimization problem (1)–(4), corresponding to the solution $\hat{x}$ can be calculated as follows:

$$viol_{\hat{x}} = \sum_{j=1}^{J} G_j(x) + \sum_{k=1}^{K} H_k(x)$$

$$G_j(x) = \begin{cases} g_j(x) & if\ g_j(x) > 0 \\ 0 & otherwise \end{cases}$$

$$H_k(x) = \begin{cases} |h_k(x)| & if\ \ |h_k(x)| - \epsilon > 0 \\ 0 & otherwise \end{cases}$$

where ϵ is predefined tolerance parameter which is fixed as 10^{-4} in the present work.

5.1. Gear train design problem

This problem is a discrete optimization problem because all the decision variables of this problem are restricted to be an integer. In this problem, four decision parameters x_1, x_2, x_3 and x_4 are involved which represents the number of tooths for four gears of a train. The objective of this problem is to find the optimal number of a tooth to minimize the gear ratio [80,81]. To deal with the discrete parameters, decision variables are rounded to the nearest integer in the algorithm. Mathematically, the problem can be stated as follows –

$$\textbf{\textit{Min}}\quad F_1(x) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2, \text{ where } x = (x_1, x_2, x_3, x_4)$$

$$\textbf{\textit{s.t.}}\quad 12 \le x_i \le 60, \text{ and } x_i \in Z^+\ \forall i = 1, 2, 3, 4.$$

Table 16
Performance comparison of proposed ISCA with other meta-heuristic algorithms with 30 dimensional CEC2017 problems.

Test problem	SCA	PSO	WOA	MFO	SSA	m-SCA	OBSCA	ISCA
F1	1.27E+10	5.56E+09	2.63E+06	1.31E+10	4.52E+03	1.22E+10	1.57E+10	1.55E+08
F2	NA	NA	NA	NA	NA	NA	NA	NA
F3	3.67E+04	1.33E+05	1.53E+05	9.48E+04	0.00E+00	3.76E+04	7.18E+04	4.95E+03
F4	9.90E+02	5.09E+02	1.43E+02	9.70E+02	9.00E+01	9.74E+02	1.61E+03	1.22E+02
F5	2.75E+02	3.04E+02	2.74E+02	2.20E+02	1.12E+02	2.85E+02	3.07E+02	6.63E+01
F6	5.00E+01	5.60E+01	6.91E+01	4.00E+01	3.00E+01	5.12E+01	5.57E+01	1.12E+00
F7	4.30E+02	4.30E+02	5.34E+02	4.60E+02	1.63E+02	4.33E+02	4.74E+02	1.09E+02
F8	2.50E+02	2.90E+02	2.10E+02	2.20E+02	1.19E+02	2.52E+02	2.73E+02	6.23E+01
F9	4.74E+03	5.70E+03	7.10E+03	6.57E+03	1.88E+03	4.86E+03	6.33E+03	1.32E+02
F10	7.13E+03	8.29E+03	5.16E+03	4.34E+03	3.79E+03	7.11E+03	6.01E+03	2.91E+03
F11	9.70E+02	2.54E+03	3.67E+02	5.30E+03	1.90E+02	1.04E+03	1.26E+03	1.37E+02
F12	1.18E+09	6.47E+08	3.16E+07	3.81E+08	1.52E+06	1.16E+09	1.77E+09	7.63E+06
F13	3.96E+08	1.93E+08	1.41E+05	9.52E+07	8.28E+04	4.14E+08	5.95E+08	8.32E+04
F14	1.53E+05	1.11E+06	1.07E+06	2.32E+05	3.18E+03	1.35E+05	2.12E+05	1.08E+04
F15	1.62E+07	3.91E+07	6.67E+04	5.21E+04	6.20E+04	1.34E+07	1.62E+07	1.34E+04
F16	2.04E+03	2.35E+03	1.93E+03	1.55E+03	9.20E+02	2.03E+03	2.11E+03	4.90E+02
F17	7.00E+02	9.60E+02	8.55E+02	8.80E+02	2.80E+02	6.93E+02	9.04E+02	1.51E+02
F18	3.16E+06	7.90E+06	2.83E+06	3.19E+06	1.54E+05	2.72E+06	4.74E+06	1.96E+05
F19	2.37E+07	5.17E+07	2.79E+06	2.42E+07	2.67E+05	2.47E+07	5.00E+07	1.03E+04
F20	6.10E+02	1.02E+03	7.78E+02	6.80E+02	4.10E+02	6.11E+02	7.68E+02	2.21E+02
F21	4.60E+02	5.00E+02	4.76E+02	4.10E+02	3.00E+02	4.55E+02	2.49E+02	2.60E+02
F22	5.78E+03	4.64E+03	5.01E+03	4.27E+03	2.50E+03	5.92E+03	5.31E+03	2.14E+02
F23	6.90E+02	7.80E+02	7.29E+02	5.30E+02	4.50E+02	6.97E+02	6.25E+02	4.19E+02
F24	7.60E+02	8.10E+02	7.51E+02	5.90E+02	5.10E+02	7.61E+02	7.71E+02	4.91E+02
F25	7.10E+02	7.10E+02	4.52E+02	8.30E+02	3.90E+02	6.89E+02	8.31E+02	4.25E+02
F26	4.34E+03	4.05E+03	4.76E+03	3.26E+03	1.72E+03	4.42E+03	2.31E+03	1.64E+03
F27	7.00E+02	6.60E+02	6.47E+02	5.60E+02	5.40E+02	7.02E+02	7.17E+02	5.15E+02
F28	1.00E+03	8.40E+02	5.09E+02	1.76E+03	4.10E+02	1.01E+03	1.23E+03	5.14E+02
F29	1.73E+03	2.02E+03	2.03E+03	1.25E+03	1.00E+03	1.75E+03	2.09E+03	6.63E+02
F30	6.69E+07	4.85E+07	1.01E+07	1.02E+06	9.93E+05	7.23E+07	3.82E+07	5.45E+05

Table 17
Comparison of results on gear train design problem.

Algorithm	Optimal decision variables				Objective function value ($F_{1,min}$)	Number of function evaluations
	x_1	x_2	x_3	x_4		
ISCA	43	16	19	49	**2.7009E−12**	**700**
SCA	52	25	12	40	2.3576E−09	700
GA	33	14	17	50	1.3616E−09	NA
ALM	33	15	13	41	2.4073E−08	NA
MBA	43	16	19	49	2.7009E−12	10,000
ABC	43	16	19	49	2.7800E−11	40,000
CS	43	16	19	49	2.7009E−12	5,000

Table 18
Comparison of results on parameter estimation for frequency-modulated.

Algorithm	Min	Mean	Max	STD
ISCA	2.62E−05	6.72	11.71	5.30
SCA	10.50	14.58	19.47	3.02
PSO	25.24	27.631	29.65	1.17
G-CMA-ES	3.33	38.75	55.09	16.77
CPSOH	3.45	27.08	42.52	60.61

This problem is solved by proposed method ISCA and the obtained best solution is reported in Table 17. In the table the solution obtained from classical SCA and other algorithms (Augmented Lagrange multiplier (ALM) [82], Genetic Algorithm (GA) [83], Artificial Bee Colony (ABC) algorithm [22], Cuckoo Search (CS) algorithm [84] and Mine Blast Algorithm (MBA) [85]) are also reported. From the results presented in Table 17, it can be analyzed that the ISCA provides a better solution in terms of accuracy and function evaluation cost. As CS and MBA provide the same objective function value as in ISCA but consumes more function evaluations.

