Knowledge-Based Systems 257 (2022) 109484

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Special Relativity Search: A novel metaheuristic method based on special relativity physics

Vahid Goodarzimehr [a], Saeed Shojaee [a,*], Saleh Hamzehei-Javaran [a], Siamak Talatahari [b]

[a] Department of Civil Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
[b] Faculty of Engineering & IT, University of Technology Sydney, Ultimo, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 1 February 2022
Received in revised form 28 April 2022
Accepted 15 July 2022
Available online 22 July 2022

Keywords:
Special Relativity Search
Metaheuristic
Optimization
Magnetic field
Global optimal

ABSTRACT

In this work, a novel metaheuristic optimization algorithm called Special Relativity Search (SRS) is proposed. The SRS is inspired by the interaction of particles in an electromagnetic field. Particle interactions are calculated using the Lorentz force, and the equation of motion is developed using angular frequency. The magnetic force between particles is perpendicular to the velocity of charged particles and the magnetic field, which causes particles to move in a circular trajectory. In this method, for the first time, the theory of special relativity physics is utilized to determine the coordinates of charged particles in each rotation. The SRS main step equation is developed using the two phenomena of length contraction and time dilation. Charged particles are members of the initial population that is randomly generated, and their charge is determined based on their fitness. To show the efficiency and robustness of the SRS in solving optimization problems, 83 benchmark functions, which are a wide range of mathematical problems, are selected and optimized based on the values of Best, Mean, Median, and Standard Deviation (SD). The statistical test of the Wilcoxon Signed Ranks (WSR) is performed to fairly compare the results of this new method with other popular metaheuristic algorithms. The test results show that in most cases SRS is superior to other methods from the state-of-the-art. The results of evaluated optimization problems show that SRS is more effective and efficient compared to some other well-known metaheuristic methods in solving optimization problems.

1. Introduction

In recent years, the development of a new optimization method has become a major challenge for researchers in all fields of science. Therefore, developing a method that can solve optimization problems in a reasonable time and with high accuracy is unavoidable. Generally, optimization methods are divided into two categories; deterministic and stochastic [1]. As optimization problems become more complex, deterministic methods do not work efficiently, and a robust strategy is needed. Despite the difficulty, such as the need for derivation info and continuity, the possibility of convergence to local optimal points increases the computational time [2]. Stochastic methods are more popular than deterministic ones for four reasons: simplicity, flexibility, derivation free mechanism, and avoidance of trapping in the local optimal point [3]. These methods do not require any problem-derived information, and they can escape from the local optimal point and discover the global optimal point with their special operators. Also, the required computational time increases in a linear or polynomial manner with increasing the dimensions of the problem [4]. A metaheuristic method is an efficient tool that provides an acceptable solution to complex optimization problems through a smart trial and error process [5]. Due to the complexity of the problem, they offer a solution with a possible combination to achieve the best answer. Although there is no guarantee of obtaining an optimal solution, metaheuristic methods are so efficient, and in most cases, they provide optimal quality answers. Another class is based on the hybridization of deterministic and stochastic. The hybrid method develops according to the complexity of the problem, and the main idea is to eliminate the weaknesses of deterministic method. For example, a hill-climbing starting from a random point is a good example. The main idea is to use a deterministic algorithm but start with random starting points. It has advantages over the simple hill-climbing method, which will be trapped at a local peak. However, there is a random component in these hybrid algorithms, which is often classified as a stochastic algorithm in the optimization literature [6]. This paper deals with metaheuristics that represent a more general approximation mechanism, which can be applied to solve different optimization problems. Metaheuristics solve optimization problems by searching for the optimal answer in the feasible space.

