Engineering Applications of Artificial Intelligence 82 (2019) 148–174

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

STOA: A bio-inspired based optimization algorithm for industrial engineering problems☆

Gaurav Dhiman, Amandeep Kaur *

Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala 147004, Punjab, India

ARTICLE INFO

Keywords:
Optimization
Bio-inspired metaheuristic techniques
Constrained problems
Benchmark test problems

ABSTRACT

This paper presents a bio-inspired algorithm called Sooty Tern Optimization Algorithm (STOA) for solving constrained industrial problems. The main inspiration of this algorithm is the migration and attacking behaviors of sea bird *sooty tern* in nature. These two steps are implemented and mathematically modeled to emphasize exploitation and exploration in a given search space. The proposed algorithm is compared with nine well-known bio-inspired algorithms over 44 benchmark test functions. The analysis of convergence behaviors and computational complexity of the proposed algorithm have been evaluated. Furthermore, to demonstrate its applicability it is then employed to solve six constrained industrial applications. The outcomes of experiment reveal that the proposed algorithm is able to solve challenging constrained problems and is very competitive compared with other optimization algorithms.

1. Introduction

Optimization is the process of defining the decision variables of a function to minimize or maximize its values (Dhiman and Kumar, 2018a). Most of the real world problems (Chandrawat et al., 2017; Singh and Dhiman, 2017; Kaur and Dhiman, 2019; Singh and Dhiman, 2018b; Singh et al., 2018b,a; Kaur et al., 2018; Dhiman et al., 2018, 2019; Dhiman and Kumar, 2018) have high computational cost, non-linear constraints, non-convex, and huge amount of solution spaces and are complicated. Therefore, solving such problems with large number of variables and constraints is very tedious and complex (Spears et al., 2012). Secondly, there are many local optimum solutions that do not guarantee the best overall solution using classical numerical methods.

To overcome these problems, metaheuristic optimization algorithms (Dhiman and Kaur, 2019; Dhiman and Kumar, 2019b; Dhiman and Kaur, 2017) are introduced which are capable of solving such complex problems throughout course of iterations. Recently, researchers are gaining interest in developing metaheuristic algorithms (Dhiman and Kumar, 2018c; Singh and Dhiman, 2018a; Dhiman and Kaur, 2018; Dhiman and Kumar, 2018b, 2019a) that are computationally inexpensive, flexible, and simple by nature. Metaheuristics are broadly classified into two categories namely single solution and population based algorithms. Algorithms based on a single solution are those in which a solution is generated randomly and improved until the best result is obtained. Population-based algorithms are those in which a set of solutions is randomly generated in a given search space and the solution values are updated during the iterations until the best solution is found.

However, algorithms based on a single solution can be trapped in local optima, thus not allowing the discovery of the global optimum. This is because, for a given problem, it reforms only one randomly generated solution. On the other hand, population based algorithms are able to find the global optimum. Due to this, researchers have been attracted towards population based algorithms nowadays.

The categorization of population based algorithms is based on the theory of evolutionary algorithms, logical behavior of physics algorithms, swarm intelligence of particles, and biological behavior of bio-inspired algorithms. Evolutionary algorithms are inspired by the evolutionary processes such as reproduction, mutation, recombination, and selection. These algorithms are based on the survival fitness of candidate in a population (i.e., a set of solutions) for a given environment. The physics law based algorithms are inspired by physical processes according to some physics rules such as gravitational force, electromagnetic force, inertia force, heating and cooling of materials. Swarm intelligence based algorithms are inspired by the collective intelligence of swarms. The collective intelligence is found among colonies of flocks, ants, and so on.

Generally, swarm intelligence based algorithms are very easier to implement than evolutionary algorithms due to the smaller number of parameters required. The best known algorithms of the swarm intelligence technique are Particle Swarm Optimization (PSO) (Kennedy

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.03.021.

* Corresponding author.
E-mail addresses: gaurav.dhiman@thapar.edu, gdhiman0001@gmail.com (G. Dhiman), kaur.amandeep@thapar.edu (A. Kaur).

https://doi.org/10.1016/j.engappai.2019.03.021
Received 18 June 2018; Received in revised form 22 January 2019; Accepted 19 March 2019
Available online xxxx

and Eberhart, 1995) and Ant Colony Optimization (ACO) (Dorigo et al., 2006).

Every optimization algorithm needs to address the exploration and exploitation of a search space (Alba and Dorronsoro, 2005) and sustains a good balance between exploration and exploitation. The exploration phase in an algorithm investigates the different promising regions in a search space, while exploitation is able to search for optimal solutions around the promising regions (Lozano and Garcia-Martinez, 2010). Therefore, fine tuning of these two phases is needed to achieve the near optimal solutions. Despite the significant number of recently developed optimization algorithms, the question that should be raised is: why do we need to develop more optimization techniques. The answer lies in No Free Lunch (NFL) theorem (Wolpert and Macready, 1997). According to this theorem, the performance of one optimization algorithm for a specific set of problems does not guarantee to solve other optimization problems because of their different nature. The NFL theorem permits researchers to propose some novel optimization algorithms for solving the problems in different application areas. This motivated us to develop a gradient free technique and effective optimization algorithm to optimize constrained engineering problems.

This paper presents a bio-inspired metaheuristic algorithm named as Sooty Tern Optimization Algorithm (STOA) for optimizing constrained problems. As its name implies, STOA simulate the migration and attacking behaviors of sooty terns in real life. The performance of STOA algorithm is evaluated on forty four well-known benchmark test functions. Further, STOA algorithm is employed to optimize the designs of pressure vessel, speed reducer, welded beam, tension/compression spring, 25-bar truss, and rolling element bearing. The results show that STOA performance is very competitive when compared to existing algorithms.

The rest of this paper is structured as follows: the related works done in the field of single-objective optimization is presented in Section 2. Section 3 describes the proposed STOA algorithm in detail. Section 4 presents the experimental results and their discussion. In Section 5, the performance of STOA is tested on six constrained industrial optimization problems and it is compared to other well-known algorithms. Finally, the conclusion and some future research directions are discussed in Section 6.

2. Related works

Population based metaheuristic algorithms are broadly categorized into three facets namely Physics-based, Evolutionary-based, and Swarm-based methods.

The first approach is Physics-based algorithms in which search agents can communicate around the search space according to laws of physics such as electromagnetic force, inertia force, gravitational force. The general mechanism of these algorithms is different from other approaches due to the strategy of search agents as per physics rules. The popular Physics-based optimization algorithms are Simulated Annealing (SA) (Kirkpatrick et al., 1983) and Gravitational Search Algorithm (GSA) (Rashedi et al., 2009). Simulated Annealing is inspired from annealing in metallurgy that involves heating and controlled cooling attributes of a material. These attributes depend on its thermodynamic free energy. Gravitational Search Algorithm is based on the law of gravity and mass interactions. The population solutions interact with each other through gravity force and their performance is measured by its mass. Some of the other popular algorithms are: Big-Bang Big-Crunch (BBBC) (Erol and Eksin, 2006), Charged System Search (CSS) (Kaveh and Talatahari, 2010a), Black Hole (BH) (Hatamlou, 2013) algorithm, Central Force Optimization (CFO) (Formato, 2009), Small-World Optimization Algorithm (SWOA) (Du et al., 2006), and Ray Optimization (RO) algorithm (Kaveh and Khayatazad, 2012).

The second subclass of metaheuristic algorithms is Evolutionary-based algorithms. These algorithms are stimulated by theory of natural selection and biological evolution. These algorithms often perform well to find near optimal solutions because they do not make any belief about the basic fitness landscape. The most popular evolutionary algorithm is Genetic Algorithm (GA). GA was first proposed by Holland (1992). Differential Evolution (DE) (Storn and Price, 1997) is another Evolutionary-based algorithm which optimize a problem by maintaining a candidate solutions and creates new candidate solutions by combining the existing ones. Apart from these, some of the other popular evolutionary-based algorithms are: Genetic Programming (GP) (Koza, 1992), Evolution Strategy (ES) (Beyer and Schwefel, 2002), and Biogeography-Based Optimizer (BBO) (Simon, 2008).

The third main branch in metaheuristics is Swarm-based algorithms which are based on the collective behavior of social creatures. These techniques are generally inspired from natural colonies, flock, and herds. The most popular algorithm is Particle Swarm Optimization (PSO) which was first proposed by Kennedy and Eberhart (1995). In PSO, particles move around the search space using a combination of the best solution which they have individually found and the best solution that any particle in their neighborhood has found (Slowik and Kwasnicka, 2017). The whole process is repeated until the termination criterion is satisfied. Ant Colony Optimization (ACO) is another popular swarm intelligence algorithm which was proposed by Dorigo et al. (2006). The main inspiration behind this algorithm is the social behavior of ants in ant colony. The social intelligence of ants is to find the shortest path between the source food and nest. Bat-inspired Algorithm (BA) (Yang, 2010) is another intelligence based algorithm that is inspired by the echolocation behavior of bats. Another well-known swarm based technique is Artificial Bee Colony (ABC) algorithm (Karaboga and Basturk, 2007) which is inspired by the combined behavior of bees to find the food sources. Spotted Hyena Optimizer (SHO) (Dhiman and Kumar, 2017) is a recently develop bio-inspired optimization algorithm which mimics the searching, hunting, and attacking behaviors of spotted hyenas in nature. The main concepts of this technique area related to the social relationship, and collective behavior, of spotted hyenas when hunting. Cuckoo Search (CS) (Yang and Deb, 2009) is another bio-inspired optimization approach which is inspired by the obligate brood parasitism of cuckoo species. These species lay its eggs in the nest of other species. Each egg, and cuckoo egg, represent a solution and a new solution, respectively.

3. Sooty tern optimization algorithm (STOA)

In this section, the inspiration and mathematical modeling of proposed algorithm is discuss in detail.

3.1. Biological paradigm

Sooty terns, scientifically called *Onychoprion fuscatus*, are sea birds which can be found all over the planet. There are wide range of sooty terns species with different sizes and masses. Sooty terns are omnivorous and eat reptiles, insects, earthworms, fish, amphibians, and so on. To attract earthworms hidden under the ground, sooty terns produce rain-like sound with their feet and to attract fish they use bread crumbs. Generally, sooty terns live in colonies. To find and attack the prey they use their intelligence. The eminent thing about the sooty terns is their migrating and attacking behaviors. Migration is defined as the seasonal movement of sooty terns from one place to another to find the richest and most abundant food sources that will provide adequate energy. This behavior is described as follows:

- During migration, sooty terns travel in a group. The initial positions of sooty terns are different to avoid the collisions between each other.
- In a group, sooty terns can travel towards the direction of best survival fittest sooty tern, i.e., a sooty tern whose fitness value is low as compared to others.
- Based on the fittest sooty tern, other sooty terns can update their initial positions.

Sooty terns used a flapping mode of flight (i.e., spiral shape movement) while attacking in an air (Watson, 1908). These behaviors can be formulated in such a way that it can be associated with the objective function to be optimized.

3.2. Mathematical model

The mathematical models of migration and attacking behaviors are discussed below.

3.2.1. Migration behavior (exploration)

During migration, a sooty tern should satisfy the three conditions which are described as follows:

- **Collision avoidance:** S_A is used for the computation of new search agent position to avoid the collision avoidance between its neighboring search agents (i.e., sooty terns).

$$\vec{C}_{st} = S_A \times \vec{P}_{st}(z) \tag{1}$$

where $\vec{C}_{st}$ is the position of search agent which does not collide with other search agent, $\vec{P}_{st}$ represents the current position of search agent, z indicates the current iteration, and S_A indicates the movement of search agent in a given search space.

$$S_A = C_f - (z \times (C_f / Max_{iterations}))$$
where, (2)
$$z = 0, 1, 2, \ldots, Max_{iterations}$$

where C_f is a controlling variable to adjust the S_A which is linearly decreased from C_f to 0. The value of variable C_f is set to 2 in this paper.

- **Converge towards the direction of best neighbor's:** After collision avoidance, the search agents converge towards the direction of best neighbor.

$$\vec{M}_{st} = C_B \times (\vec{P}_{bst}(z) - \vec{P}_{st}(z)) \tag{3}$$

where $\vec{M}_{st}$ indicates the different locations of search agent $\vec{P}_{st}$ towards the best fittest search agent $\vec{P}_{bst}$. C_B is a random variable which is responsible for better exploration. C_B is computed as follows:

$$C_B = 0.5 \times R_{and} \tag{4}$$

where R_{and} is a random number which lies between the range of $[0, 1]$.

- **Updation corresponding to best search agent:** Finally, the search agent or sooty tern can update its position with respect to best search agent.

$$\vec{D}_{st} = \vec{C}_{st} + \vec{M}_{st} \tag{5}$$

where $\vec{D}_{st}$ defines the gap between the search agent and best fittest search agent.

3.2.2. Attacking behavior (exploitation)

During migration, sooty terns can change their velocity and angle of attack. They use their wings to increase their altitude. While attacking on the prey, they generate the spiral behavior in the air which is described as follows (Tamura and Yasuda, 2017):

$$x' = R_{adius} \times \sin(i) \tag{6}$$

$$y' = R_{adius} \times \cos(i) \tag{7}$$

$$z' = R_{adius} \times i \tag{8}$$

$$r = u \times e^{kv} \tag{9}$$

where R_{adius} represents the radius of each turn of the spiral, i represents the variable lies between the range of $[0 \leq k \leq 2\pi]$. u and v are constants to define the spiral shape, and e is the base of the natural logarithm. The value of constants u and v is 1 is assumed in this paper. Hence, the updated position of search agent is computed using Eqs. (6)–(9).

$$\vec{P}_{st}(z) = (\vec{D}_{st} \times (x' + y' + z')) \times \vec{P}_{bst}(z) \tag{10}$$

where $\vec{P}_{st}(z)$ updates the positions of other search agents and saves the best optimal solution.

Algorithm : Sooty Tern Optimization Algorithm

Input: Population $\vec{P}_{st}$
Output: Best search agent, $\vec{P}_{bst}$
1: **procedure** STOA
2: Initialize the parameters S_A and C_B
3: Calculate the fitness of each search agent
4: $\vec{P}_{bst} \leftarrow$ best search agent
5: **while** $(z < Max_{iterations})$ **do**
6: **for** each search agent **do**
7: Update the positions of search agents by using Eq. (10)
8: **end for**
9: Update the parameters S_A and C_B
10: Calculate the fitness value of each search agent
11: Update $\vec{P}_{bst}$ if there is a better solution than previous optimal solution
12: $z \leftarrow z + 1$
13: **end while**
14: return $\vec{P}_{bst}$
15: **end procedure**

3.3. Computational complexity

The complexity metric is one of the most prominent measure to judge performance of an algorithm. The population initialization process of STOA and other algorithms takes $\mathcal{O}(n_o \times n_p)$ time, where n_o indicates the number of objectives and n_p indicates the number of population size. The complexity of computing the fitness of search agents for all algorithms needs $\mathcal{O}(Max_{iterations} \times o_f)$, where o_f indicates the objective function for a given problem. Whereas, the space complexity of all algorithms is the maximum amount of utilized space at any one time. This space complexity is considered in the initial steps of algorithms. In this paper, the space complexity for all algorithms are considered as $\mathcal{O}(n_o \times n_p)$.

The computational complexity of proposed STOA algorithm is $\mathcal{O}(N \times Max_{iterations} \times n_o \times n_p \times o_f)$. The computational complexity of SHO algorithm is $\mathcal{O}(N \times G \times Max_{iterations} \times n_o \times n_p \times o_f)$ since it requires $\mathcal{O}(G)$ time to define the group of spotted hyenas. The computational complexities of MFO, MVO, GWO, PSO, and SCA algorithms are $\mathcal{O}(N \times Max_{iterations} \times n_o \times n_p \times o_f)$. The GSA algorithm requires $\mathcal{O}(f^2)$ time for calculating the force component and thus, overall requires $\mathcal{O}(N \times f^2 \times Max_{iterations} \times n_o \times n_p \times o_f)$. Finally, the computational complexities of DE and GA algorithms are $\mathcal{O}(N \times Max_{iterations} \times n_o \times n_p \times o_f \times (c_s + m_t))$, where c_s and m_t indicate the crossover and mutation operators, respectively.