5.2. Parameter estimation for frequency-modulated (FM) [86]

The objective of this problem is to estimate the decision parameters of a frequency-modulated synthesizer. This problem is highly complex and multimodal with strong epistasis. In this problem, six decision variables are involved. The mathematical form of the problem can be defined as follows:

$$Min \quad F_2(x) = \sum_{t=1}^{100} (X(t) - X_0(t))^2, \qquad x = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$$

$$s.t. \quad -6.4 \le a_1, \omega_1, a_2, \omega_2, a_3, \omega_3 \le 6.35$$

where

$$X(t) = a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta)))$$

The expression for target sound waves is given by –

$$X_0(t) = a_1 \sin\left(5t \times \frac{2\pi}{100} + 1.5 \sin\left(4.8t \times \frac{2\pi}{100} + 2 \sin\left(4.9t \times \frac{2\pi}{100}\right)\right)\right)$$

This problem is solved by proposed method ISCA and the obtained best solution is reported in Table 18. In the literature, this problem is solved by various algorithms like Particle Swarm Optimization (PSO) [19], G-CMA-ES [87,88] and CPSOH [88,89]. The obtained results from all these algorithms are presented in the same table. From the presented results, it can be analyzed that ISCA is more efficient as compared to other algorithms.

5.3. Cantilever beam deign problem

This design problem is related to the minimization of weight in cantilever beam with square cross-section [90]. The cantilever beam is supported at node 1 and the vertical force is acting at node 5. In this problem four decision parameters are involved which represents the height or width of different beams. Schematic diagram of a cantilever beam is plotted in Fig. 12 [84]. Mathematically, the problem can be stated as follows:

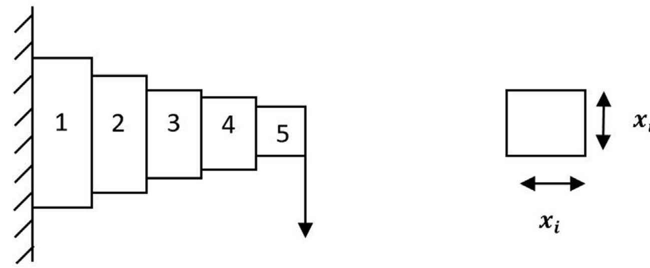

Fig. 12. Cantilever beam.

Table 19
Comparison of results on cantilever beam design problem.

Algorithm	Optimum decision variables					Objective function value ($F_{3,min}$)
	x_1	x_2	x_3	x_4	x_5	
ISCA	6.0089	5.3049	4.5023	3.5077	2.1504	**1.3399**
SCA	6.3435	4.8755	4.1788	3.4916	2.1887	1.3400
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CONLIN	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.3399
GCA (I)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA (II)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400

$$\boldsymbol{Min} \quad F_3(x) = 0.0624 \times \sum_{i=1}^{5} x_i$$

$$\boldsymbol{s.t.} \quad g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0$$

$$0.01 \le x_j \le 100, \ \text{for } j = 1, 2, 3, 4, 5.$$

This problem is solved by the proposed algorithm ISCA with the same number of function evaluation used in [84]. The best solution obtained by ISCA and classical SCA is presented in Table 19. In the table, comparison is made between ISCA, SCA, and various other techniques [84,91].

5.4. Speed reducer design problem [92]

This problem consists of 7 decision variables namely face width (b), module of teeth (m), number of teeth on pinion (z), length of shaft 1 between bearings (l_1), length of shaft 2 between bearings (l_2), diameter of shaft 1(d_1), and diameter of shaft 2(d_2). The objective of this problem is to minimize the total weight of the speed reducer. Mathematically, the problem can be stated as follows:

$$\boldsymbol{Min}\ \boldsymbol{F_4}(\boldsymbol{x}) = 0.7854 x_1 x_2^2 \left(14.9334 x_3 + 3.3333 x_3^2 - 43.0934\right)$$

Table 20
Comparison of results on speed reducer design problem.

Test problems	Variables	SCA	ES	CS	Taguchi-aided search	ISCA
	$x_1(b)$	3.51889	3.506163	3.5015	3.60	3.5004
	$x_2(m)$	0.7	0.700831	0.7	0.7	0.7
	$x_3(z)$	17	17	17	17	17
F_3	$x_4(l_1)$	7.3	7.460181	7.6050	7.30	7.30159
	$x_5(l_2)$	8.3	7.962143	7.8181	7.80	7.8
	$x_6(d_1)$	3.35899	3.3629	3.3520	3.40	3.35065
	$x_7(d_2)$	5.30519	5.308949	5.2875	5.00	5.28679
	Optimum value	3028.8657	3025.005127	3000.9810	2876.1176[a]	**2996.7017**

[a]Represents the infeasible solution.

$$+ 7.4777\left(x_6{}^3 + x_7{}^3\right) - 1.508x_1\left(x_6{}^2 + x_7{}^2\right) + 0.7854(x_4x_6{}^2 + x_5x_7{}^2)$$

$$x = (b, m, z, l_1, l_2, d_1, d_2) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

$$\begin{aligned}
\textbf{s.t.}\quad & g_1(x) = \frac{27}{x_1x_2{}^2x_3} \le 1 \\
& g_2(x) = \frac{397.5}{x_1x_2{}^2x_3{}^2} \le 1 \\
& g_3(x) = \frac{1.93x_5{}^3}{x_2x_3x_7{}^4} \le 1 \\
& g_4(x) = \frac{1.93x_4{}^3}{x_2x_3x_6{}^4} \le 1 \\
& g_5(x) = \frac{\sqrt{1.57\times10^8 + \left(\frac{745x_5}{x_2x_3}\right)^2}}{85x_7{}^3} \le 1 \\
& g_6(x) = \frac{\sqrt{1.69\times10^7 + \left(\frac{745x_4}{x_2x_3}\right)^2}}{110x_6{}^3} \le 1 \\
& g_7(x) = \frac{x_2x_3}{40} \le 1 \\
& g_8(x) = \frac{x_1}{12x_2} \le 1 \\
& g_9(x) = \frac{5x_2}{x_1} \le 1 \\
& g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} \le 1 \\
& g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} \le 1 \\
& 2.6 \le x_1 \le 3.6 \\
& 0.7 \le x_2 \le 0.8 \\
& 17 \le x_3 \le 28 \\
& 7.3 \le x_4 \le 8.3 \\
& 7.8 \le x_5 \le 8.3 \\
& 2.9 \le x_6 \le 3.9 \\
& 5 \le x_7 \le 5.5.
\end{aligned}$$

This problem is solved by the proposed algorithm, ISCA, using the same number of function evaluation as used [84] and the obtained results are reported in Table 20. In the table, the results of ES [93], CS [84], Taguchi-Aided Search [94] are also reported which are used in the literature to solve this problem. From the table, it can be observed that the proposed method ISCA outperforms other techniques to minimize the weight of speed reducer.