* Corresponding author.
E-mail address: saeed.shojaee@uk.ac.ir (S. Shojaee).

https://doi.org/10.1016/j.knosys.2022.109484
0950-7051/

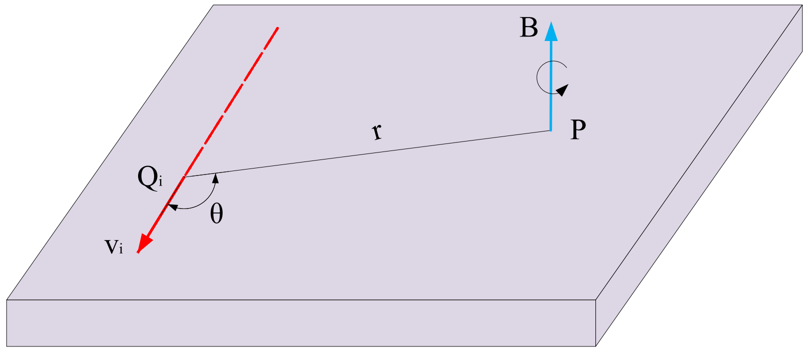

Fig. 1. Charged particle Q_i moving with initial velocity v_i and produces magnetice field B.

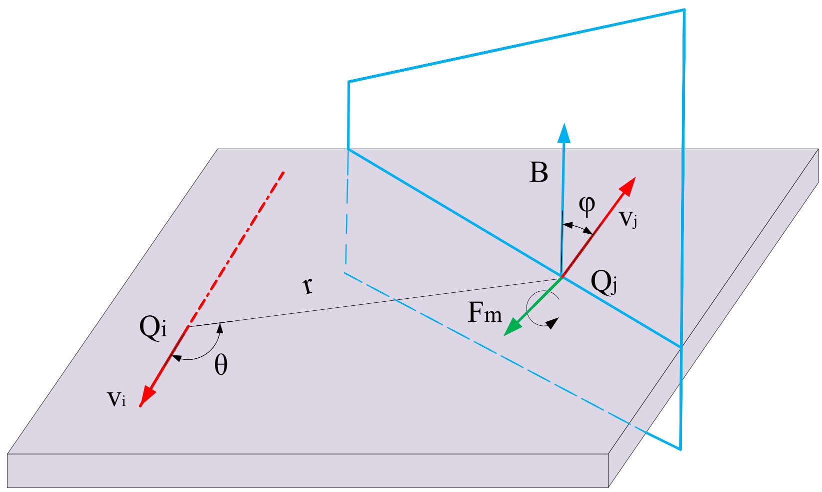

Fig. 2. Exerted magnetice force F_m on particle Q_j.

The main components of any metaheuristic algorithm are intensification and diversification or exploitation and exploration [7]. Exploitation means that at every stage of the search process, the information related to an optimal solution needs to ensure that there is no better answer in the neighborhood. Exploration means producing diverse solutions for the search space on a global scale. Creating diversity through randomization increases many solutions while avoiding being trapped in the local optimal point. The right combination of these two components ensures the best optimal solution. Unexplored areas, should be investigated to ensure that all areas of the search space are uniformly explored and the search process unlimited to smaller spaces. To fulfill this aim, stochastic search algorithms generate random solutions in each iteration. But, local search methods investigate the areas near the best solution to improve the current solution [8]. Researchers have developed many metaheuristic algorithms to solve optimization problems. These problems are a wide range of different mathematical problems designed to evaluate the performance of the metaheuristics [9,10] and [11]. In some cases, they are compatible with some metaheuristic algorithms, and the algorithm can solve the problem. However, these problems are incompatible with some metaheuristic algorithms, and they will not get better results.

Based on the No Free Lunch (NFL) theory, no metaheuristic algorithm succeeds in solving all optimization problems. In other words, a particular metaheuristic algorithm may provide acceptable results for some problems, but the same algorithm may show poor performance for other problems [12]. Therefore, despite the development of many metaheuristic algorithms, researchers are still trying to provide a new method that can outperform others methods. So, the purpose of developing metaheuristic algorithms is not only to achieve a better optimal answer but also to solve a wide range of optimization problems. Therefore, the development of a new method that can solve a large number of problems is welcomed. Many optimization models presented by metaheuristic algorithms are inspired by nature. Thus, finding a natural phenomenon that can provide a successful optimization model is still one of the most significant factors in developing metaheuristic algorithms. In general, natural phenomena that inspire metaheuristics to provide an optimization algorithm are divided into several categories: (a) Evolutionary-based Algorithms (EAs), (b) Physics/Chemistry-based (P/C) algorithms, (c) Swarm Intelligence-based (SI), (d) Human-Based algorithms (HB) [13]. Each of these categories has its subcategories; In the following, we will briefly review some of them:

Evolutionary-based Algorithms (EAs): inspired by Darwin's theory, evolutionary algorithms have been able to provide a model for solving optimization problems [14]. Evolution is based on adaptability. For example, we need to improve our abilities to adapt to our environment. For this reason, it is said that, evolution is an optimization process that aims to improve the capabilities of a system to a better presentation of itself in a dynamic environment. Some of the most popular methods in this field are: Genetic Algorithm (GA) [15], Differential Evolution (DE) [16], Evolution Strategy (ES) [17], Biogeography-Based Optimization (BBO) [18], Estimation of Distribution Algorithms (EDA) [19], Artificial Immune Systems (AIS) [20], and Memetic Algorithms (MA) [21], Wildebeests Herd Optimization (WHO) [22], Tree Growth Algorithm (TGA) [23], Kidney-Inspired Algorithm

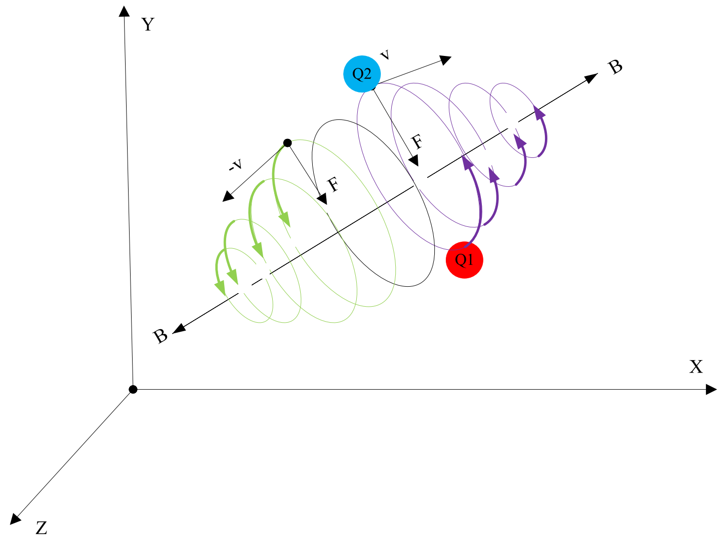

Fig. 3. Circular motion of moving particles in a magnetic field.

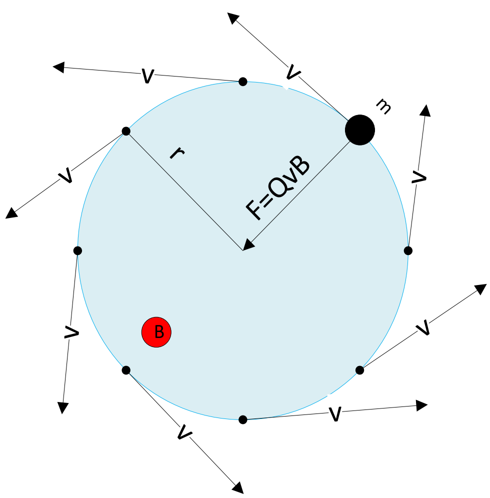

Fig. 4. Circular rotation of a particle with mass m in a circular path.

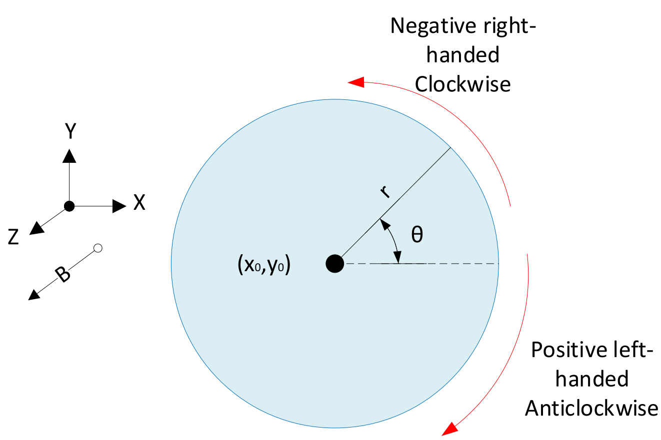

Fig. 5. Cyclotron orbit and radius (x0, y0).