4. Experimental results and discussion

In this section, the proposed algorithm is tested on 44 well-known benchmark test functions and the results are compared with other competitor methods. There are five genres of these test functions such as unimodal, multimodal, fixed-dimension multimodal, CEC 2005 (Dhiman and Kumar, 2017), and CEC 2015 special session test functions (Chen et al., 2014). To demonstrate the efficiency of STOA algorithm, six constrained industrial optimization applications are also employed. To validate the performance of the proposed algorithm, the nine well-known metaheuristics are chosen and benchmarked for comparison. These are Spotted Hyena Optimizer (SHO) (Dhiman and

Table 1
Parameter setting values for algorithms involved.

Algorithms	Parameters	Values
Spotted Hyena Optimizer (SHO)	Control Parameter ($\vec{h}$)	[5, 0]
	$\vec{M}$ Constant	[0.5, 1]
Grey Wolf Optimizer (GWO)	Control Parameter ($\vec{a}$)	[2, 0]
Particle Swarm Optimization (PSO)	Inertia Coefficient	0.75
	Cognitive and Social Coefficient	1.8, 2
Moth–Flame Optimization (MFO)	Convergence Constant	[−1, -2]
	Logarithmic Spiral	0.75
Multi-Verse Optimizer (MVO)	Wormhole Existence Probability	[0.2, 1]
	Traveling Distance Rate	[0.6, 1]
Sine Cosine Algorithm (SCA)	Number of Elites	2
Gravitational Search Algorithm (GSA)	Gravitational Constant	100
	Alpha Coefficient	20
Genetic Algorithm (GA)	Crossover	0.9
	Mutation	0.05
Differential Evolution (DE)	Crossover	0.9
	Scale factor (F)	0.5

Kumar, 2017), Genetic Algorithm (GA) (Bonabeau et al., 1999), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Sine Cosine Algorithm (SCA) (Mirjalili, 2016), Moth–Flame Optimization (MFO) (Mirjalili, 2015), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), Multi-Verse Optimizer (MVO) (Mirjalili et al., 2016), Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), and Differential Evolution (DE) (Storn and Price, 1997).

4.1. Experimental setup

The parameter settings of STOA and other competitor algorithms are given in Table 1. The experimentation and algorithms are implemented in MATLAB R2017a (version 9.2) and simulations are tested on Core i7 processor with 3.2 GHz and 8 GB main memory.

4.2. Exploitation analysis

The unimodal test functions are suitable to evaluate the exploitation capability of the algorithm. Table 2 reveals the optimal values obtained by the proposed and competitor algorithms for unimodal benchmark test functions. It can be seen that STOA and SHO algorithms provide better results on $F_1(Sphere)$ test function. The GWO and GSA are the second and third best algorithms in terms of average and standard deviation. For $F_2(Schwefel's\,Problem\,2.22)$ test function, STOA and SHO algorithms are significantly better than other competitor algorithms, followed by GWO and GA as second and third best algorithms respectively. SHO algorithm obtains the best results on $F_3(Schwefel's\,Problem\,1.2)$ test function whereas STOA and GWO algorithms become second and third best optimizer, respectively. For $F_4(Schwefel's\,Problem\,2.21)$ test function, GWO obtains better results followed by SHO and STOA algorithms when compared with other algorithms. The results obtained by STOA algorithm are better than the other approaches on $F_5(Generalized\ Rosenbrock's)$ test function. For F_5 test function, SHO and GWO are the second and third best optimization algorithm. GSA algorithm may be obtains competitive results than the other metaheuristics on $F_6(Step)$ test function. Whereas, PSO and MFO algorithms may be the second and third most best optimization algorithms on this benchmark. For $F_7(Quartic)$ test function, the results obtained by STOA algorithm are superior than others. Whereas, SHO and GA algorithms outperform over other algorithms as a second and third best algorithm, respectively. Therefore, it has been concluded that the proposed STOA algorithm is able to find the best results corresponding to most of the test functions. In particular, STOA reveals the robustness for functions F_1, F_2, F_5, and F_7 which show the capability for exploitation around the optimum point.

4.3. Exploration analysis

These functions have a capability to avoid local optima problem. Multimodal and fixed-dimension multimodal test functions are suitable for better exploration of a proposed algorithm. Tables 3 and 4 depict the results of different methods on multimodal and fixed-dimension multimodal benchmark test functions, respectively. For F_8 (*Generalized Schwefel's Problem* 2.26) function, STOA algorithm secured first position comparatively existing competitive approaches. MFO and MVO are the second and third best optimization algorithms, respectively. For F_9 (*Generalized Rastrigin's Function*) function, SHO becomes the best optimizer than others. GWO and GA are the second and third best optimization algorithms on F_9 test function. For F_{10} (*Ackley's Function*) function, the average value of STOA algorithm is best as compared to other approaches whereas the standard deviation of GWO algorithm is better than the other metaheuristics. For F_{11} (*Generalized Griewank Function*) function, the performance of both STOA and SHO algorithms is identical whereas the performance of GA and DE algorithms are the second and third best than competitive optimizers. The outcomes obtained by PSO algorithm are superior than other metaheuristics on F_{12} (*Generalized Penalized Function*) function. For F_{13} (*Generalized Penalized Function*) function, GSA is the best optimization algorithm while PSO and GA obtain comparatively better results than the other algorithms. For F_{14} (*Shekel's Foxholes Function*) function, MVO provides better results. SCA and MFO are second and third best optimization algorithms, respectively. For F_{15} (*Kowalik's Function*), F_{16} (*Six-Hump Camel-Back Function*), F_{17} (*Branin Function*) and F_{18} (*Goldstein–Price Function*) functions, the results produced from STOA algorithm are superior than others. For $F_{19}(Hartman's)$ function, the standard deviation value obtained by MFO algorithm is better, while the average value obtained by PSO algorithm is better than competitive approaches. For $F_{20}(Hartman's)$ function, STOA algorithm performs superior whereas PSO and GWO are the second and third best optimizers. The standard deviation value of SHO algorithm is better on F_{21} (*Shekel's Foxholes Function*) test function while the average value of GWO algorithm is best on this test instance. On F_{22} (*Shekel's Foxholes Function*) and F_{23} (*Shekel's Foxholes Function*) benchmark test functions, the results obtained by STOA algorithm are superior than the competitor approaches. It can be seen that STOA algorithm has good exploration capability and is competitive on seven test problems (i.e., $F15, F16, F17, F18, F20, F22$, and $F23$) out of ten test functions.

4.4. Escape local minima with CEC 2005 test functions

Table 5 represents the results obtained by all algorithms in terms of average and standard deviation. For $CF1$ test function, GSA algorithm

Table 2
Results of STOA on unimodal benchmark test functions.

F	STOA		SHO		GWO		PSO		MFO		MVO		SCA		GSA		GA		DE	
	Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*
F_1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.69E−47	4.50E−48	4.98E−09	1.41E−10	3.15E−04	5.00E−05	2.81E−01	1.10E−02	3.55E−02	1.07E−02	1.16E−16	6.10E−17	1.95E−12	2.04E−13	3.10E−10	3.21E−11
F_2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.20E−24	1.00E−24	7.29E−04	1.57E−05	3.71E+01	2.16E+01	3.96E−01	1.41E−01	3.23E−05	8.28E−06	1.70E−01	5.29E−02	6.53E−18	4.00E−19	4.99E−13	4.81E−14
F_3	3.60E−20	1.31E−23	**0.00E+00**	**0.00E+00**	1.00E−14	4.18E−15	1.40E+01	7.13E+00	4.42E+03	3.71E+03	4.31E+01	8.97E+00	4.91E+03	3.89E+03	4.16E+02	1.56E+02	7.70E−10	5.46E−12	5.58E−02	1.50E−03
F_4	9.39E−04	2.76E−05	7.78E−12	5.06E−12	**2.02E−14**	**2.95E−15**	6.00E−01	1.72E−01	6.70E+01	1.06E+01	8.80E−01	2.50E−01	1.87E+01	8.21E+00	1.12E+00	9.89E−01	9.17E+01	5.67E+01	5.21E+01	1.01E−01
F_5	**7.03E+00**	**7.28E−01**	8.59E+00	5.53E−01	2.79E+01	1.84E+00	4.93E+01	3.89E+01	3.50E+03	2.83E+03	1.18E+02	1.00E+02	7.37E+02	1.90E+02	3.85E+01	3.47E+01	5.57E+02	4.16E+01	2.10E+02	1.73E+02
F_6	5.33E−01	2.67E−04	2.46E−01	1.78E−01	6.58E−01	3.38E−01	9.23E−09	8.48E−10	1.66E−04	2.49E−05	3.15E−01	9.98E−02	4.88E+00	9.75E−01	**1.08E−16**	**4.00E−17**	3.15E−01	9.98E−02	9.77E−02	5.13E−02
F_7	**6.69E−06**	**9.82E−06**	3.29E−05	2.43E−05	7.80E−04	3.85E−04	6.92E−02	2.87E−02	3.22E−01	2.93E−01	2.02E−02	7.43E−03	3.88E−02	1.15E−02	7.68E−01	5.70E−01	6.79E−04	5.21E−04	4.00E−03	2.27E−03

Table 3
Results of STOA on multimodal benchmark test functions.

F	STOA		SHO		GWO		PSO		MFO		MVO		SCA		GSA		GA		DE	
	Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*
F_8	**−8.91E+03**	3.49E+02	−1.16E+03	**2.72E+02**	−6.14E+03	9.32E+02	−6.01E+03	1.30E+03	−8.04E+03	8.80E+02	−6.92E+03	9.19E+02	−3.81E+03	2.83E+02	−2.75E+03	5.72E+02	−5.11E+03	4.37E+02	−3.94E+03	5.81E+02
F_9	2.01E+02	3.97E+01	**0.00E+00**	**0.00E+00**	4.34E−01	2.68E−01	4.72E+01	1.03E+01	1.63E+02	3.74E+01	1.01E+02	1.89E+01	2.23E+01	1.28E+01	3.35E+01	1.19E+01	1.23E+01	1.01E+01	8.50E+01	1.00E+01
F_{10}	**3.43E−15**	5.08E−12	2.48E+00	1.41E+00	1.63E−14	**3.14E−15**	3.86E−02	2.91E−04	1.60E+01	6.18E+00	1.15E+00	7.87E−01	1.55E+01	8.11E+00	8.25E−09	1.90E−09	5.31E−11	9.10E−12	7.40E−06	5.44E−07
F_{11}	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.29E−03	1.21E−03	5.50E−03	4.32E−04	5.03E−02	1.29E−02	5.74E−01	1.12E−01	3.01E−01	2.89E−01	8.19E+00	3.70E+00	3.31E−06	1.43E−06	7.09E−04	5.58E−04
F_{12}	4.91E−01	3.60E−01	3.68E−02	1.15E−02	3.93E−02	2.42E−02	**1.05E−10**	**2.86E−11**	1.26E+00	1.83E+00	1.27E+00	1.02E+00	5.21E+01	2.19E+01	2.65E−01	1.13E−01	9.16E−08	2.80E−08	1.06E−06	1.00E−06
F_{13}	2.55E−01	6.43E−02	9.29E−01	9.52E−02	4.75E−01	2.38E−01	4.03E−03	3.01E−03	7.24E−01	1.59E−01	6.60E−02	4.33E−02	2.81E+02	5.62E+01	**5.73E−4**	**1.30E−4**	6.39E−02	4.49E−02	9.78E−02	3.42E−02

Table 4
Results of STOA on fixed-dimension multimodal benchmark test functions.

F	STOA		SHO		GWO		PSO		MFO		MVO		SCA		GSA		GA		DE	
	Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*
F_{14}	4.90E+00	2.27E+00	9.68E+00	3.29E+00	3.71E+00	2.15E+00	2.77E+00	2.20E+00	2.21E+00	1.80E+00	**9.98E−01**	**4.13E−02**	1.26E+00	6.86E−01	3.61E+00	2.96E+00	4.39E+00	4.41E−02	3.97E+00	1.42E+00
F_{15}	**4.27E−04**	**5.83E−05**	9.01E−04	1.06E−04	3.66E−03	3.48E−04	9.09E−04	2.38E−04	1.58E−03	2.48E−04	7.15E−03	1.40E−03	1.01E−03	3.75E−04	6.84E−03	2.77E−03	7.36E−03	2.39E−04	1.01E−04	1.00E−04
F_{16}	**−1.07E+01**	**5.06E−12**	−1.06E+01	2.86E−11	−1.03E+00	7.02E−09	−1.03E+00	0.00E+00	−1.03E+00	0.00E+00	−1.03E+00	4.74E−08	−1.03E+00	3.23E−05	−1.03E+00	0.00E+00	−1.04E+00	4.19E−07	−1.03E+00	3.60E−04
F_{17}	**3.97E−01**	2.18E−03	3.98E−01	2.46E−01	3.98E−01	2.42E−02	3.98E−01	4.11E−02	3.98E−01	1.13E−03	3.98E−01	2.15E−02	3.99E−01	7.61E−04	3.98E−01	1.74E−04	3.98E−01	**1.00E−04**	3.98E−01	8.40E−02
F_{18}	**3.00E+00**	3.27E−05	**3.00E+00**	8.15E−05	**3.00E+00**	7.16E−06	**3.00E+00**	6.59E−05	**3.00E+00**	**1.42E−07**	5.70E+00	8.40E−04	**3.00E+00**	2.25E−05	3.01E+00	3.24E−02	3.01E+00	6.33E−07	**3.00E+00**	9.68E−04
F_{19}	−3.86E+00	6.57E−09	−3.75E+00	4.39E−01	−3.86E+00	1.57E−03	**−3.90E+00**	3.37E−15	−3.86E+00	**3.16E−15**	−3.86E+00	3.53E−07	−3.86E+00	2.55E−03	−3.22E+00	4.15E−01	−3.30E+00	4.37E−10	−3.40E+00	6.56E−06
F_{20}	**−3.32E+00**	**4.61E−02**	−1.44E+00	5.47E−01	−3.27E+00	7.27E−02	**−3.32E+00**	2.66E−01	−3.23E+00	6.65E−02	−3.23E+00	5.37E−02	−2.84E+00	3.71E−01	−1.47E+00	5.32E−01	−2.39E+00	4.37E−01	−1.79E+00	7.07E−02
F_{21}	**−1.01E+01**	4.98E+00	**−1.01E+01**	**3.80E−01**	**−1.01E+01**	1.54E+00	**−1.01E+01**	2.77E+00	**−1.01E+01**	3.52E+00	**−1.01E+01**	2.91E+00	**−1.01E+01**	1.80E+00	**−1.01E+01**	1.30E+00	**−1.01E+01**	2.34E+00	**−1.01E+01**	1.91E+00
F_{22}	**−1.04E+01**	**1.63E−04**	**−1.04E+01**	2.04E−04	−1.03E+01	2.73E−04	**−1.04E+01**	3.08E+00	**−1.04E+01**	3.20E+00	−1.01E+01	3.02E+00	**−1.04E+01**	1.99E+00	**−1.04E+01**	2.64E+00	**−1.04E+01**	1.37E−02	**−1.04E+01**	5.60E−03
F_{23}	**−1.05E+01**	**1.41E−01**	**−1.05E+01**	2.64E−01	−1.01E+01	8.17E+00	**−1.05E+01**	2.52E+00	**−1.05E+01**	3.68E+00	−1.03E+01	3.13E+00	**−1.05E+01**	1.96E+00	**−1.05E+01**	2.75E+00	**−1.05E+01**	2.37E+00	**−1.05E+01**	1.60E+00

Table 5
Results of STOA on CEC 2005 special session benchmark test functions.