5.5. *Pressure vessel design problem [95]*

In this problem, the objective is to minimize the total cost consisting of forming, material and welding of a cylindrical vessel. The schematic diagram of a pressure vessel is presented in Fig. 13 [84]. The head of the vessel has a hemispherical shape and the ends of vessels are capped. In the problem, four variables — thickness of shell T_s, thickness of head T_h, inner radius R and length of cylindrical shell L are involved. The first two decision parameters are discrete and are multiple of 0.0625. Therefore this problem can be considered as a mixed-integer optimization problem. Also, the problem consists four constraints in which three are linear and one is non-linear. The mathematical form of the problem is presented is as follows:

$$\begin{aligned}
\textbf{\textit{Min}}\quad F_4(x) = {} & 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 19.84x_1^2x_3 \\
& + 3.1661x_1^2x_4
\end{aligned}$$

$$x = (x_1, x_2, x_3, x_4) = (T_s, T_h, R, L)$$

$$\begin{aligned}
\textbf{s.t.}\quad & g_1(x) = 0.0193x_3 - x_1 \le 0 \\
& g_2(x) = 0.00954x_3 - x_2 \le 0 \\
& g_3(x) = 1296000 - \frac{4}{3}\pi x_3^3 - \pi x_3^2x_4 \le 0 \\
& g_4(x) = x_4 - 240 \le 0 \\
& 1\times0.0625 \le x_1, x_2 \le 99\times0.0625 \\
& 10 \le x_3, x_4 \le 200
\end{aligned}$$

To obtain the solution of this problem proposed algorithm ISCA is employed with the same number of function evaluations as used in [84] and the obtained results are reported in Table 21. In the literature, various optimization techniques like GSA [96], PSO (He and Wang) [97], GA (Coello) [98], GA (Deb) [99], Branch and Bound (Sandgren) [100], Lagrangian Multiplier [82] are used to solve this problem. The results of these techniques are also presented in the same table. The comparative results in the table show that the proposed improved version of SCA is more efficient to obtain the minimum cost as compared to other optimization techniques.

6. Application of ISCA in multilevel thresholding

In this section, the proposed algorithm (ISCA) has been used to for multilevel thresholding in image segmentation.

6.1. *Multilevel thresholding*

The bi-level thresholding partitions the image into two classes — (*i*) the object and (*ii*) the background. But if the image is more complex and it contains a number of objects then the bi-level thresholding is not very effective. In such situations to segment an image into various classes, the method of multilevel thresholding

Table 21
Comparison of results on Pressure vessel design problem.

Algorithm	Optimum decision variables				Optimum cost ($F_{4,min}$)
	x_1	x_2	x_3	x_4	
ISCA	0.8125	0.4375	42.09842	176.6382	6059.7457
SCA	0.8125	0.4375	42.04861	177.7078	6076.3651
GSA	1.1250	0.6250	55.9887	84.4542	8538.8360
PSO (He and Wang)	0.8125	0.4375	42.0913	176.7465	6061.0780
GA (Coello)	0.8125	0.4345	40.3239	20.0000	6288.7450
GA (Deb)	0.9375	0.5000	48.3290	112.6790	6410.3810
Branch and bound (Sandgren)	1.1250	0.6250	47.7000	117.7010	8129.1040
Lagrangian multiplier	1.1250	0.6250	58.2910	43.6900	7198.043

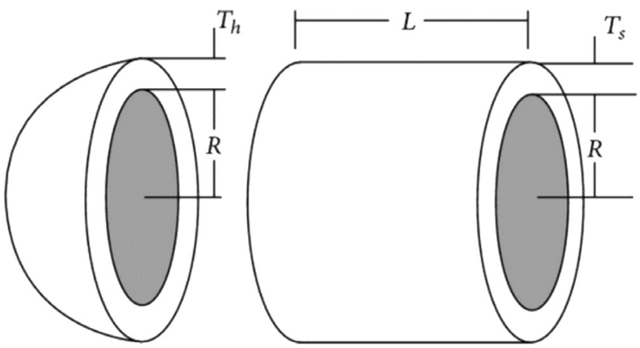

Fig. 13. Pressure vessel design.

Table 22
Parameter values for different algorithms.

Algorithm	Parameter	Value
SCA	Population size	12
	Iterations	100
	r_1	2 to 0 in decreasing nature
ISCA	Population size	12
	Iterations	100
	r_1	2 to 0 in decreasing nature
HS	Population size	12
	Iterations	100
	Band width (BW)	0.5
	PAR	0.5
	HMCR	0.7
PSO	Number of particles	12
	Iterations	100
	Inertia weight	$w_{max} - t\left(\frac{w_{max}-w_{min}}{T}\right)$ where $w_{min} = 0.2$ $w_{max} = 0.9$

is used. In multilevel thresholding, the selection of appropriate values of thresholds is very crucial to segment an image properly. The techniques for the selection of optimal thresholds optimizes the objective function. Otsu's method (based on class variance) is a widely used methodology for the optimal thresholding. In this section, the brief formulation of this method has been provided.

Let there are L number of gray levels in an image within the range $\{0, 1, 2, \ldots, L-1\}$. Let N be the number which denotes the total number of pixels in the image. Let, in the image, a particular gray level j occur n_i times in the image. The gray level histogram of an image can be normalized and regarded as a probability distribution. The occurrence probability of a particular gray level can be defined as $p_i = \frac{n_i}{N}$. The Otsu's between class variance functions methods is defined as follows:

6.1.1. Otsu method (between class variance) [101]

Otsu method is a non-parametric and unsupervised threshold selection method. In the Otsu method, the threshold optimum thresholds are selected by maximizing the class variance between segmented classes. Suppose m number of thresholds $(t_1, t_2, \ldots, t_m)$ are to be selected. These thresholds divides the image into $m+1$ classes: $c_0, c_1, c_2, \ldots, c_m$ by maximizing the class variance as:

$$F(t_1, t_2, \ldots, t_m) = v_0^2 + v_1^2 + \cdots v_m^2$$

where,

$$v_0^2 = w_0 (u_0 - u_T)^2, \qquad w_0 = \sum_{j=0}^{t_1-1} p_j, \qquad u_0 = \sum_{j=0}^{j=t_1-1} \frac{jp_j}{w_0}$$

$$v_k^2 = w_k (u_k - u_T)^2, \qquad w_k = \sum_{j=t_k}^{j=t_{k+1}-1} p_j, \qquad u_k = \sum_{j=t_k}^{j=t_{k+1}-1} \frac{jp_j}{w_k},$$

$$k = 1, 2, \ldots, m-1$$

$$v_m^2 = w_m (u_m - u_T)^2, \qquad w_k = \sum_{j=t_m}^{j=L-1} p_j, \qquad u_k = \sum_{j=t_m}^{j=L-1} \frac{jp_j}{w_m}$$

$v_0, v_1, \ldots, v_m$ are the variances, $w_0, w_1, \ldots, w_m$ are the class probabilities, $u_0, u_1, \ldots, u_m$ are the mean levels of the segmented classes $c_0, c_1, c_2, \ldots, c_m$. u_T is the mean intensity for the image and can be obtained as: $u_T = \sum_{j=0}^{m} w_j u_j$ and $\sum_{j=0}^{m} w_j$.

In Otsu's method, the objective function is of maximization type, which can be converted into minimization type also by the following transformation:

$$F'(t_1, t_2, \ldots, t_m) = \frac{1}{1 + F(t_1, t_2, \ldots, t_m)}$$

6.2. Experimental environment and results analysis for multilevel thresholding

In this section, the experimental environment is introduced for the proposed algorithm, ISCA. In this direction, the benchmark images are introduced first, and then the parameter setting for the algorithm is presented. The obtained experimental results with some set of quality metrics: Mean, Median, Maximum, Average, Standard deviation of objective function values and Peak Signal-to-Noise Ratio (PSNR).

6.3. Benchmark images

In the paper, the proposed algorithm (ISCA) is tested on eight common grayscale images — Cameraman, Baboon, Peppers, Male, Couple, Lena, Bridge and Boat. To observe the efficiency of the proposed algorithm, a single colored test image (Baboon) is also taken in the experimentation. These test images have been collected from the database of USC-SIPI. The test images are shown in Fig. 14. The parameter selection for various algorithms is provided in Table 22.

Fig. 14. Data set of test images: (A) Cameraman (B) Baboon (C) Peppers (D) Male (E) Couple (F) Lena (G) Bridge (H) Boat (I) Baboon (colored)..

6.4. Experimental results and analysis of results

The obtained results on the test image with the same parameter setting as explained in Table 22, are presented in Tables 23 and 24, and the corresponding threshold values are presented in Table 25. In these tables, the thresholds are fixed as 4 and 5. In Table 23, the obtained mean objective fitness by Otsu method is presented. In Table 24, the obtained best function values using the Otsu method is reported respectively. In these tables, the results are also compared from PSO and HS. All the results are reported by conducting 30 trials of each algorithm. From the table, it can be observed that the proposed algorithm provides better results as compared to classical SCA and other algorithms. The obtained segmented images for Lena test image and their thresholds have been shown in Fig. 15 with 4, 5 and 6 thresholds. The obtained segmented output images of all benchmark test images with 4, 5 and 6 thresholds are presented in Figs. 16 and 17. In order to evaluate the segmented image quality, Peak Signal-to-Noise Ratio (PSNR) values are presented in Table 26. PSNR is a well-known image processing metric. The PSNR depends directly on the intensity values in the image. The PSNR measure indicates the accuracy of the final segmented image. The PSNR value can be calculated as follows:

$$PSNR = 10\log_{10}\frac{255^2}{MSE}$$

$$\text{where} \quad MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I\left(i,j\right) - O\left(i,j\right)\right)^2$$

where I and O represent the original and final segmented images respectively.

The results corresponding to the colored image are presented in Table 27. The PSNR measure presented in the Tables 26 and 27

Table 23
The mean objective function values obtained by SCA and ISCA based on Otsu's method.

Test image	No. of thresholds	Otsu's mean objective function values			
		PSO	HS	SCA	ISCA
Cameraman	4	3737.83	3744.83	3771.81	**3782.31**
	5	3773.40	3773.23	3798.04	**3813.58**
Baboon	4	1619.01	1621.11	1677.84	**1692.98**
	5	1666.54	1656.85	1700.69	**1717.44**
Peppers	4	2696.65	2703.21	2755.37	**2766.17**
	5	2748.51	2742.55	2790.58	**2810.48**
Male	4	3146.43	3170.87	3190.67	**3208.69**
	5	3199.12	3201.51	3235.75	**3254.52**
Couple	4	1391.11	1410.54	1446.40	**1448.96**
	5	1444.24	1452.93	1477.10	**1497.12**
Lena	4	2097.13	2097.62	2173.13	**2191.57**
	5	2152.36	2145.51	2208.49	**2216.63**
Bridge	4	2748.64	2757.91	2818.72	**2822.53**
	5	2810.84	2812.05	2850.07	**2873.84**
Boat	4	2001.20	2008.69	2052.93	**2059.82**
	5	2039.83	2047.84	2085.44	**2092.57**

Table 24
The best objective function values obtained by SCA and ISCA based on Otsu's method.

Test image	No. of thresholds	Otsu's objective function values			
		PSO	HS	SCA	ISCA
Cameraman	4	3776.62	3765.97	3772.63	**3782.40**
	5	3800.15	3802.23	3804.78	**3813.74**
Baboon	4	1688.40	1679.78	1689.90	**1693.20**
	5	1703.30	1712.05	1709.29	**1718.97**
Peppers	4	2747.38	2758.53	2755.37	**2766.46**
	5	2793.18	2793.01	2790.58	**2810.83**
Male	4	3207.56	3207.58	3190.67	**3208.81**
	5	3244.58	3248.91	3235.75	**3254.80**
Couple	4	1437.60	1445.16	1446.40	**1449.81**
	5	1483.28	1484.06	1477.10	**1497.60**
Lena	4	2180.90	2181.50	2173.13	**2191.87**
	5	2189.79	2201.52	2208.49	**2217.68**
Bridge	4	2811.86	2803.63	2818.72	**2822.71**
	5	2851.20	2859.03	2850.07	**2874.24**
Boat	4	2054.17	2050.26	2052.93	**2059.87**
	5	2076.25	2080.15	2085.44	**2092.77**

Table 25
The threshold values obtained by SCA and ISCA based Otsu's method.

Test image	No. of thresholds	Threshold values	
		SCA	ISCA
Cameraman	4	34, 104, 140, 172	42, 95, 140, 170
	5	29, 84, 119, 139, 168	36, 82, 122, 149, 173
Baboon	4	73, 107, 132, 167	72, 106, 137, 168
	5	59, 83, 116, 147, 168	68, 99, 126, 151, 176
Peppers	4	51, 105, 141, 178	47, 86, 126, 169
	5	57, 84, 125, 160, 179	43, 79, 113, 147, 178
Male	4	36, 82, 138, 171	35, 82, 124, 164
	5	26, 71, 94, 139, 186	28, 65, 100, 134, 172
Couple	4	57, 100, 132, 170	63, 103, 137, 179
	5	46, 78, 119, 144, 178	59, 98, 129, 158, 204
Lena	4	81, 102, 139, 179	75, 114, 145, 180
	5	56, 86, 123, 152, 179	72, 108, 136, 160, 188
Bridge	4	61, 99, 144, 199	64, 103, 146, 194
	5	51, 91, 106, 151, 209	55, 89, 121, 158, 200
Boat	4	64, 120, 154, 181	65, 114, 147, 179
	5	65, 97, 132, 156, 194	54, 96, 131, 155, 186

Table 26
PSNR values obtained by SCA and ISCA based on Otsu's and Kapur's method.

Test image	No. of thresholds	Otsu's PSNR values	
		SCA	ISCA
Cameraman	4	21.2713	**21.5079**
	5	22.2374	**23.2643**
Baboon	4	19.6511	**20.2622**
	5	21.0420	**22.0478**
Peppers	4	20.1015	**20.6615**
	5	21.3918	**22.3423**
Male	4	20.6055	**20.9807**
	5	21.8956	**22.6074**
Couple	4	19.4479	**20.3839**
	5	20.7038	**21.4646**
Lena	4	18.4593	**18.6264**
	5	19.3817	**19.5601**
Bridge	4	18.6292	**18.9109**
	5	20.0340	**20.6011**
Boat	4	19.9443	**20.2858**
	5	21.1678	**22.0481**
Airplane	4	19.9443	**20.2858**
	5	21.1678	**22.0481**

for both the methods (classical SCA and ISCA) favor the significance of proposal of modification in search strategy of SCA. The results presented the experimental tables shows that the proposed algorithm provides better results as compared to classical SCA and other algorithms.