F	STOA		SHO		GWO		PSO		MFO		MVO		SCA		GSA		GA		DE	
	Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*	*Ave*	*Std*
CF1	3.83E+01	**2.17E+00**	2.30E+02	1.37E+02	8.39E+01	6.02E+01	6.00E+01	5.05E+01	1.18E+02	7.40E+01	1.40E+02	1.00E+02	1.20E+02	3.11E+01	**1.81E+01**	1.05E+01	5.97E+02	1.34E+02	4.41E+02	3.57E+02
CF2	**7.26E+01**	**1.33E+00**	4.08E+02	9.36E+01	1.48E+02	3.78E+01	2.44E+02	1.73E+02	9.20E+01	1.84E+01	2.50E+02	1.44E+02	1.14E+02	1.84E+00	2.03E+02	1.99E+01	4.09E+02	2.10E+01	6.01E+02	1.14E+02
CF3	**3.11E+02**	**2.49E+01**	3.39E+02	3.14E+01	3.53E+02	5.88E+01	3.39E+02	8.36E+01	4.19E+02	1.15E+02	4.05E+02	1.67E+02	3.89E+02	5.41E+01	3.67E+02	8.38E+01	9.30E+02	8.31E+01	3.70E+02	2.11E+02
CF4	4.39E+02	2.55E+01	7.26E+02	1.21E+02	4.23E+02	1.14E+02	4.49E+02	1.42E+02	**3.31E+02**	**2.09E+01**	3.77E+02	1.28E+02	4.31E+02	2.94E+01	5.32E+02	1.01E+02	4.97E+02	3.24E+01	5.00E+02	2.12E+02
CF5	**7.60E+01**	**1.20E+01**	1.06E+02	1.38E+01	1.36E+02	2.75E+01	2.40E+02	7.49E+01	1.13E+02	9.27E+01	2.45E+02	9.96E+01	1.56E+02	8.30E+01	1.44E+02	1.09E+02	1.90E+02	5.03E+01	4.06E+02	3.45E+01
CF6	**4.61E+02**	**3.86E+00**	5.97E+02	4.98E+00	8.26E+02	1.74E+02	8.22E+02	1.80E+02	8.92E+02	2.41E+01	8.33E+02	1.68E+02	6.06E+02	1.66E+02	8.13E+02	1.13E+02	6.65E+02	3.37E+02	9.55E+02	6.04E+02

outperforms the other competitor metaheuristics. For $CF2$ and $CF3$ test function, the performance of STOA algorithm is better than the aforementioned algorithms. For $CF4$ test function, the results obtained by MFO algorithm are better than the existing metaheuristics. For $CF5$ and $CF6$ test functions, STOA algorithm outperforms the comparative optimization algorithms. SHO and MFO algorithms are the second and third-most best optimizers on $CF5$ and $CF6$ test functions. It has been concluded that the proposed STOA algorithm provides prominent results for the majority of test functions (i.e., $CF2, CF3, CF5$, and $CF6$).

4.5. Global minima finding with CEC 2015 test functions

For proper exploration and exploitation, the proposed algorithm is tested on fifteen CEC special session benchmark test functions. Table 6 shows the best obtained optimal solutions by all algorithms in terms of average and standard deviation. For $CEC - 1$ test function, the average value of STOA algorithm is better than other approaches. On this test instance, PSO algorithm provides superior value of standard deviation. For $CEC-2$ test function, the performance of GSA algorithm is better as compared to other metaheuristics. The standard deviation value of GSA algorithm is best than other approaches on $CEC - 3$ test instance, while the average value obtained by STOA algorithm is better on this test instance. For $CEC - 4, CEC - 5$, and $CEC - 6$ test functions, PSO generates optimal solutions and very competitive with other optimizers. In terms of average, the optimal values obtained by the proposed STOA algorithm is superior than all of the other optimization algorithms on $CEC-7, CEC-8, CEC-9, CEC-10, CEC-11, CEC - 12, CEC - 13, CEC - 14$, and $CEC - 15$ benchmark test instances. Despite, the standard deviation values of PSO algorithm on $CEC-8, CEC-9, CEC-10$, and $CEC-15$ test functions are better than others. The standard deviation obtained by GSA algorithm on $CEC-14$ test instance is superior as compared to other metaheuristic techniques. Fig. 1 shows the box plot for the proposed STOA and other competitor algorithms on CEC 2015 benchmark test functions. Basically, box plot shows the graphical examination and variations between approaches. The spacings between the different parts of the box indicate the degree of dispersion and skewness in the data. The results reveal that for most of the test functions STOA algorithm is the best optimizer.

4.6. Analysis of STOA algorithm

Another feature that helps a better understanding of explorative and exploitative mechanisms is the convergence analysis of the metaheuristic algorithm. Sooty terns tend to explore the diverse promising regions of the search space. The search agents can change rapidly during the initial generation of optimization process. Fig. 2 shows the convergence curves of STOA and other competitor methods. While optimizing the test functions, STOA shows three different convergence behaviors, i.e., initial, final, and explicit. Due of its adaptive mechanism, STOA converges more rapidly towards the promising regions in the initial step of iterations. This behavior is shown in $F1, F3, F7, F9$ and $F11$ test functions. During the initial steps of iterations on $F5$ test function, GSA encourages good convergence and achieves better performance. Whereas, MFO converges towards the optimum for $F21$ and $F23$ test functions. Furthermore, during the final iterations it is observed that STOA converges towards the optimum on $F21$ and $F23$ test functions. GA and MFO algorithms also converge towards optimum on $F1$, $F3$ and $F11$ test functions. Whereas, SCA and MVO converge towards optimum on $F5$ test function. In addition, the PSO algorithm on $F7$ and $F9$ test functions converge towards the optimum. The last step is the explicit convergence. This behavior for STOA algorithm is shown on $F21$ and $F23$ test functions. On the other hand, for MFO, GA, and DE algorithms the behavior is shown on $F9$ test function.

The following, three metrics (shown in Fig. 3) are also employed in a given search space to demonstrate the convergence analysis of STOA algorithm.

- **Search history** shows the location history of sooty terns during optimization.
- **Trajectory** curves show that in the initial steps of optimization the sooty terns reveal huge and unexpected changes. According to van den Bergh and Engelbrecht (2006), this behavior can guarantee that a Swarm-based method eventually converges to a point in search space.
- **Average fitness** indicates the average objective value of all sooty terns in each iteration. The curves show descending behavior on all of the test functions. This proves that STOA algorithm improves the accuracy of the approximated optimum during simulation runs.

Therefore, the success rate of STOA algorithm is computationally high for solving optimization problems.

4.7. Scalability study

In order to solve the large scale optimization problems, the proposed algorithm is executed on highly scalable environment. The dimensionality of the various test functions is varied from 30 to 50, 50 to 80, and 80 to 100. In Fig. 4, the performance of STOA algorithm provides different behaviors which shows the robustness of proposed algorithm. It is also observed that when it is applied on the highly scalable environment the performance degradation is not too much.

4.8. Statistical testing

The comparison of algorithms does not guarantee the efficacy and superiority of the proposed algorithm. The possibility of getting good results, by chance, cannot be ignored. For this, Wilcoxon statistical test (Wilcoxon, 1945) is performed at 5% level of significance and the $p-$ values are tabulated in Table 7. During statistical testing, the best algorithm is chosen and compared with other competitor algorithms. Whereas, $p-$ values are less than 0.05 to demonstrate the statistical superiority of proposed STOA. The test functions are taken from CEC 2015 special session which are most challenging test functions in the literature. Apart from this, Mann–Whitney U rank sum test (Mann and Whitney, 1947) has also been conducted on the average values of CEC 2015 special session test functions. Table 8 shows the results of the Mann–Whitney U rank sum test for statistical superiority of proposed STOA at 0.05 significance level. By using Mann–Whitney U rank sum test, firstly the significant difference between two data sets has been observed. If there is no significant difference then sign '=' appears. If a significant difference is observed, then signs '+' or '-' appears for the case where STOA takes less or more function evaluations than the other competitor algorithms, respectively. In Table 8, '+' shows that STOA is significantly better and '-' shows that STOA is worse. As Table 8 includes 120 '+' signs out of 135 comparisons, it can be concluded that the results of STOA are significantly effective than competitive approaches (i.e., SHO, GWO, PSO, MFO, MVO, SCA, GSA, GA, and DE) over CEC 2015 benchmark test suite. Overall, the results reveal that STOA outperforms other algorithms.

4.9. Parameters analysis of STOA

The parameter tuning of metaheuristics is one of the major challenge. As reported in existing literature it is observed that different values of parameter might yield different results. STOA algorithm involves three parameters namely number of search agents, Maximum number of iterations, and parameter C_f.

The sensitivity investigation of these parameters has been discussed by varying their values.

Table 6
Results of STOA on CEC 2015 benchmark test functions.

F	STOA	SHO	GWO	PSO	MFO	MVO	SCA	GSA	GA	DE
CEC-1	**2.59E+05**	2.28E+06	2.02E+06	4.37E+05	1.47E+06	6.06E+05	7.65E+06	3.20E+07	8.89E+06	6.09E+06
	(2.31E+05)	*(2.18E+06)*	*(2.01E+06)*	***(1.81E+05)***	*(1.00E+06)*	*(5.02E+05)*	*(3.07E+06)*	*(2.98E+06)*	*(6.95E+06)*	*(5.11E+06)*
CEC-2	5.44E+06	3.13E+05	5.65E+06	9.41E+03	1.97E+04	1.43E+04	7.33E+08	**4.58E+03**	2.97E+05	4.40E+04
	(9.75E+05)	*(2.10E+05)*	*(2.19E+06)*	*(4.82E+03)*	*(1.46E+04)*	*(1.03E+04)*	*(2.33E+08)*	***(1.09E+03)***	*(2.85E+03)*	*(2.75E+04)*
CEC-3	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**	**3.20E+02**
	(4.23E−03)	*(3.76E−02)*	*(7.08E−02)*	*(8.61E−02)*	*(9.14E−02)*	*(3.19E−02)*	*(7.53E−02)*	***(1.02E−03)***	*(2.78E−02)*	*(1.15E−03)*
CEC-4	4.11E+02	4.11E+02	4.16E+02	**4.09E+02**	4.26E+02	4.18E+02	4.42E+02	4.39E+02	6.99E+02	4.91E+06
	(2.97E+01)	*(1.71E+01)*	*(1.03E+01)*	***(3.96E+00)***	*(1.17E+01)*	*(1.03E+01)*	*(7.72E+00)*	*(7.25E+00)*	*(6.43E+00)*	*(6.03E+00)*
CEC-5	9.84E+02	9.13E+02	9.20E+02	**8.65E+02**	1.33E+03	1.09E+03	1.76E+03	1.75E+03	1.26E+03	3.34E+03
	(2.37E+02)	*(1.85E+02)*	***(1.78E+02)***	*(2.16E+02)*	*(3.45E+02)*	*(2.81E+02)*	*(2.30E+02)*	*(2.79E+02)*	*(1.86E+02)*	*(3.01E+02)*
CEC-6	2.57E+03	1.29E+04	2.26E+04	**1.86E+03**	7.35E+03	3.82E+03	2.30E+04	3.91E+06	2.91E+05	5.39E+04
	(1.42E+03)	*(1.15E+04)*	*(2.07E+04)*	***(1.28E+03)***	*(3.82E+03)*	*(2.44E+03)*	*(1.91E+04)*	*(2.70E+06)*	*(1.67E+05)*	*(2.40E+04)*
CEC-7	**7.02E+02**	**7.02E+02**	**7.02E+02**	**7.02E+02**	**7.02E+02**	**7.02E+02**	7.06E+02	7.08E+02	7.08E+02	7.06E+02
	(5.80E−01)	*(6.76E−01)*	*(7.07E−01)*	*(7.75E−01)*	*(1.10E+00)*	*(9.40E−01)*	*(9.07E−01)*	*(1.32E+00)*	*(2.97E+00)*	*(3.87E+00)*
CEC-8	**1.40E+03**	1.86E+03	3.49E+03	3.43E+03	9.93E+03	2.58E+03	6.73E+03	6.07E+05	5.79E+04	5.60E+03
	(2.57E+03)	*(1.98E+03)*	*(2.04E+03)*	*(2.77E+03)*	*(8.74E+03)*	***(1.61E+03)***	*(3.36E+03)*	*(4.81E+05)*	*(2.76E+04)*	*(5.15E+04)*
CEC-9	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**	**1.00E+03**
	(6.97E−01)	*(1.43E−01)*	*(1.28E−01)*	*(7.23E−02)*	*(2.20E−01)*	***(5.29E−02)***	*(9.79E−01)*	*(5.33E+00)*	*(3.97E+00)*	*(4.05E+00)*
CEC-10	**1.81E+03**	2.00E+03	4.00E+03	3.27E+03	8.39E+03	2.62E+03	9.91E+03	3.42E+05	4.13E+04	3.10E+04
	(1.26E+03)	*(1.79E+03)*	*(2.82E+03)*	*(1.84E+03)*	*(3.82E+03)*	***(1.18E+03)***	*(8.83E+03)*	*(1.74E+05)*	*(2.39E+04)*	*(2.76E+04)*
CEC-11	**1.35E+03**	1.38E+03	1.40E+03	1.35E+03	1.37E+03	1.39E+03	1.35E+03	1.41E+03	1.36E+03	1.65E+03
	(1.53E+01)	*(2.42E+01)*	*(5.81E+01)*	*(1.12E+02)*	*(8.97E+01)*	*(5.42E+01)*	*(1.11E+02)*	*(7.73E+01)*	*(5.39E+01)*	*(3.00E+01)*
CEC-12	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	1.31E+03	1.31E+03	1.31E+03	**1.30E+03**
	(3.01E+00)	*(7.89E−01)*	***(6.69E−01)***	*(6.94E−01)*	*(9.14E−01)*	*(8.07E−01)*	*(1.54E+00)*	*(2.05E+00)*	*(1.65E+00)*	*(9.36E−01)*
CEC-13	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	1.35E+03	1.35E+03	**1.30E+03**
	(6.49E−05)	*(2.76E−04)*	*(1.92E−04)*	*(5.44E−03)*	*(1.04E−03)*	*(2.43E−04)*	*(3.78E−03)*	*(4.70E+01)*	*(3.97E+01)*	*(4.96E−02)*
CEC-14	**3.00E+03**	4.25E+03	7.29E+03	7.10E+03	7.60E+03	7.34E+03	7.51E+03	9.30E+03	8.96E+03	6.08E+03
	(2.05E+03)	*(1.73E+03)*	*(2.45E+03)*	*(3.12E+03)*	*(1.29E+03)*	*(2.47E+03)*	*(1.52E+03)*	***(1.94E+03)***	*(6.32E+03)*	*(5.83E+03)*
CEC-15	**1.60E+03**	**1.60E+03**	1.61E+03	**1.60E+03**	1.61E+03	**1.60E+03**	1.62E+03	1.64E+03	1.63E+03	1.61E+03
	(5.29E−02)	*(3.76E+00)*	*(4.94E+00)*	***(1.06E−02)***	*(1.13E+01)*	*(1.80E−02)*	*(3.64E+00)*	*(1.12E+01)*	*(3.67E+01)*	*(1.08E+00)*

Table 7
$p-$ values obtained from Wilcoxon ranksum test for CEC 2015 benchmark functions.

CEC	SHO	GWO	PSO	MFO	MVO	SCA	GSA	GA	DE
1	0.0007	0.0002	0.0002	0.0002	0.0075	0.0001	0.0001	0.0007	0.0070
2	0.0010	0.0001	0.0001	0.0060	0.0001	0.0083	0.0038	0.0020	0.0055
3	0.0001	0.0255	0.0116	0.0179	0.0001	0.0071	0.0001	0.0121	0.0001
4	0.0067	0.0280	0.0264	0.0051	0.0037	0.0045	0.0002	0.0095	0.0001
5	0.007	0.0480	0.0001	0.0687	0.0001	0.2411	0.0011	0.0002	0.0491
6	0.0001	0.0004	0.0033	0.0050	0.8930	0.4726	0.0001	0.0046	0.0557
7	0.0090	0.0085	0.0001	0.0033	0.0033	0.0025	0.0001	0.0079	0.0013
8	0.0093	0.5915	0.0001	0.0859	0.0030	0.0001	0.0230	0.0064	0.0087
9	0.0008	0.0015	0.0001	0.0211	0.0001	0.0001	0.0093	0.0195	0.0007
10	0.0001	0.0001	0.0001	0.0807	0.0001	0.0090	0.0001	0.0550	0.0043
11	0.0027	0.0113	0.0002	0.0001	0.0008	0.0491	0.0040	0.0005	0.0110
12	0.0001	0.0092	0.0001	0.0001	0.0007	0.0001	0.0001	0.0010	0.0259
13	0.0001	0.0045	0.0001	0.0001	0.0393	0.0001	0.0030	0.0012	0.0800
14	0.0001	0.0009	0.0045	0.0001	0.0855	0.0001	0.0001	0.0065	0.0030
15	0.0022	0.0013	0.0022	0.0491	0.0050	0.0001	0.0001	0.0033	0.0090

Table 8
Mann–Whitney U rank sum test at 0.05 significance level (where '+' indicates STOA is significantly better, '-' indicates STOA is significantly worst and '=' indicates that there is no significant difference).