Some concluding remarks that can be made based on all the presented results on benchmark test problems and application problems are as follows:

1. The better balance of social and cognitive component has been established in the proposed algorithm ISCA as compared to classical SCA by using the personal best memory of search agents.
2. The numerical results presented for classical benchmarks, standard benchmark CEC 2014 and latest collection of benchmarks CEC 2017 demonstrate the better search ability in the proposed algorithm as compared to the classical SCA.
3. The average distance plot in each iteration between the search agents demonstrates the balance between the exploration and exploitation during the search process in the proposed algorithm.
4. The engineering test problems which consists various difficulty levels and search space also favors the better efficacy of proposed ISCA in terms of obtaining the solutions.
5. The proposed ISCA is also used for thresholding of images and the results show that the proposed algorithm is better optimizer as compared to classical SCA other compared algorithms.

7. Conclusions and future scope

The paper introduces an improved version of Sine Cosine Algorithm through the crossover with personal best state of solutions. In the proposed algorithm, the search equations are modified by integrating the personal best state in place of the global best state to decide the region of search space around the personal best state of a solution and to prevent from the problem of stagnation at local optima. In the search equation, global best or social direction is also integrated with random steps to provide the information of best position preserved in the memory to the candidate solutions. A greedy selection mechanism and crossover with personal best

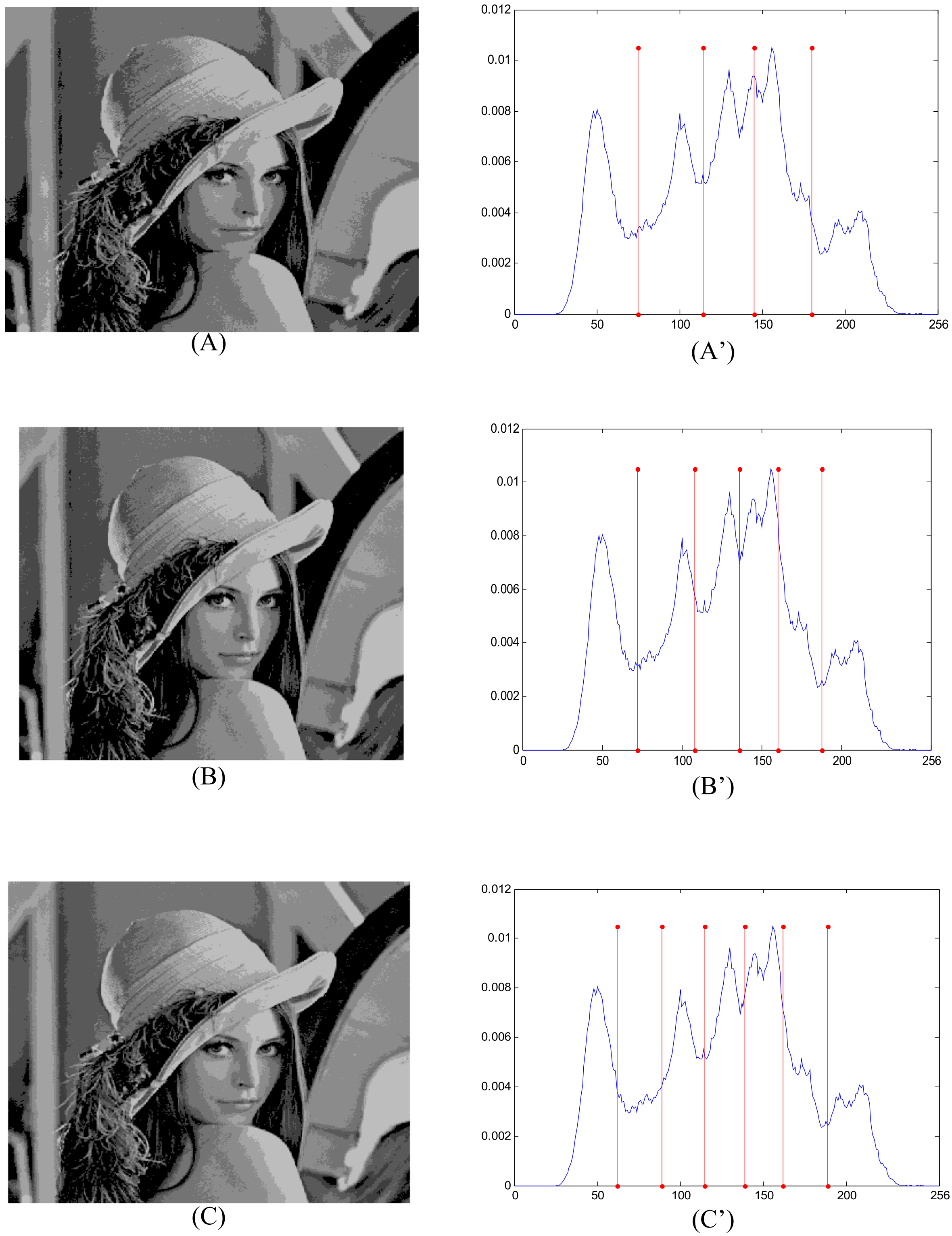

Fig. 15. Segmentation results for the test Image: Lena by ISCA based for Otsu's method. (A)–(C) represents segmented images into five, six and seven classes and (A′)–(C′) represents the thresholds for segmented image.

state reduce the overflow of diversity. To examine the significance of integrated strategies in the proposed algorithm (ISCA), it has been passed through three levels of benchmarks — classical, standard IEEE CEC 2014 and recent benchmarks IEEE CEC 2017. Furthermore, to observe the effectiveness of the proposed algorithm on real-life problems, it has been used to solve five well-known engineering optimization problems with various difficulty levels. At the end of the paper, the proposed algorithm is also used for multilevel image segmentation. The statistical, convergence behavior analysis and Performance Index analysis have proven the efficacy, accuracy and robustness of the proposed algorithm (ISCA) as compared to the classical SCA and other state-of-the-art algorithms. The conducted analysis in the paper, recommends that

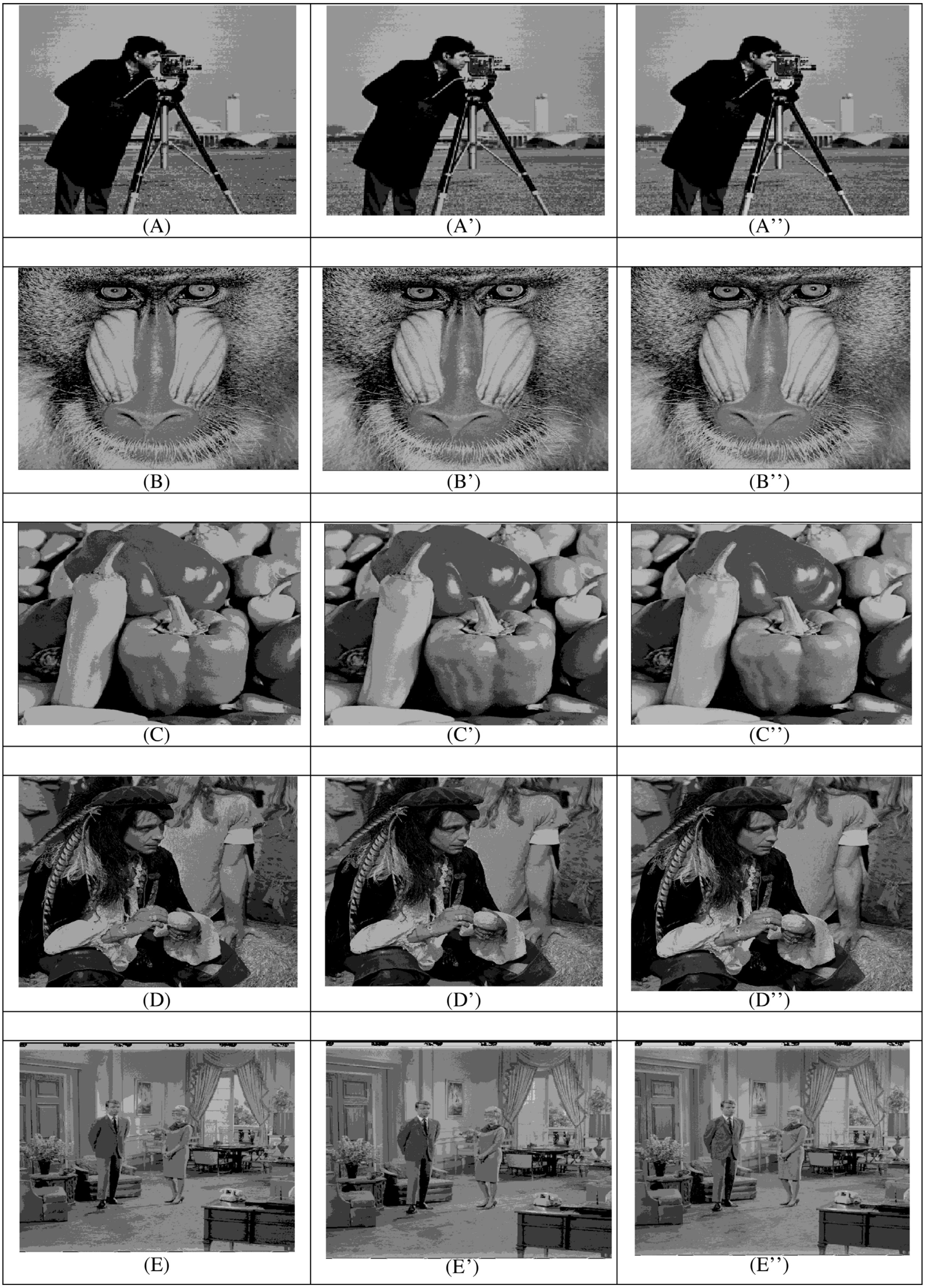

Fig. 16. Segmented images obtained from ISCA using Otsu’s method (A–E): corresponding to 4-level, (A′–E′): corresponding to 5-level and (A″–E″): corresponding to 6-level thresholding.

Fig. 17. Segmented Images obtained from ISCA using Otsu's method (F–H): corresponding to 4-level, (F′–H′): corresponding to 5-level and (F″–H″): corresponding to 6-level thresholding.

Table 27
Obtained results for color test image (Baboon).

Test image	No. of thresholds	Algorithm	Threshold values R	G	B	PSNR
Airplane	4	SCA	43, 76, 143, 209	33, 86, 139, 182	58, 86, 118, 186	18.2554
		ISCA	79, 114, 160, 208	69, 103, 141, 176	62, 101, 142, 190	**18.5576**
	5	SCA	51, 101, 154, 182, 199	56, 81, 132, 155, 198	30, 60, 104, 178, 213	19.5562
		ISCA	43, 76, 86, 141, 195	54, 92, 116, 154, 160	45, 88, 138, 176, 214	**19.7876**

the proposed algorithm can be used over classical SCA as it has outperformed classical SCA.

This paper opens several research directions for future studies. According to the promising results of the proposed algorithm on test problems, in the future, we attempt to use it on complex real-life application problems. Also, in the future, we attempt to develop the multiobjective ISCA.

Acknowledgment

The first author gratefully acknowledges to the Ministry of Human Resource and Development (MHRD), Government of India for their financial support. Grant No. MHR-02-41-113-429.

References

[1] Seyed Hamid Reza Pasandideh, Seyed Taghi Akhavan Niaki, Abolfazl Gharaei, Optimization of a multiproduct economic production quantity problem with stochastic constraints using sequential quadratic programming, Knowl.-Based Syst. 84 (2015) 98–107.

[2] Kenji Kira, Larry A. Rendell, Rendell The Feature Selection Problem: Traditional Methods and a New Algorithm, in: Aaai, Vol. 2, 1992, pp. 129–134.

[3] Suvrit Sra, Sebastian Nowozin, Stephen J. Wright (Eds.), Wright eds Optimization for Machine Learning, Mit Press, 2012.

[4] Hoseini Shekarabi, Seyed Ashkan, Abolfazl Gharaei, Mostafa Karimi, Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: generalised outer approximation, Int. J. Syst. Sci.: Operat. Logist. (2018) 1–21.

[5] Abolfazl Gharaei, Seyed Hamid Reza Pasandideh, Alireza Arshadi Khamseh, Inventory model in a four-echelon integrated supply chain: Modeling and optimization, J. Modell. Manag. 12 (4) (2017) 739–762.

[6] Abolfazl Gharaei, Seyed HamidReza Pasandideh, Modeling and optimization of four-level integrated supply chain with the aim of determining the optimum stockpile and period length: sequential quadratic programming, J. Ind. Prod. Eng. 34 (7) (2017) 529–541.

[7] Abolfazl Gharaei, Seyed Hamid Reza Pasandideh, Seyed Taghi Akhavan Niaki, An optimal integrated lot sizing policy of inventory in a bi-objective multi-level supply chain with stochastic constraints and imperfect products, J. Ind. Prod. Eng. 35 (1) (2018) 6–20.

[8] Abolfazl Gharaei, Seyed Hamid Reza Pasandideh, Four-echelon integrated supply chain model with stochastic constraints under shortage condition, Ind. Eng. Manag. Syst. 16 (3) (2017) 316–329.

[9] Abolfazl Gharaei, Seyed Hamid Reza Pasandideh, Modelling and optimization the four-level integrated supply chain: sequential quadratic programming, Int. J. Comput. Sci. Inf. Secur. 14 (7) (2016) 650–669.

[10] Abolfazl Gharaei, Mostafa Karimi, Seyed Hoseini Shekarabi Ashkan, An integrated multi-product multi-buyer supply chain under the penalty, green, and VMI-CS policies: OA/ER/AP algorithm, Appl. Math. Model. (2018) http://dx.doi.org/10.1016/j.apm.2018.11.035.