F	SHO	GWO	PSO	MFO	MVO	SCA	GSA	GA	DE
CEC-1	+	+	+	+	+	+	+	+	+
CEC-2	−	+	−	−	−	+	−	−	−
CEC-3	+	+	+	+	+	+	=	+	=
CEC-4	+	+	−	+	+	+	+	+	+
CEC-5	−	−	−	+	+	+	+	+	+
CEC-6	+	+	−	+	+	+	+	+	+
CEC-7	+	+	+	+	+	+	+	+	+
CEC-8	+	+	+	+	+	+	+	+	+
CEC-9	+	+	+	+	−	+	+	+	+
CEC-10	+	+	+	+	+	+	+	+	+
CEC-11	+	+	+	+	+	+	+	+	+
CEC-12	+	+	+	+	+	+	+	+	+
CEC-13	+	+	+	+	+	+	+	+	+
CEC-14	+	+	+	+	+	+	+	+	+
CEC-15	+	+	+	+	+	+	+	+	+

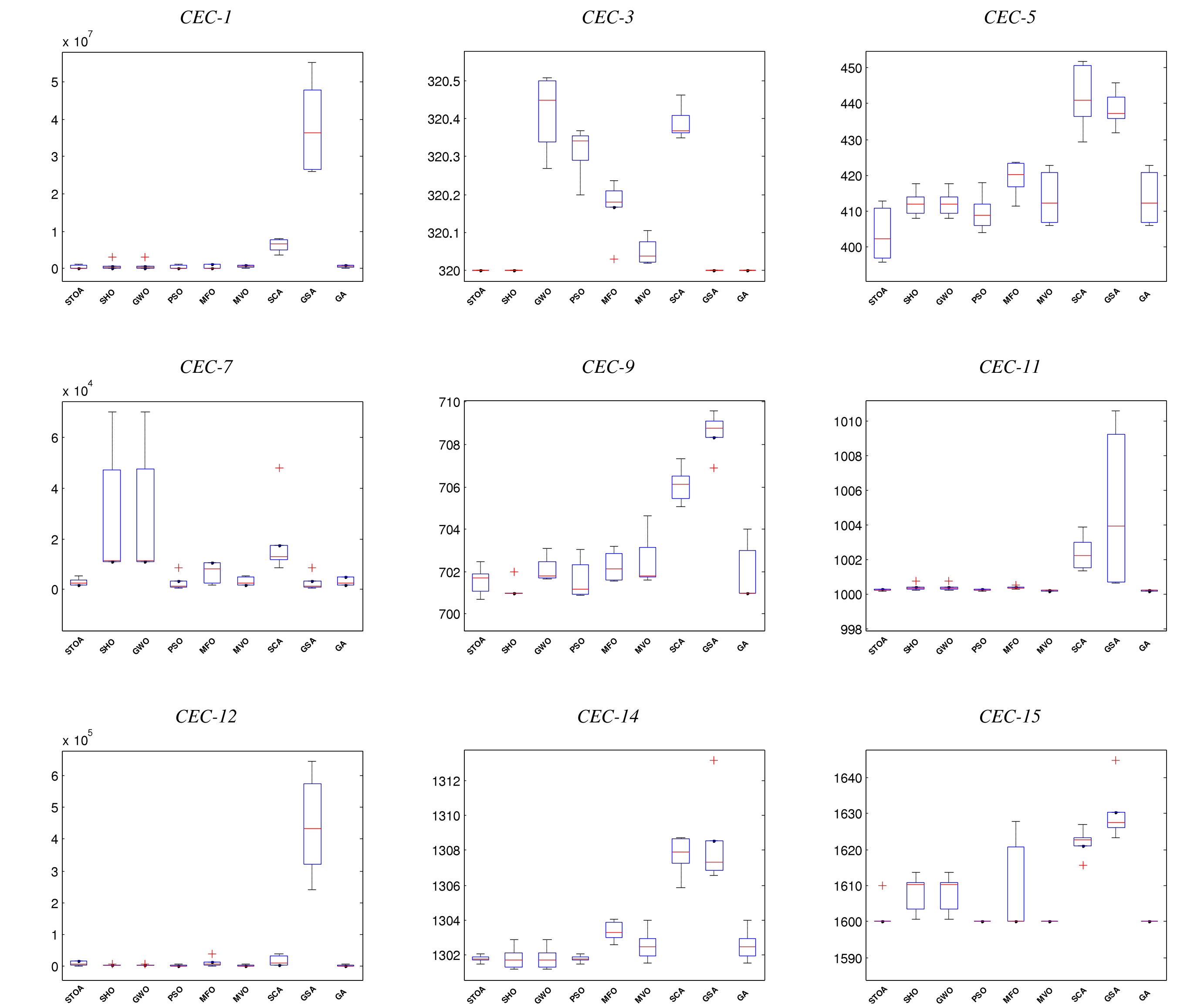

Fig. 1. Box plot obtained from proposed STOA and other competitor algorithms on CEC 2015 benchmark test functions.

1. Maximum number of iterations: STOA algorithm was run for different number of iterations. The values of $Max_{iterations}$ used in experimentation are 100, 500, 800, and 1000. Fig. 5(a) reveals that when the number of iterations are increased, STOA converges towards the optimum. The detailed analysis related to iterations process is given in Table 9.
2. Number of search agents: STOA algorithm was executed for 50, 80, 100, and 200 test functions in order to investigate the effect of number of search agents on all test function. For brevity, the $F1$ test function is chosen for analysis. Fig. 5(b) shows the effect of number of search agents on the performance of the above-mentioned algorithms. It is found that when the number of search agent is set to 100, STOA provides best optimal solutions for most of these functions. The detailed results are given in Table 10.
3. Variation in parameter C_f: STOA algorithm was run for different values of parameter C_f keeping other parameters fixed, i.e., number of search agents, and maximum number of iterations. The values of C_f used in experimentation are 1, 2, 3, 4, and 5. Fig. 5(c) shows that the function $F1$ obtains best optimal solution when the value of C_f is set to 2. It is observed that $C_f = 2$ is a reasonable value for most of the test functions. The detailed sensitivity results related to C_f are given in Table 11.

5. Industrial engineering applications

In this section, the efficiency of proposed STOA algorithm should be investigated in solving six constrained optimization problems and compared with other state-of-the-art algorithms. These problems have several equality and inequality constraints. STOA algorithm should be equipped with a constraint handling method to optimize these problems.

5.1. Constraint handling

Constraint handling is one of the main challenges in solving optimization problems using metaheuristic techniques. There are five constraint handling techniques (Coello, 2002): penalty functions, hybrid methods, separation of objective functions and constraints, repair algorithms, and special operators. Among these techniques, the penalty functions are simple and easy to implement. There are distinct penalty functions such as static, annealing, adaptive, co-evolutionary,

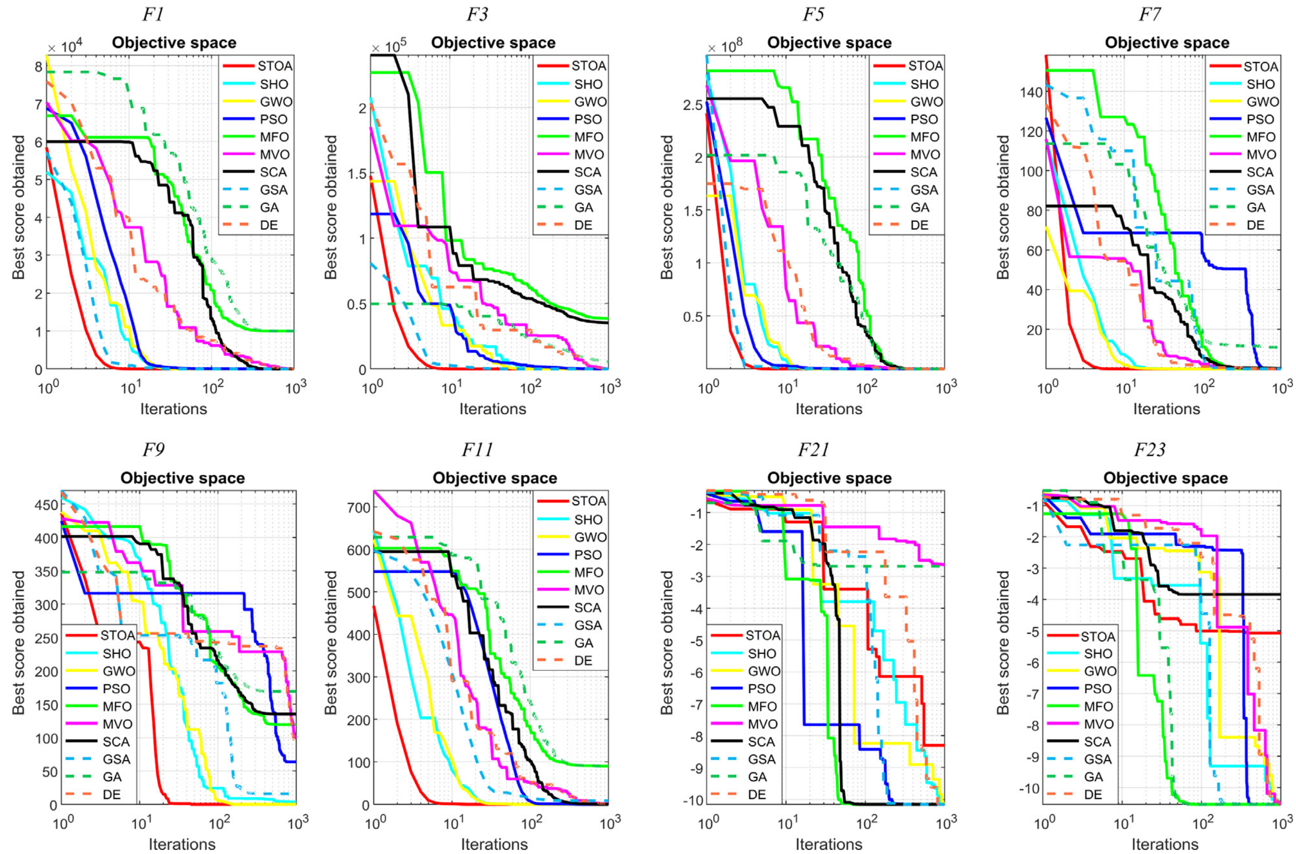

Fig. 2. Convergence analysis of proposed STOA and competitor algorithms on some of the benchmark test functions.

Table 9
The obtained optimal values on unimodal, multimodal, fixed-dimension multimodal, CEC 2005, and CEC 2015 benchmark test functions using different simulation runs. The best results are in bold.

Functions	Iterations	Optimal values
F_1	100	6.44E+00
	500	9.16E−16
	800	3.54E−29
	1000	**0.00E+00**
F_5	100	5.86E+11
	500	7.37E+08
	800	4.14E+03
	1000	**7.14E+00**
F_{11}	100	3.04E+01
	500	6.16E−02
	800	2.77E−03
	1000	**0.00E+00**
F_{17}	100	3.59E+03
	500	2.72E+00
	800	9.90E−01
	1000	**4.04E−01**
F_{23}	100	−2.03E+00
	500	−9.80E+00
	800	−1.01E+01
	1000	**−1.07E+01**
$CF1$	100	1.80E+08
	500	4.437+05
	800	8.96E+03
	1000	**4.12E+01**
$CEC-1$	100	7.39E+11
	500	2.53E+08
	800	5.03E+06
	1000	**3.05E+05**

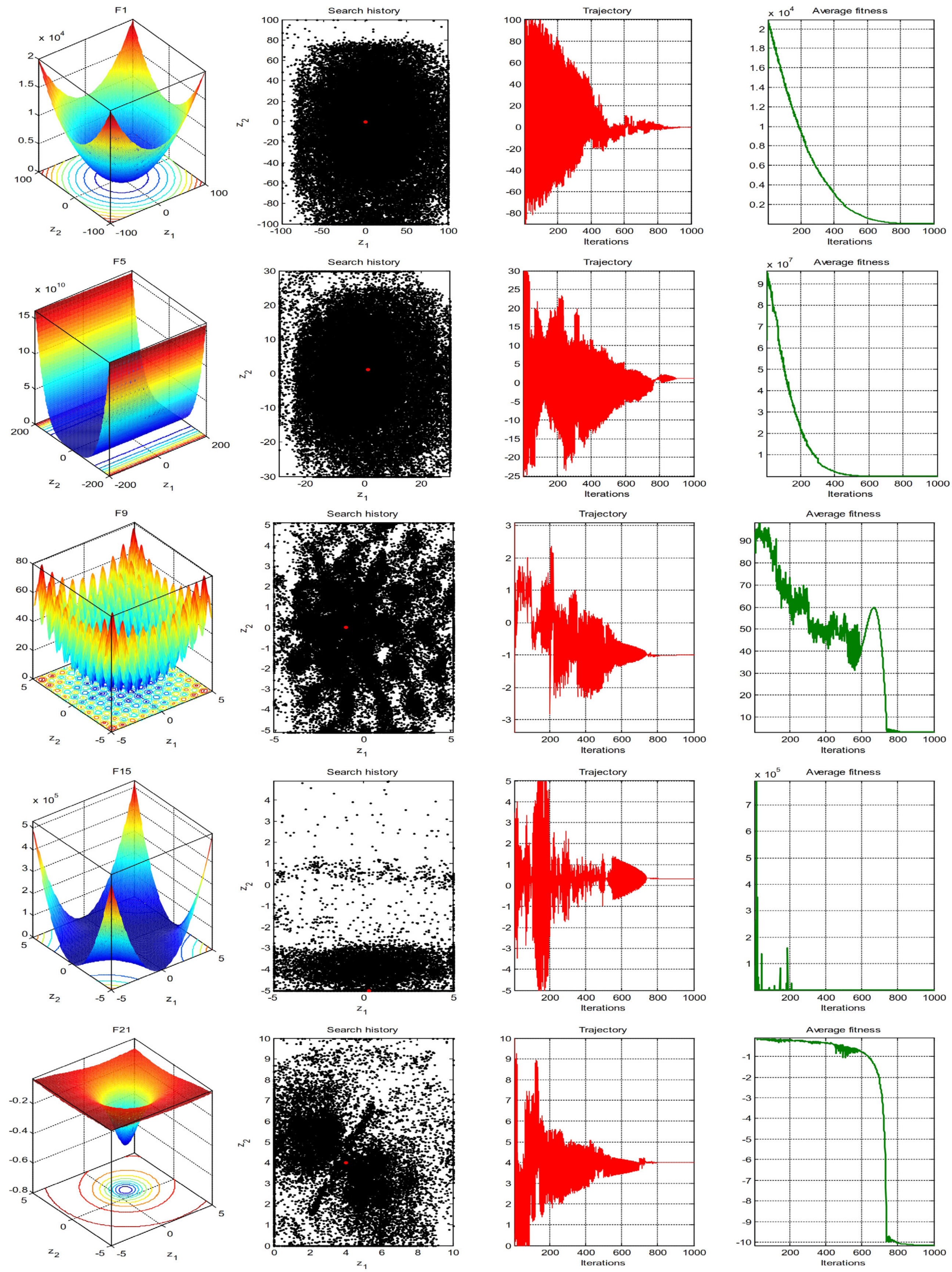

Fig. 3. Search history, trajectory, and average fitness of STOA algorithm on 2D benchmark test problems.

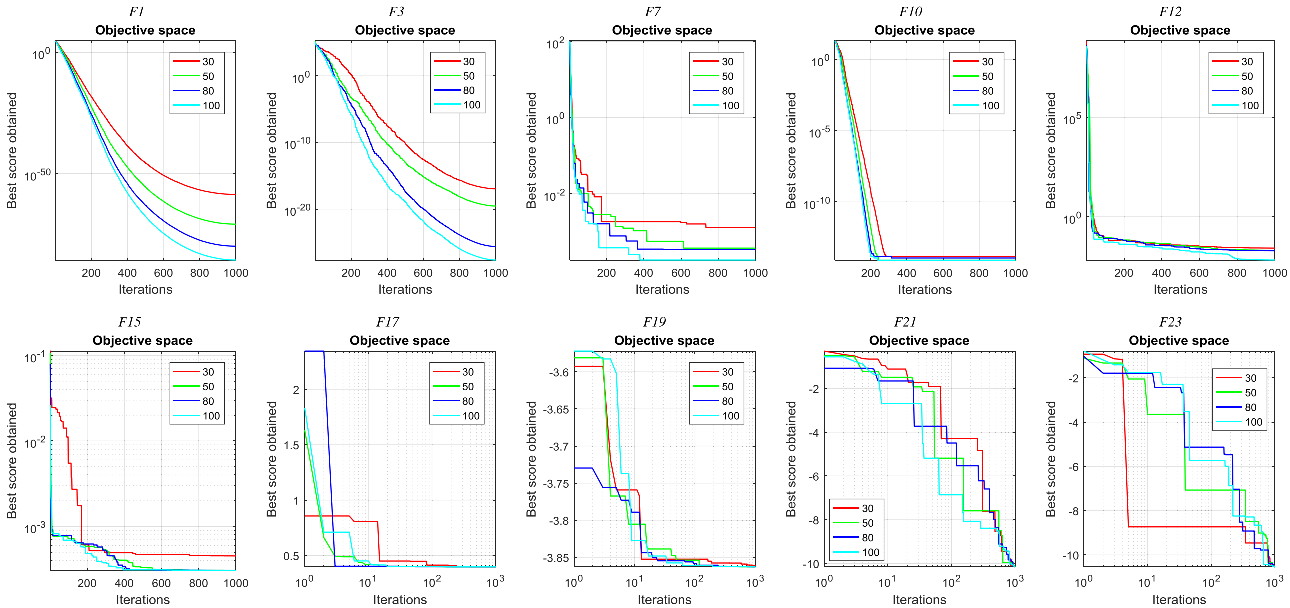

Fig. 4. Scalability analysis of proposed STOA algorithm on some of the benchmark test functions.