[11] Chaoqun Duan, Chao Deng, Abolfazl Gharaei, Jun Wu, Bingran Wang, Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions, Int. J. Prod. Res. (2018) 1–19.

[12] Gharaei Abolfazl, Bahman Naderi, Mohammad Mohammadi, Optimization of rewards in single machine scheduling in the rewards-driven systems, Manag. Sci. Lett. 5 (6) (2015) 629–638.

[13] Paolo Toth, Daniele Vigo (Eds.), Vehicle Routing: Problems, Methods, and Applications, Society for Industrial and Applied Mathematics, 2014.

[14] Wayne L. Winston, Jeffrey B. Goldberg, Operations Research: Aapplications and Algorithms. Vol. 3, Thomson Brooks/Cole, Belmont, 2004.

[15] Ahron Ben-Tal, Arkadi Nemirovski, Nemirovski Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, Vol. 2, Siam, 2001.

[16] J. Hoffmann, B. Nebel, The FF planning system: fast plan generation through heuristic search, J. Artificial Intelligence Res. 14 (2001) 253–302.

[17] Peter JM. Van Laarhoven, Emile HL. Aarts, Simulated annealing, in: Simulated Annealing: Theory and Applications, Springer, Dordrecht, 1987, pp. 7–15.

[18] John Henry Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992.

[19] Russell Eberhart, James Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95, Proceedings of the Sixth International Symposium on, IEEE, 1995, pp. 39–43.

[20] Rainer Storn, Kenneth Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[21] Marco Dorigo, Mauro Birattari, Birattari Ant colony optimization, in: Encyclopedia of Machine Learning, Springer, Boston, MA, 2011, pp. 36–39.

[22] Dervis Karaboga, Bahriye Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459–471.

[23] Seyedali Mirjalili, SeyedMohammad Mirjalili, Andrew Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[24] Seyedali Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.

[25] Seyedali Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.

[26] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[27] Bo Xing, Wen-Jing Gao, Innovative Computational Intelligence: a Rough Guide to 134 Clever Algorithms, Springer International Publishing, Cham, Heidelberg, New York, Dordrecht, London, 2014.

[28] Yongquan Zhou, Qifang Luo, Huan Chen, Anping He, Jinzhao Wu, A discrete invasive weed optimization algorithm for solving traveling salesman problem, Neurocomputing 151 (2015) 1227–1236.

[29] Xin-She Yang, Flower pollination algorithm for global optimization, in: International Conference on Unconventional Computing and Natural Ccomputation, Springer, Berlin, Heidelberg, 2012, pp. 240–249.

[30] Seyedali Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 9612 (2016) 0–133.

[31] Ahmed Ibrahem Hafez, Hossam M. Zawbaa, E. Emary, Ella Hassanien Aboul, Emary Aboul Ella Hassanien Sine cosine optimization algorithm for feature selection, in: INnovations in Intelligent SysTems and Applications (INISTA), 2016 International Symposium on, IEEE, 2016, pp. 1–5.

[32] Ahmed T. Sahlol, Ahmed A. Ewees, Ahmed Monem Hemdan, Aboul Ella Hassanien, Training feedforward neural networks using Sine-Cosine algorithm to improve the prediction of liver enzymes on fish farmed on nano-selenite, in: Computer Engineering Conference (ICENCO), 2016 12th International, IEEE, 2016, pp. 35–40.

[33] Mohamed Abd Elfattah, Sherihan Abuelenin, Aboul Ella Hassanien, Jeng-Shyang Pan, Handwritten arabic manuscript image binarization using sine cosine optimization algorithm, in: International Conference on Genetic and Evolutionary Computing, Springer, Cham, 2016, pp. 273–280.

[34] Sadeque Hamdan, Shaikha Binkhatim, Anwar Jarndal, Imad Alsyouf, On the performance of artificial neural network with sine-cosine algorithm in forecasting electricity load demand, in: Electrical and Computing Technologies and Applications (ICECTA), 2017 International Conference on, IEEE, 2017, pp. 1–5.

[35] V. Kumar, D. Kumar, Data clustering using sine cosine algorithm: Data clustering using SCA, in: Handbook of Research on Machine Learning Innovations and Trends, IGI Global, 2017, pp. 715–726.

[36] Sai Li, Huajing Fang, Xiaoyong Liu, Parameter optimization of support vector regression based on sine cosine algorithm, Expert Syst. Appl. 91 (2018) 63–77.

[37] Betül Sultan Yıldız, Ali Rıza Yıldız, Comparison of grey wolf, whale, water cycle, ant lion and sine-cosine algorithms for the optimization of a vehicle engine connecting rod, Mater. Test. 60 (3) (2018) 311–315.

[38] Belkacem Mahdad, K. Srairi, A new interactive sine cosine algorithm for loading margin stability improvement under contingency, Electr. Eng. 100 (2) (2018) 913–933.

[39] Abdel-Fattah Attia, Ragab A. El Sehiemy, Hany M. Hasanien, Optimal power flow solution in power systems using a novel Sine-Cosine algorithm, Int. J. Electr. Power Energy Syst. 99 (2018) 331–343.

[40] M. Laxmidevi Ramanaiah, M. Damodar Reddy, Sine Cosine Algorithm for Loss Reduction in Distribution System with Unified Power Quality Conditioner, i-Manag. J. Power Syst. Eng. 5 (3) (2017) 10.

[41] MA Mushahhid Majeed, Patri Srihari Rao, Optimization of CMOS analog circuits using sine cosine algorithm, in: Computing, Communication and Networking Technologies (ICCCNT), 2017 8th International Conference on, IEEE, 2017, pp. 1–6.

[42] Rizk M. Rizk-Allah, An improved sine–cosine algorithm based on orthogonal parallel information for global optimization, Soft Comput. (2018) 1–27.

[43] Ning Li, Gang Li, Zhong Liang Deng, An improved sine cosine algorithm based on levy flight, in: Ninth International Conference on Digital Image Processing (ICDIP 2017), 10420, 104204R, International Society for Optics and Photonics, 2017.

[44] Divya Bairathi, Dinesh Gopalani, Opposition-Based Sine Cosine Algorithm (OSCA) for Training Feed-Forward Neural Networks, in: Signal-Image Technology & Internet-Based Systems (SITIS), 2017 13th International Conference on, IEEE, 2017, pp. 438–444.

[45] Mohamed Abd Elaziz, Diego Oliva, Shengwu Xiong, An improved opposition-based sine cosine algorithm for global optimization, Expert Syst. Appl. 90 (2017) 484–500.

[46] Sandeep Kaur, Shushil Prashar, A novel sine cosine algorithm for the solution of unit commitment problem, International Journal of Science, Engineering and Technology Research 5 (2016) 12.

[47] Mostafa Meshkat, Mohsen Parhizgar, A novel weighted update position mechanism to improve the performance of sine cosine algorithm, in: Fuzzy and Intelligent Systems (CFIS), 2017 5th Iranian Joint Congress on, IEEE, 2017, pp. 166–171.

[48] Shubham Gupta, Kusum Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2018) 210–230.

[49] Deepak Ranjan Nayak, Ratnakar Dash, Banshidhar Majhi, Shuihua Wang, Combining extreme learning machine with modified sine cosine algorithm for detection of pathological brain, Comput. Electr. Eng. 68 (2018) 366–380.