Table 10
The obtained optimal values on unimodal, multimodal, fixed-dimension multimodal, CEC 2005, and CEC 2015 benchmark test functions where number of iterations is fixed as 1000. The number of search agents are varied from 50 to 200. The best results are in bold.

Functions	Search agents	Optimal values
F_1	**50**	**0.00E+00**
	80	**0.00E+00**
	100	**0.00E+00**
	200	**0.00E+00**
F_5	50	6.29E+01
	80	8.46E+01
	100	**7.17E+00**
	200	9.16E+00
F_{11}	**50**	**0.00E+00**
	80	**0.00E+00**
	100	**0.00E+00**
	200	**0.00E+00**
F_{17}	50	9.73E−01
	80	6.87E−01
	100	**4.07E−01**
	200	8.41E−01
F_{23}	50	−4.42E+00
	80	−2.48E+00
	100	**−1.13E+01**
	200	−1.09E+01
$CF1$	50	7.43E+01
	80	9.75E+01
	100	**4.01E+01**
	200	4.37E+01
$CEC-1$	50	6.03E+05
	80	4.57E+06
	100	**2.81E+05**
	200	4.35E+05

and death penalty. These approaches converted constrained problems into unconstraint problem by adding some penalty values. In this paper, a static penalty approach is used to handle constraints in optimization problems.

$$\zeta(z) = f(z) \pm \left[\sum_{i=1}^{m} l_i \times max(0, t_i(z))^{\alpha} + \sum_{j=1}^{n} o_j \times \mid U_j(z) \mid \right] \tag{11}$$

where $\zeta(z)$ is the modified objective function and l_i, o_j are positive penalty values. This approach assigns the penalty value for each infeasible solution. In death penalty approach, a large value is assigned to objective function of infeasible solution. Therefore, the static penalty function is employed which helps the search agents to move towards the feasible search space of the problem.

Table 11
The obtained optimal values on unimodal, multimodal, fixed-dimension multimodal, CEC 2005, and CEC 2015 benchmark test functions where number of iterations and search agents are fixed as 1000 and 100, respectively. The parameter C_f is varied from 1 to 5. The best results are in bold.

Functions	C_f	Optimal values
F_1	1	3.47E−10
	2	**0.00E+00**
	3	2.48E−27
	4	4.39E−21
	5	5.00E−14
F_5	1	5.45E+03
	2	**7.09E+00**
	3	5.02E+01
	4	3.46E+01
	5	1.73E+05
F_{11}	1	6.46E−12
	2	**0.00E+00**
	3	3.40E−25
	4	5.24E−20
	5	2.62E−17
F_{17}	1	1.23E+03
	2	**3.97E−01**
	3	5.45E+01
	4	9.00E+05
	5	1.79E+02
F_{23}	1	−5.46E+00
	2	**−1.10E+01**
	3	−2.13E+00
	4	−8.40E+00
	5	−9.89E+00
$CF1$	1	1.26E+07
	2	**3.90E+01**
	3	9.29E+02
	4	1.02E+04
	5	5.43E+04
$CEC-1$	1	1.05E+10
	2	**2.66E+05**
	3	5.05E+13
	4	2.48E+10
	5	8.73E+08

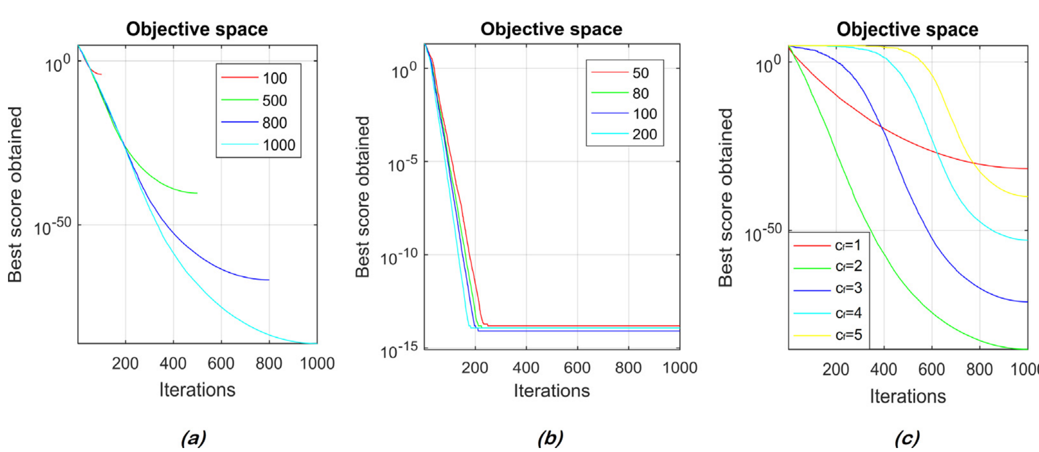

Fig. 5. Parameters analysis of proposed STOA algorithm, (a) Effect of iterations; (b) Effect of search agents; (c) Effect of parameter C_f.

5.2. Pressure vessel design problem

This problem was first proposed by Kannan and Kramer (1994) to minimize the total fabrication cost. Fig. 6 shows the schematic view of pressure vessel which are capped at both ends by hemispherical heads. There are four variables in this problem:

- T_s (z_1, thickness of the shell).
- T_h (z_2, thickness of the head).
- R (z_3, inner radius).
- L (z_4, length of the cylindrical section without considering the head).

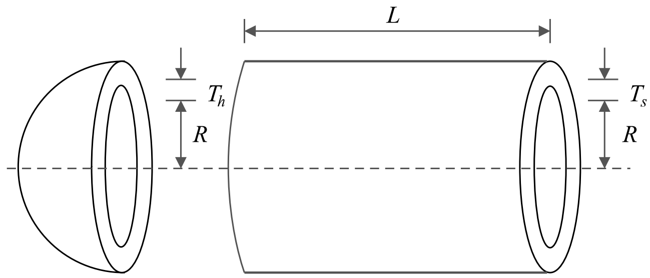

Fig. 6. Schematic view of pressure vessel problem.

Among these four design variables, R and L are continuous variables. T_s and T_h are integer values which are multiples of 0.0625 *in*. The mathematical formulation of this problem is given below:

$$\begin{aligned}
&\text{Consider } \vec{z} = [z_1\ z_2\ z_3\ z_4] = [T_s\ T_h\ R\ L],\\
&\text{Minimize } f(\vec{z}) = 0.6224 z_1 z_3 z_4 + 1.7781 z_2 z_3^2 + 3.1661 z_1^2 z_4 + 19.84 z_1^2 z_3,\\
&\text{Subject to:}\\
&g_1(\vec{z}) = -z_1 + 0.0193 z_3 \le 0,\\
&g_2(\vec{z}) = -z_3 + 0.00954 z_3 \le 0,\\
&g_3(\vec{z}) = -\pi z_3^2 z_4 - \frac{4}{3}\pi z_3^3 + 1,296,000 \le 0,\\
&g_4(\vec{z}) = z_4 - 240 \le 0,\\
&\text{where,}\\
&1 \times 0.0625 \le z_1,\ z_2 \le 99 \times 0.0625,\ 10.0 \le z_3,\ z_4 \le 200.0.
\end{aligned} \tag{12}$$

Table 12 reveals the best comparison of optimal solution which is obtained so far from STOA and other methods. STOA generates optimal result at $z^* = (0.778095, 0.383240, 40.315118, 200.00000)$ and its fitness value as $f(z^*) = 5879.1253$. It can be seen from Table that STOA is able to find design this problem with low computational cost. The statistical results of pressure vessel design problem are tabulated in Table 13. The results show that STOA performance is better than the other approaches. Fig. 7 shows the convergence analysis of optimal solution at different iterations obtained from proposed STOA.

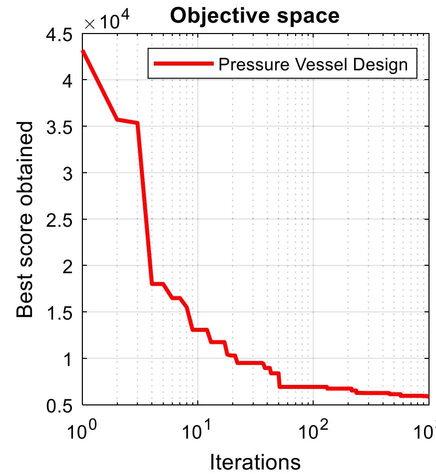

Fig. 7. Convergence analysis of STOA for pressure vessel design problem.

5.3. Speed reducer design problem

The speed reducer design problem is a challenging benchmark due to its seven design variables (Gandomi and Yang, 2011) as shown in Fig. 8. The main objective of this problem is to minimize the weight of speed reducer with subject to constraints (Mezura-Montes and Coello, 2005):

- Bending stress of the gear teeth.
- Surface stress.
- Transverse deflections of the shafts.
- Stresses in the shafts.

There are seven design variables ($z_1 - z_7$) which can represent as the face width (b), module of teeth (m), number of teeth in the pinion (p), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), the diameter of first (d_1) shafts, and the diameter of second shafts (d_2). The third variable i.e., number of teeth in the pinion (p) is of integer values.

The mathematical formulation of this problem is formulated as follows:

$$\begin{aligned}
&\text{Consider } \vec{z} = [z_1\ z_2\ z_3\ z_4\ z_5\ z_6\ z_7] = [b\ m\ p\ l_1\ l_2\ d_1\ d_2],\\
&\text{Minimize } f(\vec{z}) = 0.7854 z_1 z_2^2 (3.3333 z_3^2 + 14.9334 z_3 - 43.0934)\\
&- 1.508 z_1 (z_6^2 + z_7^2) + 7.4777 (z_6^3 + z_7^3) + 0.7854 (z_4 z_6^2 + z_5 z_7^2),\\
&\text{Subject to:}\\
&g_1(\vec{z}) = \frac{27}{z_1 z_2^2 z_3} - 1 \le 0,\\
&g_2(\vec{z}) = \frac{397.5}{z_1 z_2^2 z_3^2} - 1 \le 0,\\
&g_3(\vec{z}) = \frac{1.93 z_4^3}{z_2 z_6^4 z_3} - 1 \le 0,\\
&g_4(\vec{z}) = \frac{1.93 z_5^3}{z_2 z_7^4 z_3} - 1 \le 0,\\
&g_5(\vec{z}) = \frac{[(745(z_4/z_2 z_3))^2 + 16.9 \times 10^6]^{1/2}}{110 z_6^3} - 1 \le 0,\\
&g_6(\vec{z}) = \frac{[(745(z_5/z_2 z_3))^2 + 157.5 \times 10^6]^{1/2}}{85 z_7^3} - 1 \le 0,\\
&g_7(\vec{z}) = \frac{z_2 z_3}{40} - 1 \le 0,\\
&g_8(\vec{z}) = \frac{5 z_2}{z_1} - 1 \le 0,\\
&g_9(\vec{z}) = \frac{z_1}{12 z_2} - 1 \le 0,\\
&g_{10}(\vec{z}) = \frac{1.5 z_6 + 1.9}{z_4} - 1 \le 0,\\
&g_{11}(\vec{z}) = \frac{1.1 z_7 + 1.9}{z_5} - 1 \le 0,\\
&\text{where,}\\
&2.6 \le z_1 \le 3.6,\ 0.7 \le z_2 \le 0.8,\ 17 \le z_3 \le 28,\ 7.3 \le z_4 \le 8.3,\\
&7.3 \le z_5 \le 8.3,\ 2.9 \le z_6 \le 3.9,\ 5.0 \le z_7 \le 5.5.
\end{aligned} \tag{13}$$

The obtained solutions and comparison of best optimum is tabulated in Table 14. STOA generates optimal solution at points $z^* = (3.50124, 0.7, 17, 7.3, 7.8, 3.33425, 5.26538)$ and its fitness value as $f(z^*) = 2995.9578$. The statistical results of STOA and eight well-known optimization approaches are presented in Table 15. The results show that STOA outperforms than other methods. Fig. 9 shows that STOA algorithm converges towards the optimal solution for speed reducer design problem.

Table 12
Comparison results for pressure vessel design problem.

Algorithms	Optimum variables				Optimum cost
	T_s	T_h	R	L	
STOA	0.778095	0.383240	40.315118	200.00000	**5879.1253**
SHO	0.778210	0.384889	40.315040	200.00000	5885.5773
GWO	0.779035	0.384660	40.327793	199.65029	5889.3689
PSO	0.778961	0.384683	40.320913	200.00000	5891.3879
MVO	0.845719	0.418564	43.816270	156.38164	6011.5148
SCA	0.817577	0.417932	41.74939	183.57270	6137.3724
GSA	1.085800	0.949614	49.345231	169.48741	11550.2976
GA	0.752362	0.399540	40.452514	198.00268	5890.3279
DE	1.099523	0.906579	44.456397	179.65887	6550.0230

Table 13
Statistical results obtained from different algorithms for pressure vessel design problem.

Algorithms	Best	Mean	Worst	Std. Dev.	Median
STOA	**5879.1253**	**5883.2236**	5890.5475	036.448	**5882.9574**
SHO	5885.5773	5887.4441	5892.3207	**002.893**	5886.2282
GWO	5889.3689	5891.5247	5894.6238	013.910	5890.6497
PSO	5891.3879	6531.5032	7394.5879	534.119	6416.1138
MVO	6011.5148	6477.3050	7250.9170	327.007	6397.4805
SCA	6137.3724	6326.7606	6512.3541	126.609	6318.3179
GSA	11550.2976	23342.2909	**33226.2526**	5790.625	24010.0415
GA	5890.3279	6264.0053	7005.7500	496.128	6112.6899
DE	6550.0230	6643.9870	8005.4397	657.523	7586.0085

Table 14
Comparison results for speed reducer design problem.

Algorithms	Optimum variables							Optimum cost
	b	m	p	l_1	l_2	d_1	d_2	
STOA	3.50124	0.7	17	7.3	7.8	3.33425	5.26538	**2995.9578**
SHO	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
GWO	3.506690	0.7	17	7.380933	7.815726	3.357847	5.286768	3001.288
PSO	3.500019	0.7	17	8.3	7.8	3.352412	5.286715	3005.763
MVO	3.508502	0.7	17	7.392843	7.816034	3.358073	5.286777	3002.928
SCA	3.508755	0.7	17	7.3	7.8	3.461020	5.289213	3030.563
GSA	3.600000	0.7	17	8.3	7.8	3.369658	5.289224	3051.120
GA	3.510253	0.7	17	8.35	7.8	3.362201	5.287723	3067.561
DE	3.520124	0.7	17	8.37	7.8	3.366970	5.288719	3029.002

Table 15
Statistical results obtained from different algorithms for speed reducer design problem.

Algorithms	Best	Mean	Worst	Std. Dev.	Median
STOA	**2995.9578**	**2998.754**	2930.745	**1.79526**	**2997.555**
SHO	2998.5507	2999.640	3003.889	1.93193	2999.187
GWO	3001.288	3005.845	3008.752	5.83794	3004.519
PSO	3005.763	3105.252	3211.174	79.6381	3105.252
MVO	3002.928	3028.841	3060.958	13.0186	3027.031
SCA	3030.563	3065.917	3104.779	18.0742	3065.609
GSA	3051.120	3170.334	**3363.873**	92.5726	3156.752
GA	3067.561	3186.523	3313.199	17.1186	3198.187
DE	3029.002	3295.329	3619.465	57.0235	3288.657

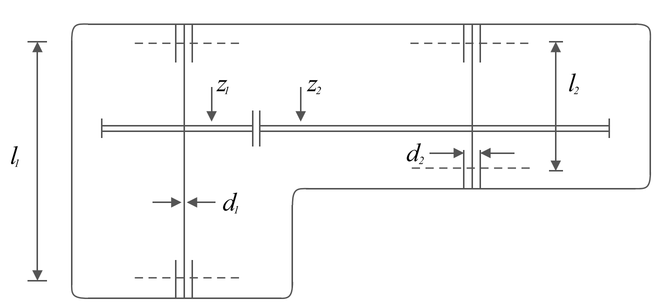

Fig. 8. Schematic view of speed reducer problem.

5.4. Welded beam design problem

The objective of this design problem is to minimize the fabrication cost of welded beam (see Fig. 10). The optimization constraints of welded beam are shear stress (τ) and bending stress (θ) in the beam, buckling load (P_c) on the bar, end deflection (δ) of the beam. There are four design variables of this problem such as

- Thickness of weld (h)
- Length of the clamped bar (l)
- Height of the bar (t)
- Thickness of the bar (b)

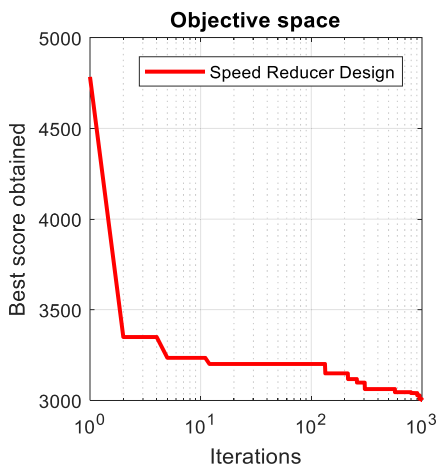

Fig. 9. Convergence analysis of STOA for speed reducer design problem.

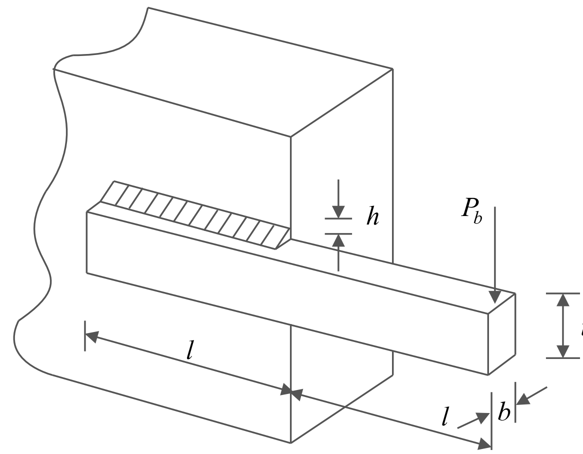

Fig. 10. Schematic view of welded beam problem.

The mathematical formulation is described as follows:

Consider $\vec{z} = [z_1\ z_2\ z_3\ z_4] = [h\ l\ t\ b]$,

Minimize $f(\vec{z}) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14.0 + z_2)$,

Subject to:

$$
\begin{aligned}
&g_1(\vec{z}) = \tau(\vec{z}) - 13,600 \le 0, \\
&g_2(\vec{z}) = \sigma(\vec{z}) - 30,000 \le 0, \\
&g_3(\vec{z}) = \delta(\vec{z}) - 0.25 \le 0, \\
&g_4(\vec{z}) = z_1 - z_4 \le 0, \\
&g_5(\vec{z}) = 6,000 - P_c(\vec{z}) \le 0, \\
&g_6(\vec{z}) = 0.125 - z_1 \le 0, \\
&g_7(\vec{z}) = 1.10471 z_1^2 + 0.04811 z_3 z_4 (14.0 + z_2) - 5.0 \le 0,
\end{aligned} \tag{14}
$$

where,

$0.1 \le z_1,\ 0.1 \le z_2,\ z_3 \le 10.0,\ z_4 \le 2.0,$

$$\tau(\vec{z}) = \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'')/\sqrt{0.25(l^2 + (h+t)^2)}},$$

$$\tau' = \frac{6,000}{\sqrt{2}hl}, \qquad \sigma(\vec{z}) = \frac{504,000}{t^2 b}, \qquad \delta(\vec{z}) = \frac{65,856,000}{(30 \times 10^6) b t^3},$$

$$\tau'' = \frac{6,000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2[0.707hl(l^2/12 + 0.25(h+t)^2)]},$$

$$P_c(\vec{z}) = 64,746.022(1 - 0.0282346t)tb^3.$$

The best obtained solutions from proposed STOA and competitor approaches is presented in Table 16. The proposed STOA algorithm produces optimal result at points $z^* = (0.205415, 3.472346, 9.035220, 0.201160)$ and its fitness value as $f(z*) = 1.723590$. The results reveal that STOA converges towards the best optimal design. The statistical comparison between proposed and other algorithms is tabulated in Table 17. It can be seen that STOA achieve better results than other algorithms in terms of best, mean, and median measurements. Fig. 11 shows the convergence behavior of STOA for finding best optimal design.

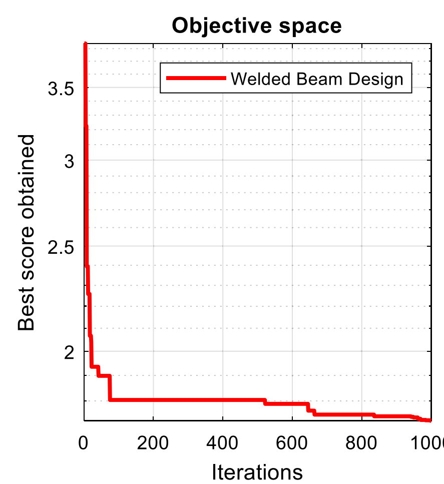

Fig. 11. Convergence analysis of STOA for welded beam design problem.

5.5. Tension/compression spring design problem

The main objective of this problem is to minimize the tension/compression spring weight as shown in Fig. 12. The optimization constraints of this problem are described as follows:

- Shear stress.
- Surge frequency.
- Minimum deflection.

There are three design variables such as wire diameter (d), mean coil diameter (D), and the number of active coils (P).

The mathematical representation of this problem is described as follows:

Consider $\vec{z} = [z_1\ z_2\ z_3] = [d\ D\ P]$,

Minimize $f(\vec{z}) = (z_3 + 2) z_2 z_1^2$,

Subject to:

$$
\begin{aligned}
&g_1(\vec{z}) = 1 - \frac{z_2^3 z_3}{71785 z_1^4} \le 0, \\
&g_2(\vec{z}) = \frac{4z_2^2 - z_1 z_2}{12566(z_2 z_1^3 - z_1^4)} + \frac{1}{5108 z_1^2} \le 0, \\
&g_3(\vec{z}) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \le 0, \\
&g_4(\vec{z}) = \frac{z_1 + z_2}{1.5} - 1 \le 0,
\end{aligned} \tag{15}
$$

where,

$0.05 \le z_1 \le 2.0,\ 0.25 \le z_2 \le 1.3,\ 2.0 \le z_3 \le 15.0.$

The comparison between proposed STOA and other competitive methods is depicted in Table 18. The best solution obtained by STOA at design variables $z^* = (0.051090, 0.342910, 12.0900)$ with its corresponding objective function value as $f(z^*) = 0.012656990$. The results

Table 16
Comparison results for welded beam design problem.

Algorithms	Optimum variables				Optimum cost
	h	*l*	*t*	*b*	
STOA	0.205415	3.472346	9.035220	0.201160	**1.723590**
SHO	0.205563	3.474846	9.035799	0.205811	1.725661
GWO	0.205678	3.475403	9.036964	0.206229	1.726995
PSO	0.197411	3.315061	10.00000	0.201395	1.820395
MVO	0.205611	3.472103	9.040931	0.205709	1.725472
SCA	0.204695	3.536291	9.004290	0.210025	1.759173
GSA	0.147098	5.490744	10.00000	0.217725	2.172858
GA	0.164171	4.032541	10.00000	0.223647	1.873971
DE	0.206487	3.635872	10.00000	0.203249	1.836250

Table 17
Statistical results obtained from different algorithms for welded beam design problem.

Algorithms	Best	Mean	Worst	Std. Dev.	Median
STOA	**1.723590**	**1.725126**	1.727215	0.004330	**1.724401**
SHO	1.725661	1.725828	1.726064	**0.000287**	1.725787
GWO	1.726995	1.727128	1.727564	0.001157	1.727087
PSO	1.820395	2.230310	**3.048231**	0.324525	2.244663
MVO	1.725472	1.729680	1.741651	0.004866	1.727420
SCA	1.759173	1.817657	1.873408	0.027543	1.820128
GSA	2.172858	2.544239	3.003657	0.255859	2.495114
GA	1.873971	2.119240	2.320125	0.034820	2.097048
DE	1.836250	1.363527	2.035247	0.139485	1.9357485

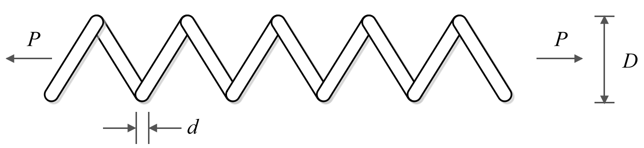

Fig. 12. Schematic view of tension/compression spring problem.

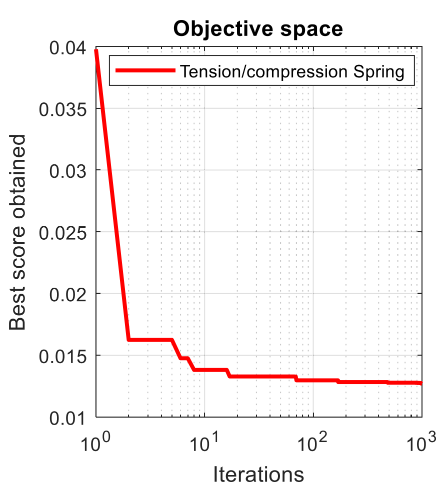

Fig. 13. Convergence analysis of STOA for tension/compression spring design problem.

show that STOA performs best than the other methods. The statistical results are compared and depicted in Table 19. It is observed that STOA obtains better statistical results in terms of best, mean, and median measurements. The convergence behavior of optimal solution obtained from proposed STOA is shown in Fig. 13.

5.6. 25-bar truss design problem

The truss design problem is a popular optimization problem as shown in Fig. 14. There are 10 nodes which are fixed and 25 bars cross-sectional members which are grouped into eight categories.

- Group 1: A_1
- Group 2: A_2, A_3, A_4, A_5
- Group 3: A_6, A_7, A_8, A_9
- Group 4: A_{10}, A_{11}
- Group 5: A_{12}, A_{13}
- Group 6: A_{14}, A_{15}, A_{17}
- Group 7: $A_{18}, A_{19}, A_{20}, A_{21}$
- Group 8: $A_{22}, A_{23}, A_{24}, A_{25}$

The other variables which affects on this problem are as follows:

- $p = 0.0272$ N/cm^3 (0.1 lb/in.^{3})
- E = 68947 MPa (10000 Ksi)
- Displacement limitation = 0.35 in.
- Maximum displacement = 0.3504 in.
- Design variable set = $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4\}$

Table 20 shows the member stress limitations for this truss problem. The loading conditions for 25-bar truss is presented in Table 21.

The comparison results with proposed and other algorithms is given in Table 22. STOA is better than other competitor algorithms in terms of best, average, and standard deviation. STOA convergence behavior to optimize this problem is shown in Fig. 15.

5.7. Rolling element bearing design problem

The objective of this problem is to maximize the dynamic load carrying capacity of a rolling element bearing as demonstrated in Fig. 16. There are 10 decision variables such as pitch diameter (D_m), ball diameter (D_b), number of balls (Z), inner (f_i) and outer (f_o)

Table 18
Comparison results for tension/compression spring design problem.

Algorithms	Optimum variables			Optimum cost
	d	*D*	*P*	
STOA	0.051090	0.342910	12.0900	**0.012656990**
SHO	0.051144	0.343751	12.0955	0.012674000
GWO	0.050178	0.341541	12.07349	0.012678321
PSO	0.05000	0.310414	15.0000	0.013192580
MVO	0.05000	0.315956	14.22623	0.012816930
SCA	0.050780	0.334779	12.72269	0.012709667
GSA	0.05000	0.317312	14.22867	0.012873881
GA	0.05010	0.310111	14.0000	0.013036251
DE	0.05025	0.316351	15.23960	0.012776352

Table 19
Statistical results obtained from different algorithms for tension/compression spring design problem.

Algorithms	Best	Mean	Worst	Std. Dev.	Median
STOA	**0.012656990**	**0.012678905**	0.012667907	0.001030	**0.012676005**
SHO	0.012674000	0.012684106	0.012715185	**0.000027**	0.012687293
GWO	0.012678321	0.012697116	0.012720757	0.000041	0.012699686
PSO	0.013192580	0.014817181	**0.017862507**	0.002272	0.013192580
MVO	0.012816930	0.014464372	0.017839737	0.001622	0.014021237
SCA	0.012709667	0.012839637	0.012998448	0.000078	0.012844664
GSA	0.012873881	0.013438871	0.014211731	0.000287	0.013367888
GA	0.013036251	0.014036254	0.016251423	0.002073	0.013002365
DE	0.012776352	0.013069872	0.015214230	0.000375	0.012952142

Table 20
Member stress limitations for 25-bar truss design problem.

Element group	Compressive stress limitations Ksi (MPa)	Tensile stress limitations Ksi (MPa)
Group 1	35.092 (241.96)	40.0 (275.80)
Group 2	11.590 (79.913)	40.0 (275.80)
Group 3	17.305 (119.31)	40.0 (275.80)
Group 4	35.092 (241.96)	40.0 (275.80)
Group 5	35.092 (241.96)	40.0 (275.80)
Group 6	6.759 (46.603)	40.0 (275.80)
Group 7	6.959 (47.982)	40.0 (275.80)
Group 8	11.082 (76.410)	40.0 (275.80)

Table 21
Two loading conditions for the 25-bar truss design problem.

Node	Case 1			Case 2		
	$P_x Kips(kN)$	$P_y Kips(kN)$	$P_z Kips(kN)$	$P_x Kips(kN)$	$P_y Kips(kN)$	$P_z Kips(kN)$
1	0.0	20.0 (89)	−5.0 (22.25)	1.0 (4.45)	10.0 (44.5)	−5.0 (22.25)
2	0.0	−20.0 (89)	−5.0 (22.25)	0.0	10.0 (44.5)	−5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 22
Statistical results obtained from different algorithms for 25-bar truss design problem.