[50] Jinzhong Zhang, Yongquan Zhou, Qifang Luo, An improved sine cosine water wave optimization algorithm for global optimization, J. Intell. Fuzzy Syst. 34 (4) (2018) 2129–2141.

[51] Sujin Bureerat, Nantiwat Pholdee, Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection, in: International Conference on Computational Science and Its Applications, Springer, Cham, 2017, pp. 71–86.

[52] Hathiram Nenavath, Ravi Kumar Jatoth, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking, Appl. Soft Comput. 62 (2018) 1019–1043.

[53] Rizk M. Rizk-Allah, Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems, J. Comput. Des. Eng. 5 (2) (2018) 249–273.

[54] Kamal Z. Zamli, Fakhrud Din, Bestoun S. Ahmed, Miroslav Bures, A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem, PLoS One 13 (5) (2018) e0195675.

[55] Jianzhou Wang, Wendong Yang, Pei Du, Tong Niu, A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm, Energy Convers. Manage. 163 (2018) 134–150.

[56] Mohamed A. Tawhid, Vimal Savsani, Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems, Neural Comput. Appl. (2017) 1–15.

[57] Pablo Moscato, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report 826 1989 1989.

[58] David B. Fogel, Michalewicz Zbigniew, Handbook of Evolutionary Computation, Taylor & Francis, London, 1997.

[59] Agoston E. Eiben, James E. Smith, Introduction to Evolutionary Computing , Vol. 53, springer, Berlin, 2003.

[60] Holger H. Hoos, Thomas Stützle, Stochastic Local Search: Foundations and Applications, Elsevier, 2004.

[61] J. J.Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 2013.

[62] 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. In Technical Report. Nanyang Technological University Singapore, 2016.

[63] Helon Vicente Hultmann Ayala, Fernando Marins dos Santos, Viviana Cocco Mariani, Image thresholding segmentation based on a novel beta differential evolution approach, Expert Syst. Appl. 42 (4) (2015) 2136–2142.

[64] Madhubanti Maitra, Amitava Chatterjee, A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding, Expert Syst. Appl. 34 (2) (2008) 1341–1350.

[65] Musrrat Ali, Chang Wook Ahn, Millie Pant, Multi-level image thresholding by synergetic differential evolution, Appl. Soft Comput. 17 (2014) 1–11.

[66] Soham Sarkar, Swagatam Das, Sheli Sinha Chaudhuri, A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution, Pattern Recognit. Lett. 54 (2015) 27–35.

[67] Kamal Hammouche, Moussa Diaf, Patrick Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, Eng. Appl. Artif. Intell. 23 (5) (2010) 676–688.

[68] Mohamed Abd El Aziz, Ahmed A. Ewees, Aboul Ella Hassanien, Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, Expert Syst. Appl. 83 (2017) 242–256.

[69] AshishKumar Bhandari, VineetKumar Singh, Anil Kumar, GirishKumar Singh, Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy, Expert Syst. Appl. 41 (7) (2014) 3538–3560.

[70] Abdul Kayom Md Khairuzzaman, Saurabh Chaudhury, Multilevel thresholding using grey wolf optimizer for image segmentation, Expert Syst. Appl. 86 (2017) 64–76.

[71] Eva Tuba, Adis Alihodzic, Milan Tuba, Multilevel image thresholding using elephant herding optimization algorithm, in: Engineering of Modern Electric Systems (EMES), 2017 14th International Conference on, IEEE, 2017, pp. 240–243.

[72] Kezong Tang, Xuan Xiao, Jun Wu, Jingyu Yang, Limin Luo, An improved multilevel thresholding approach based modified bacterial foraging optimization, Appl. Intell. 46 (1) (2017) 214–226.

[73] Lifang He, Songwei Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, Neurocomputing 240 (2017) 152–174.

[74] Xin Yao, Yong Liu, Guangming Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.

[75] Guang-Yu Zhu, Wei-Bo Zhang, Optimal foraging algorithm for global optimization, Appl. Soft Comput. 51 (2017) 294–313.

[76] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

[77] Kusum Deep, Manoj Thakur, A new crossover operator for real coded genetic algorithms, Appl. Math. Comput. 188 (1) (2007) 895–911.

[78] Seyedali Mirjalili, Amir H. Gandomi, SeyedehZahra Mirjalili, Shahrzad Saremi, Hossam Faris, SeyedMohammad Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[79] Kalyanmoy Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 311–338.

[80] Amir H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, ISA Trans. 53 (4) (2014) 1168–1183.

[81] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, J. Mech. Des. 112 (2) (1990) 223–229.

[82] B.K. Kannan, Steven N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (2) (1994) 405–411.

[83] Kalyanmoy Deb, Mayank Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Comput. Sci. Inform. 26 (1996) 30–45.

[84] Amir Hossein Gandomi, Xin-She Yang, AmirHossein Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.

[85] Ali Sadollah, Ardeshir Bahreininejad, Hadi Eskandar, Mohd Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.

[86] Swagatam Das, Ponnuthurai N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Nanyang Technological University, Kolkata 2010.

[87] Jing J. Liang, A. Kai Qin, Ponnuthurai N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[88] Peter JM Van Laarhoven, Emile HL Aarts, Simulated annealing, in: Simulated Annealing: Theory and Applications, Springer, Dordrecht, 1987, pp. 7–15.

[89] Anne Auger, Nikolaus Hansen, A restart CMA evolution strategy with increasing population size, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol. 2, IEEE, 2005, pp. 1769–1776.

[90] Claude Fleury, Vincent Braibant, Structural optimization: a new dual method using mixed variables, Internat. J. Numer. Methods Engrg. 23 (3) (1986) 409–428.

[91] HEMIANT CHICKERMANE, H.C. Gea, Structural optimization using a new local approximation method, Internat. J. Numer. Methods Engrg. 39 (5) (1996) 829–846.

[92] Amir Hossein Gandomi, Xin-She Yang, Benchmark problems in structural optimization, in: Computational Optimization, Methods and Algorithms, Springer, Berlin, Heidelberg, 2011, pp. 259–281.

[93] Efrén Mezura-Montes, C.A. Coello Coello, Ricardo Landa-Becerra, Landa-Becerra Engineering optimization using simple evolutionary algorithm, in: Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on, IEEE, 2003, pp. 149–156.

[94] K'uang J. Ku, S.S. Rao, Li Chen, Taguchi-aided search method for design optimization of engineering systems, Eng. Optim. 30 (1) (1998) 1–23.

[95] Frans Van den Bergh, Andries Petrus Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 225–239.

[96] Esmat Rashedi, Hossein Nezamabadi-Pour, Saeid Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.

[97] Qie He, Ling Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.

[98] Carlos A. Coello Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.

[99] Kalyanmoy Deb, GeneAS: A robust optimal design technique for mechanical component design, in: Evolutionary Algorithms in Engineering Applications, Springer, Berlin, Heidelberg, 1997, pp. 497–514.

[100] E. Sandgren, Nonlinear integer and discrete programming in mechanical design, in: Proceeding of the ASME Design Technology Conference, 1988, pp. 95–105.

[101] Nobuyuki Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.