Groups	STOA	ACO (Camp and Bichon, 2004)	PSO (Schutte and Groenwold, 2003)	CSS (Kaveh and Talatahari, 2010b)	BB-BC (Kaveh and Talatahari, 2009)
$A1$	**0.01**	0.01	0.01	0.01	0.01
$A2-A5$	**1.853**	2.042	2.052	2.003	1.993
$A6-A9$	**3.000**	3.001	3.001	3.007	3.056
$A10-A11$	**0.01**	0.01	0.01	0.01	0.01
$A12-A13$	**0.01**	0.01	0.01	0.01	0.01
$A14-A17$	**0.665**	0.684	0.684	0.687	0.665
$A18-A21$	1.621	1.632	**1.616**	1.655	1.642
$A22-A25$	2.671	2.670	2.673	**2.66**	2.679
Best weight	**545.00**	545.03	545.21	545.10	545.16
Average weight	**545.42**	545.74	546.84	545.58	545.66
Std. dev.	**0.401**	0.94	1.478	0.412	0.491

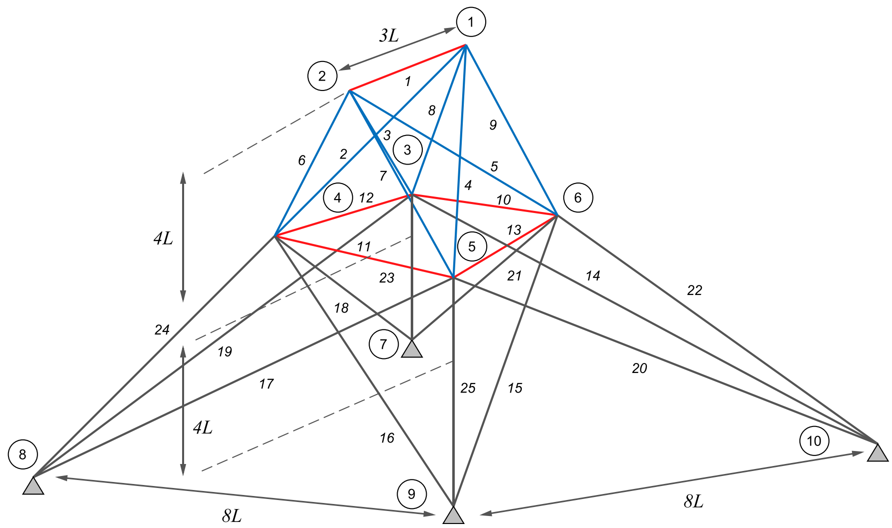

Fig. 14. Schematic view of 25-bar truss problem.

raceway curvature coefficients, K_{Dmin}, K_{Dmax}, ε, e, and ζ (see Fig. 16). The mathematical representation of this problem is given below:

$$\text{Maximize } C_d = \begin{cases} f_c Z^{2/3} D_b^{1.8}, & if \;\; D \le 25.4\text{mm} \\ C_d = 3.647 f_c Z^{2/3} D_b^{1.4}, & if \;\; D > 25.4\text{mm} \end{cases}$$

Subject to:

$$g_1(\vec{z}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \le 0,$$
$$g_2(\vec{z}) = 2D_b - K_{Dmin}(D-d) \ge 0,$$
$$g_3(\vec{z}) = K_{Dmax}(D-d) - 2D_b \ge 0,$$
$$g_4(\vec{z}) = \zeta B_w - D_b \le 0,$$
$$g_5(\vec{z}) = D_m - 0.5(D+d) \ge 0,$$
$$g_6(\vec{z}) = (0.5+e)(D+d) - D_m \ge 0,$$
$$g_7(\vec{z}) = 0.5(D - D_m - D_b) - \varepsilon D_b \ge 0,$$
$$g_8(\vec{z}) = f_i \ge 0.515,$$
$$g_9(\vec{z}) = f_o \ge 0.515,$$

where,

$$x = [\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2] \quad (16)$$
$$y = 2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}$$
$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3}$$
$$\times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41}$$
$$\phi_o = 2\pi - 2\cos^{-1}\left(\frac{x}{y}\right)$$
$$\gamma = \frac{D_b}{D_m}, \quad f_i = \frac{r_i}{D_b}, \quad f_o = \frac{r_o}{D_b}, \quad T = D - d - 2D_b$$
$$D = 160, \quad d = 90, \quad B_w = 30, \quad r_i = r_o = 11.033$$
$$0.5(D+d) \le D_m \le 0.6(D+d), \quad 0.15(D-d) \le D_b$$
$$\le 0.45(D-d), \quad 4 \le Z \le 50, \quad 0.515 \le f_i \text{ and } f_o \le 0.6,$$
$$0.4 \le K_{Dmin} \le 0.5, \quad 0.6 \le K_{Dmax} \le 0.7, \quad 0.3 \le e \le 0.4,$$
$$0.02 \le e \le 0.1, \quad 0.6 \le \zeta \le 0.85.$$

Table 23 reveals the performance of optimal solution which is obtained so far from different algorithms. STOA obtains best optimal solution at points $z^* = (125, 21.41890, 10.94113, 0.515, 0.515, 0.4, 0.7, 0.3, 0.02, 0.6)$ and its corresponding fitness value as $f(z^*) = 85067.983$.

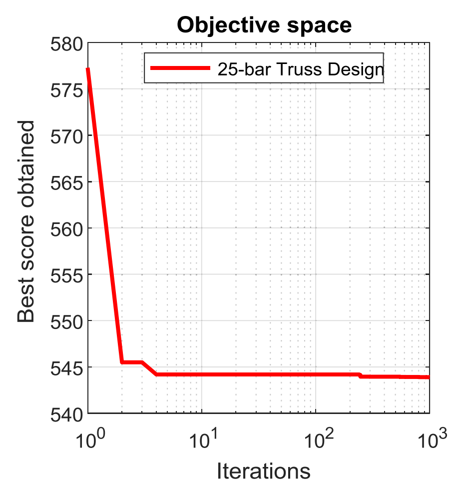

Fig. 15. Convergence analysis of STOA for 25-bar truss design problem.

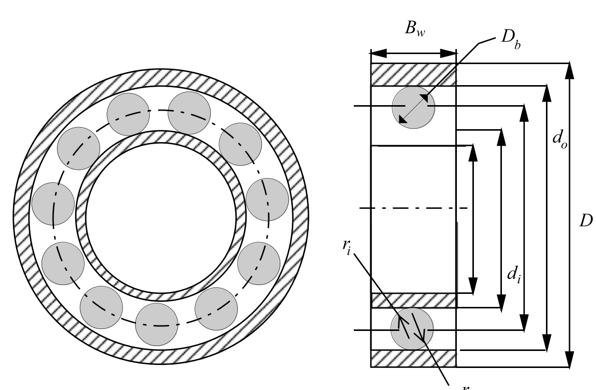

Fig. 16. Schematic view of rolling element bearing problem.

The statistical results generated from algorithms are compared and tabulated in Table 24.

Fig. 17 shows the convergence behavior of STOA approach. It can be seen STOA has an ability to achieve best optimal solution.

Table 23
Comparison results for Rolling element bearing design problem.

Algorithms	Optimum variables										Opt. cost
	D_m	D_b	Z	f_i	f_o	K_{Dmin}	K_{Dmax}	ε	e	ζ	
STOA	125	21.41890	10.94113	0.515	0.515	0.4	0.7	0.3	0.02	0.6	**85067.983**
SHO	125	21.40732	10.93268	0.515	0.515	0.4	0.7	0.3	0.02	0.6	85054.532
GWO	125.6199	21.35129	10.98781	0.515	0.515	0.5	0.68807	0.300151	0.03254	0.62701	84807.111
PSO	125	20.75388	11.17342	0.515	0.515000	0.5	0.61503	0.300000	0.05161	0.60000	81691.202
MVO	125.6002	21.32250	10.97338	0.515	0.515000	0.5	0.68782	0.301348	0.03617	0.61061	84491.266
SCA	125	21.14834	10.96928	0.515	0.515	0.5	0.7	0.3	0.02778	0.62912	83431.117
GSA	125	20.85417	11.14989	0.515	0.517746	0.5	0.61827	0.304068	0.02000	0.624638	82276.941
GA	125	20.77562	11.01247	0.515	0.515000	0.5	0.61397	0.300000	0.05004	0.610001	82773.982
DE	125	20.87123	11.16697	0.515	0.516000	0.5	0.61951	0.301128	0.05024	0.614531	81569.527

Table 24
Statistical results obtained from different algorithms for Rolling element bearing design problem.

Algorithms	Best	Mean	Worst	Std. Dev.	Median
STOA	**85067.983**	**85042.352**	86551.599	1877.09	**85056.095**
SHO	85054.532	85024.858	85853.876	0186.68	85040.241
GWO	84807.111	84791.613	84517.923	**0137.186**	84960.147
PSO	81691.202	50435.017	**32761.546**	13962.150	42287.581
MVO	84491.266	84353.685	84100.834	0392.431	84398.601
SCA	83431.117	81005.232	77992.482	1710.777	81035.109
GSA	82276.941	78002.107	71043.110	3119.904	78398.853
GA	82773.982	81198.753	80687.239	1679.367	8439.728
DE	81569.527	80397.998	79412.779	1756.902	8347.009

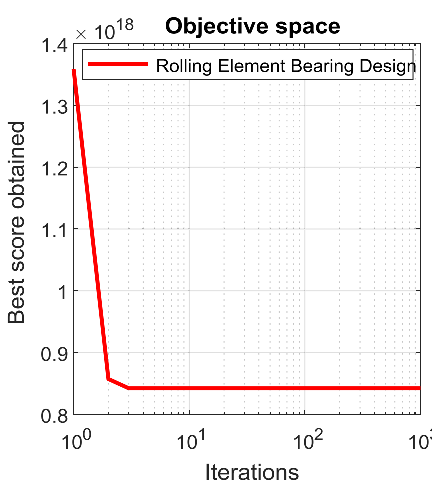

Fig. 17. Convergence analysis of STOA for rolling element bearing design problem.

6. Conclusion

This paper presents a bio-inspired based optimization algorithm called Sooty Tern Optimization Algorithm (STOA) which is tested on forty four benchmark test functions to validate its efficiency. The results reveal that STOA provides very competitive results as compared to nine well-known optimization algorithms. The results on the unimodal and multimodal test functions show the exploration and exploitation capabilities of STOA algorithm, respectively. Whereas, results of the CEC 2005 and CEC 2015 benchmark test functions reveal that STOA is capable of solving challenging and high dimensionality bound constrained real problems. The statistical testing has been performed to demonstrate the statistical significance of proposed STOA over benchmark test functions. In addition, STOA is employed to six constrained industrial applications (i.e., Pressure vessel design, Speed reducer design, Welded beam design, Tension/compression spring design, 25-bar truss design, and rolling element bearing design problems) to verify and demonstrate the performance of STOA on a given search spaces. The binary and multi-objective versions of STOA algorithm can be seen as future contribution.

Appendix. Unimodal, multimodal, and fixed-dimension multimodal benchmark test functions

A.1. Unimodal benchmark test functions

Sphere Model

$$F_1(z) = \sum_{i=1}^{30} z_i^2$$
$$-100 \leq z_i \leq 100, \quad f_{min} = 0, \quad Dim = 30$$

Schwefel's Problem 2.22

$$F_2(z) = \sum_{i=1}^{30} |z_i| + \prod_{i=1}^{30} |z_i|$$
$$-10 \leq z_i \leq 10, \quad f_{min} = 0, \quad Dim = 30$$

Schwefel's Problem 1.2

$$F_3(z) = \sum_{i=1}^{30} \Big(\sum_{j=1}^{i} z_j \Big)^2$$
$$-100 \leq z_i \leq 100, \quad f_{min} = 0, \quad Dim = 30$$

Schwefel's Problem 2.21

$$F_4(z) = max_i\{|z_i|, 1 \leq i \leq 30\}$$
$$-100 \leq z_i \leq 100, \quad f_{min} = 0, \quad Dim = 30$$

Generalized Rosenbrock's Function

$$F_5(z) = \sum_{i=1}^{29} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$$
$$-30 \leq z_i \leq 30, \quad f_{min} = 0, \quad Dim = 30$$

Table 25
Shekel's Foxholes Function F_{14}.

$(a_{ij}, i = 1, 2$ and $j = 1, 2, \dots, 25)$								
$i \backslash j$	1	2	3	4	5	6	...	25
1	−32	−16	0	16	32	−32	...	32
2	−32	−32	−32	−32	−32	−16	...	32

Table 26
Hartman Function F_{19}.

i	$(a_{ij}, j = 1, 2, 3)$			c_i	$(p_{ij}, j = 1, 2, 3)$		
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.038150	0.5743	0.8828

Table 27
Shekel Foxholes Functions F_{21}, F_{22}, F_{23}.

i	$(a_{ij}, j = 1, 2, 3, 4)$				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

Table 28
Hartman Function F_{20}.

i	$(a_{ij}, j = 1, 2, \dots, 6)$						c_i	$(p_{ij}, j = 1, 2, \dots, 6)$					
1	10	3	17	3.5	1.7	8	1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10	17	0.1	8	14	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3	3.5	1.7	10	17	8	3	0.2348	0.1415	0.3522	0.2883	0.3047	0.6650
4	17	8	0.05	10	0.1	14	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

Step Function

$$F_6(z) = \sum_{i=1}^{30} (\lfloor z_i + 0.5 \rfloor)^2$$
$$-100 \le z_i \le 100, \quad f_{min} = 0, \quad Dim = 30$$

Quartic Function

$$F_7(z) = \sum_{i=1}^{30} i z_i^4 + random[0, 1]$$
$$-1.28 \le z_i \le 1.28, \quad f_{min} = 0, \quad Dim = 30$$

A.2. Multimodal benchmark test functions

Generalized Schwefel's Problem 2.26

$$F_8(z) = \sum_{i=1}^{30} -z_i \sin(\sqrt{|z_i|})$$
$$-500 \le z_i \le 500, \quad f_{min} = -12569.5, \quad Dim = 30$$

Generalized Rastrigin's Function

$$F_9(z) = \sum_{i=1}^{30} [z_i^2 - 10\cos(2\pi z_i) + 10]$$
$$-5.12 \le z_i \le 5.12, \quad f_{min} = 0, \quad Dim = 30$$

Ackley's Function

$$F_{10}(z) = -20exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} z_i^2}\right) - exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos(2\pi z_i)\right) + 20 + e$$
$$-32 \le z_i \le 32, \quad f_{min} = 0, \quad Dim = 30$$

Generalized Griewank Function

$$F_{11}(z) = \frac{1}{4000}\sum_{i=1}^{30} z_i^2 - \prod_{i=1}^{30}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$$
$$-600 \le z_i \le 600, \quad f_{min} = 0, \quad Dim = 30$$

Generalized Penalized Functions

- $$F_{12}(z) = \frac{\pi}{30}\{10\sin(\pi x_1) + \sum_{i=1}^{29}(x_i - 1)^2[1 + 10\sin^2(\pi x_{i+1})] + (x_n - 1)^2\} + \sum_{i=1}^{30} u(z_i, 10, 100, 4)$$
  $$-50 \le z_i \le 50, \quad f_{min} = 0, \quad Dim = 30$$
- $$F_{13}(z) = 0.1\{\sin^2(3\pi z_1) + \sum_{i=1}^{29}(z_i - 1)^2[1 + \sin^2(3\pi z_i + 1)] + (z_n - 1)^2[1 + \sin^2(2\pi z_{30})]\} + \sum_{i=1}^{N} u(z_i, 5, 100, 4)$$
  $$-50 \le z_i \le 50, \quad f_{min} = 0, \quad Dim = 30$$

Table 29
Composite benchmark test functions.

Functions	Dim	Range	f_{min}
$F_{24}(CF1)$: $f_1, f_2, f_3, \dots, f_{10}$ = Sphere Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [1, 1, 1, \dots, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10	[-5, 5]	0
$F_{25}(CF2)$: $f_1, f_2, f_3, \dots, f_{10}$ = Griewank's Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [1, 1, 1, \dots, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10	[-5, 5]	0
$F_{26}(CF3)$: $f_1, f_2, f_3, \dots, f_{10}$ = Griewank's Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [1, 1, 1, \dots, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] = [1, 1, 1, \dots, 1]$	10	[-5, 5]	0
$F_{27}(CF4)$: f_1, f_2 = Ackley's Function f_3, f_4 = Rastrigin's Function f_5, f_6 = Weierstrass' Function f_7, f_8 = Griewank's Function f_9, f_{10} = Sphere Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [1, 1, 1, \dots, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	[-5, 5]	0
$F_{28}(CF5)$: f_1, f_2 = Rastrigin's Function f_3, f_4 = Weierstrass' Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley's Function f_9, f_{10} = Sphere Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [1, 1, 1, \dots, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	[-5, 5]	0
$F_{29}(CF6)$: f_1, f_2 = Rastrigin's Function f_3, f_4 = Weierstrass' Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley's Function f_9, f_{10} = Sphere Function $[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\beta_1, \beta_2, \beta_3, \dots, \beta_{10}] =$ $[0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 *$ $5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	10	[-5, 5]	0

where, $x_i = 1 + \dfrac{z_i + 1}{4}$

$$u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m & z_i > a \\ 0 & -a < z_i < a \\ k(-z_i - a)^m & z_i < -a \end{cases}$$

A.3. Fixed-dimension multimodal benchmark test functions

Shekel's Foxholes Function

$$F_{14}(z) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(z_i - a_{ij})^6} \right)^{-1}$$

$$-65.536 \le z_i \le 65.536, \quad f_{min} \approx 1, \quad Dim = 2$$

Kowalik's Function

$$F_{15}(z) = \sum_{i=1}^{11} \left[a_i - \frac{z_1(b_i^2 + b_i z_2)}{b_i^2 + b_i z_3 + z_4} \right]^2$$

$$-5 \le z_i \le 5, \quad f_{min} \approx 0.0003075, \quad Dim = 4$$

Six-Hump Camel-Back Function

$$F_{16}(z) = 4z_1^2 - 2.1z_1^4 + \frac{1}{3}z_1^6 + z_1 z_2 - 4z_2^2 + 4z_2^4$$

$$-5 \le z_i \le 5, \quad f_{min} = -1.0316285, \quad Dim = 2$$

Branin Function

$$F_{17}(z) = \left(z_2 - \frac{5.1}{4\pi^2} z_1^2 + \frac{5}{\pi} z_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos z_1 + 10$$

$$-5 \le z_1 \le 10, \quad 0 \le z_2 \le 15, \quad f_{min} = 0.398, \quad Dim = 2$$

Goldstein–Price Function

$$\begin{aligned} F_{18}(z) = {} & [1 + (z_1 + z_2 + 1)^2(19 - 14z_1 + 3z_1^2 - 14z_2 + 6z_1 z_2 + 3z_2^2)] \\ & \times [30 + (2z_1 - 3z_2)^2 \times (18 - 32z_1 + 12z_1^2 \\ & + 48z_2 - 36z_1 z_2 + 27z_2^2)] \end{aligned}$$

$$-2 \le z_i \le 2, \quad f_{min} = 3, \quad Dim = 2$$

Hartman's Family

- $F_{19}(z) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(z_j - p_{ij})^2)$

  $0 \le z_j \le 1, \quad f_{min} = -3.86, \quad Dim = 3$
- $F_{20}(z) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(z_j - p_{ij})^2)$

  $0 \le z_j \le 1, \quad f_{min} = -3.32, \quad Dim = 6$

Shekel's Foxholes Function

- $F_{21}(z) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$

  $0 \le z_i \le 10, \quad f_{min} = -10.1532, \quad Dim = 4$
- $F_{22}(z) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$

Table 30
CEC 2015 benchmark test functions.

No.	Functions	Related basic functions	Dim	f_{min}
$CEC-1$	Rotated Bent Cigar Function	Bent Cigar Function	30	100
$CEC-2$	Rotated Discus Function	Discus Function	30	200
$CEC-3$	Shifted and Rotated Weierstrass Function	Weierstrass Function	30	300
$CEC-4$	Shifted and Rotated Schwefel's Function	Schwefel's Function	30	400
$CEC-5$	Shifted and Rotated Katsuura Function	Katsuura Function	30	500
$CEC-6$	Shifted and Rotated HappyCat Function	HappyCat Function	30	600
$CEC-7$	Shifted and Rotated HGBat Function	HGBat Function	30	700
$CEC-8$	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Griewank's Function Rosenbrock's Function	30	800
$CEC-9$	Shifted and Rotated Expanded Scaffer's F6 Function	Expanded Scaffer's F6 Function	30	900
$CEC-10$	Hybrid Function 1 ($N = 3$)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	30	1000
$CEC-11$	Hybrid Function 2 ($N = 4$)	Griewank's Function Weierstrass Function Rosenbrock's Function Scaffer's F6 Function	30	1100
$CEC-12$	Hybrid Function 3 ($N = 5$)	Katsuura Function HappyCat Function Expanded Griewank's plus Rosenbrock's Function Schwefel's Function Ackley's Function	30	1200
$CEC-13$	Composition Function 1 ($N = 5$)	Rosenbrock's Function High Conditioned Elliptic Function Bent Cigar Function Discus Function High Conditioned Elliptic Function	30	1300
$CEC-14$	Composition Function 2 ($N = 3$)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	30	1400
$CEC-15$	Composition Function 3 ($N = 5$)	HGBat Function Rastrigin's Function Schwefel's Function Weierstrass Function High Conditioned Elliptic Function	30	1500

$0 \le z_i \le 10, \quad f_{min} = -10.4028, \quad Dim = 4$

- $F_{23}(z) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$

  $0 \le z_i \le 10, \quad f_{min} = -10.536, \quad Dim = 4$

A.4. Basic composite benchmark test functions

Weierstrass Function

$$F(z) = \sum_{i=1}^{30}\Big(\sum_{k=0}^{20}[0.5^k \cos(2\pi 3^k(z_i + 0.5))]\Big) - 30\sum_{k=0}^{20}[0.5^k \cos(2\pi 3^k \times 0.5)]$$

Note that the Sphere, Rastrigin's, Griewank's, and Ackley's functions in composite benchmark suite are same as above mentioned F_1, F_9, F_{11}, and F_{10} benchmark test functions.

A.5. Basic CEC 2015 benchmark test functions

Bent Cigar Function

$$F(z) = z_1^2 + 10^6 \sum_{i=2}^{30} z_i^2$$

Discus Function

$$F(z) = 10^6 z_1^2 + \sum_{i=2}^{30} z_i^2$$

Modified Schwefel's Function

$$F(z) = 418.9829 \times 30 - \sum_{i=1}^{30} g(y_i), \quad y_i = z_i + 4.209687462275036e+002$$

$$g(y_i) = \begin{cases} y_i \sin(|y_i|^{1/2}) \\ \quad where, if\ |y_i| \le 500, \\ (500 - mod(y_i, 500)) \sin(\sqrt{|500 - mod(y_i, 500)|}) - \dfrac{(y_i - 500)^2}{10000 \times 30} \\ \quad where, if\ y_i > 500, \\ (mod(|y_i|, 500) - 500) \sin(\sqrt{|mod(|y_i|, 500) - 500|}) - \dfrac{(y_i + 500)^2}{10000 \times 30} \\ \quad where, if\ y_i < -500 \end{cases}$$

Katsuura Function

$$F(z) = \frac{10}{30^2} \prod_{i=1}^{30}\Big(1 + i\sum_{j=1}^{32} \frac{|2^j z_i - round(2^j z_i)|}{2^j}\Big)^{\frac{10}{30^{1.2}}} - \frac{10}{30^2}$$

Happy Cat Function

$$F(z) = \Big|\sum_{i=1}^{30} z_i^2 - 30\Big|^{1/4} + (0.5\sum_{i=1}^{30} z_i^2 + \sum_{i=1}^{30} z_i)/30 + 0.5$$

HGBat Function

$$F(z) = \Big|(\sum_{i=1}^{30} z_i^2)^2 - (\sum_{i=1}^{30} z_i)^2\Big|^{1/2} + (0.5\sum_{i=1}^{30} z_i^2 + \sum_{i=1}^{30} z_i)/30 + 0.5$$

Expanded Griewank's plus Rosenbrock's Function

$$F(z) = F_{39}(F_{38}(z_1, z_2)) + F_{39}(F_{38}(z_2, z_3)) + \cdots + F_{39}(F_{38}(z_{30}, z_1))$$

Expanded Scaffer's $F6$ Function

Scaffer's $F6$ Function:

$$g(z, x) = 0.5 + \frac{(\sin^2(\sqrt{z^2 + x^2}) - 0.5)}{(1 + 0.001(z^2 + x^2))^2}$$

$$F(z) = g(z_1, z_2) + g(z_2, z_3) + \cdots + g(z_{30}, z_1)$$

High Conditioned Elliptic Function

$$F(z) = \sum_{i=1}^{30} (10^6)^{\frac{i-1}{30-1}} z_i^2$$

Note that the Weierstrass, Rosenbrock's, Griewank's, Rastrigin's, and Ackley's functions in CEC 2015 benchmark test suite are same as above mentioned $Weierstrass, F_5, F_{11}, F_9$, and F_{10} benchmark test functions.

A.6. Composite benchmark functions

The detailed description of six well-known composite benchmark test functions ($F_{24} - F_{29}$) are mentioned in Table 29.

A.7. CEC 2015 benchmark test functions

The detailed description of fifteen well-known CEC 2015 benchmark test functions ($CEC1 - CEC15$) are mentioned in Table 30.

References

Alba, E., Dorronsoro, B., 2005. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans. Evol. Comput. 9 (2), 126–142.

van den Bergh, F., Engelbrecht, A.P., 2006. A study of particle swarm optimization particle trajectories. Inform. Sci. 176 (8), 937–971.

Beyer, Hans-Georg, Schwefel, Hans-Paul, 2002. Evolution strategies – a comprehensive introduction. Nat. Comput. 1 (1), 3–52.

Bonabeau, Eric, Dorigo, Marco, Theraulaz, Guy, 1999. Swarm Intelligence: From Natural To Artificial Systems. Oxford University Press, Inc..

Camp, Charles V., Bichon, Barron J., 2004. Design of space trusses using ant colony optimization. J. Struct. Eng. 130 (5), 741–751.

Chandrawat, Rajesh Kumar, Kumar, Rakesh, Garg, B.P., Dhiman, Gaurav, Kumar, Sumit, 2017. An analysis of modeling and optimization production cost through fuzzy linear programming problem with symmetric and right angle triangular fuzzy number. In: Proceedings of Sixth International Conference on Soft Computing for Problem Solving. Springer, pp. 197–211.

Chen, Q., Liu, B., Zhang, Q., Liang, J., Suganthan, P., Qu, B.Y., 2014. Problem Definitions and Evaluation Criteria for CEC 2015 Special Session on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization. Technical Report, Nanyang Technological University.

Coello, Carlos A. Coello, 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Engrg. 191 (11–12), 1245–1287.

Dhiman, Gaurav, Guo, Sen, Kaur, Satnam, 2018. ED-SHO: a framework for solving non-linear economic load power dispatch problem using spotted hyena optimizer. Modern Phys. Lett. A 33 (40), 1850239.

Dhiman, Gaurav, Kaur, Amandeep, 2017. Spotted hyena optimizer for solving engineering design problems. In: Machine Learning and Data Science (MLDS), 2017 International Conference on. IEEE, pp. 114–119.

Dhiman, Gaurav, Kaur, Amandeep, 2018. Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. Designs 2 (3), 28.

Dhiman, Gaurav, Kaur, Amandeep, 2019. A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization. In: Soft Computing for Problem Solving. Springer, pp. 599–615.

Dhiman, Gaurav, Kumar, Vijay, 2017. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. Adv. Eng. Softw. 114, 48–70.

Dhiman, Gaurav, Kumar, Vijay, 2018. Astrophysics inspired multi-objective approach for automatic clustering and feature selection in real-life environment. Mod. Phys. Lett. B 1850385.

Dhiman, Gaurav, Kumar, Vijay, 2018a. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. Knowl.-Based Syst. 159, 20–50.

Dhiman, Gaurav, Kumar, Vijay, 2018b. KnRVEA: A hybrid evolutionary algorithm based on knee points and reference vector adaptation strategies for many-objective optimization. Appl. Intell. 1–27.

Dhiman, Gaurav, Kumar, Vijay, 2018c. Multi-objective spotted hyena optimizer: A multi-objective optimization algorithm for engineering problems. Knowl.-Based Syst. 150, 175–197.

Dhiman, Gaurav, Kumar, Vijay, 2019a. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowl.-Based Syst. 165, 169–196.

Dhiman, Gaurav, Kumar, Vijay, 2019b. Spotted hyena optimizer for solving complex and non-linear constrained engineering problems. In: Harmony Search and Nature Inspired Optimization Algorithms. Springer, pp. 857–867.

Dhiman, Gaurav, Singh, Pritpal, Kaur, Harsimran, Maini, Ritika, 2019. Dhiman: A novel algorithm for economic dispatch problem based on optimization method using monte carlo simulation and a strophysics concepts. Modern Phys. Lett. A 1950032.

Dorigo, Marco, Birattari, Mauro, Stutzle, Thomas, 2006. Ant colony optimization - artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag. 1, 28–39.

Du, Haifeng, Wu, Xiaodong, Zhuang, Jian, 2006. Small-World Optimization Algorithm for Function Optimization. Springer Berlin Heidelberg, pp. 264–273.

Erol, Osman K., Eksin, Ibrahim, 2006. A new optimization method: Big bang-big crunch. Adv. Eng. Softw. 37 (2), 106–111.

Formato, Richard A., 2009. Central force optimization: A new deterministic gradient-like optimization metaheuristic. Opsearch 46 (1), 25–51.

Gandomi, Amir Hossein, Yang, Xin-She, 2011. Benchmark Problems in Structural Optimization. Springer Berlin Heidelberg, pp. 259–281.

Hatamlou, Abdolreza, 2013. Abdolreza hatamlou black hole: A new heuristic optimization approach for data clustering. Inform. Sci. 222, 175–184.

Holland, John H., 1992. Genetic algorithms. Sci. Am. 267 (1), 66–72.

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

Karaboga, Dervis, Basturk, Bahriye, 2007. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 789–798.

Kaur, Amandeep, Dhiman, Gaurav, 2019. A review on search-based tools and techniques to identify bad code smells in object-oriented systems. In: Harmony Search and Nature Inspired Optimization Algorithms. Springer, pp. 909–921.

Kaur, Amandeep, Kaur, Satnam, Dhiman, Gaurav, 2018. A quantum method for dynamic nonlinear programming technique using schrödinger equation and monte carlo approach. Mod. Phys. Lett. B 1850374.

Kaveh, A., Khayatazad, M., 2012. A new meta-heuristic method: Ray optimization. Comput. Struct. 112–113, 283–294.

Kaveh, A., Talatahari, S., 2009. Size optimization of space trusses using big bang-big crunch algorithm. Comput. Struct. 87 (17–18), 1129–1140.

Kaveh, A., Talatahari, S., 2010a. A novel heuristic optimization method: charged system search. Acta Mech. 213 (3), 267–289.

Kaveh, Ali, Talatahari, Siamak, 2010b. Optimal design of skeletal structures via the charged system search algorithm. Struct. Multidiscip. Optim. 41 (6), 893–911.

Kennedy, James, Eberhart, Russell C., 1995. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. pp. 1942–1948.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Koza, John R., 1992. Genetic Programming: On the Programming of Computers By Means of Natural Selection. MIT Press.

Lozano, M., Garcia-Martinez, C., 2010. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Comput. Oper. Res. 37 (3), 481–497.

Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18 (1), 50–60, 03.

Mezura-Montes, Efrén, Coello, Carlos A. Coello, 2005. Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms. Springer Berlin Heidelberg, pp. 652–662.

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

Mirjalili, Seyedali, 2016. Sca: A sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. 96, 120–133.

Mirjalili, Seyedali, Mirjalili, Seyed Mohammad, Hatamlou, Abdolreza, 2016. Multi-verse optimizer: A nature-inspired algorithm for global optimization. Neural Comput. Appl. 27 (2), 495–513.

Mirjalili, Seyedali, Mirjalili, Seyed Mohammad, Lewis, Andrew, 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.

Rashedi, Esmat, Nezamabadi-pour, Hossein, Saryazdi, Saeid, 2009. GSA: A gravitational search algorithm. Inform. Sci. 179 (13), 2232–2248.

Schutte, J.F., Groenwold, A.A., 2003. Sizing design of truss structures using particle swarms. Struct. Multidiscip. Optim. 25 (4), 261–269.

Simon, D., 2008. Biogeography-based optimization. IEEE Trans. Evol. Comput. 12 (6), 702–713.

Singh, Pritpal, Dhiman, Gaurav, 2017. A fuzzy-LP approach in time series forecasting. In: International Conference on Pattern Recognition and Machine Intelligence. Springer, pp. 243–253.

Singh, Pritpal, Dhiman, Gaurav, 2018a. A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. J. Comput. Sci. 27, 370–385.
Singh, Pritpal, Dhiman, Gaurav, 2018b. Uncertainty representation using fuzzy-entropy approach: Special application in remotely sensed high-resolution satellite images (RSHRSIs). Appl. Soft Comput. 72, 121–139.
Singh, Pritpal, Dhiman, Gaurav, Kaur, Amandeep, 2018a. A quantum approach for time series data based on graph and schrodinger equations methods. Modern Phys. Lett. A 33 (35), 1850208.
Singh, Pritpal, Rabadiya, Kinjal, Dhiman, Gaurav, 2018b. A four-way decision-making system for the indian summer monsoon rainfall. Mod. Phys. Lett. B 32 (25), 1850304.
Slowik, A., Kwasnicka, H., 2017. Nature inspired methods and their industry applications - swarm intelligence algorithms. IEEE Trans. Ind. Inf. (99), 1.
Spears, William M., Green, Derek T., Spears, Diana F., 2012. Biases in particle swarm optimization. In: Innovations and Developments of Swarm Intelligence Applications. IGI Global, pp. 20–43.
Storn, Rainer, Price, Kenneth, 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.
Tamura, Kenichi, Yasuda, Keiichiro, 2017. The spiral optimization algorithm: Convergence conditions and settings. IEEE Trans. Syst. Man Cybern.: Syst. (99), 1–16.
Watson, John B., 1908. Viii. the behavior of noddy and sooty terns. Papers 2, 185.
Wilcoxon, Frank, 1945. Frank wilcoxon individual comparisons by ranking methods. Biom. Bull. 1 (6), 80–83.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.
Yang, Xin-She, 2010. A New Metaheuristic Bat-Inspired Algorithm. Springer Berlin Heidelberg, pp. 65–74.
Yang, X.S., Deb, Suash, 2009. Cuckoo search via levy flights. In: World Congress on Nature Biologically Inspired Computing. pp. 210–214.