(KIA) [24], Chemotherapy Science (CS) [25], and Multivariable Grey Prediction Evolution Algorithm (MGPEA) [26].

Physics/Chemistry-based (P/C) algorithms: Physics/Chemistry-based algorithms are a kind of metaheuristic algorithms that provide an optimization model using physical laws between particles and phenomena. This type of algorithm is widely welcomed by researchers due to its compatibility with optimization problems. Examples of the most important of these algorithms include: Simulated Annealing (SA) [27], Gravitational Search Algorithm (GSA) [28], Henry Gas Solubility Optimization (HGSO) [29], Charged System Search (CSS) [30], Big-Bang Big-Crunch (BB-BC) [31], Chemical Reaction Optimization (CRO) [32], Ray Optimization (RO) [33], Black Hole (BH) [34], Multi-Verse Optimizer (MVO) [35], Ions Motion Algorithm (IMA) [36], Vertx Search (VS) [37], Atomic Orbital Search Algorithm (AOSA) [38], Crystal Structure Algorithm (CSA) [39], Chaos Game Optimization (CGO) [40], and Gradient-based optimizer (GBO) [41], Newton Metaheuristic Algorithm (NMA) [42], Arithmetic Optimization Algorithm (AOA) [43], Lichtenberg Algorithm (LA) [44], and Sine Cosine Algorithm (SCA) [45].

Swarm Intelligence-based (SI) algorithms: in algorithms based on swarm intelligence, there is a direct and indirect relationship between the answers of the algorithm. In swarm intelligence methods, optimal answers are optimum candidates which need to communicate with each other to find the optimal answer. Swarm intelligence algorithms are inspired by the behavior of animals, such as: Particle Swarm Optimization (PSO) [46], Artificial Bee Colony (ABC) [47], Ant Colony Optimization (ACO) [48], Firefly Algorithm (FA) [49], Cuckoo Search (CS) [50], Krill Herd (KH) [51], Grasshopper Optimisation Algorithm (GOA) [52], Gray Wolf Optimizer (GWO) [53], Bat Algorithm (BA) [54], Whale Optimization Algorithm (WOA) [55], Fruit Fly Optimization (FFA) Algorithm [56], Crow Search Algorithm (CSA) [57], Artificial Algae Algorithm (AAA) [58], Slap Swarm Optimization (ASO) [59], Emperor Penguin Optimizer (EPO) [60], Squirrel Search Algorithm (SSA) [61], White Shark Optimizer (WSO) [62], African Vultures Optimization Algorithm (AVOA) [63], Honey Badger Algorithm (HBA) [64], Lévy flight distribution (LFD) [65], Aquila Optimizer (AO) [66], Chameleon Swarm Algorithm (CSA) [67],

Table 1
The SRS time complexity.

Dimension	T_0	T_1	$\overline{T_2}$	$\frac{\overline{T_2}-T_1}{T_0}$
10		34.9897	36.0636	6.7796
30	0.1584	41.7717	47.6474	37.0940
50		56.7941	59.8526	19.3087

Table 2
Information of the CEC2005 test functions.

Type	Function	Initialization range	Search bound	Dimension
Unimodal	**F1**: Shifted Sphere Function	[−100,100]	[−100,100]	30
	F2: Shifted Schwefel's Problem 1.2	[−100,100]	[−100,100]	30
	F3: Shifted Rotated High Conditioned Elliptic Function	[−100,100]	[−100,100]	30
	F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness	[−100,100]	[−100,100]	30
	F5: Schwefel's Problem 2.6 with Global Optimum on Bounds	[−100,100]	[−100,100]	30
Basic multimodal	**F6**: Shifted Rosenbrock's Function	[−100,100]	[−100,100]	30
	F7: Shifted Rotated Griewank's Function without Bounds	[0,600]	[−600,600]	30
	F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	[−32,32]	[−32,32]	30
	F9: Shifted Rastrigin's Function	[−5,5]	[−5,5]	30
	F10: Shifted Rotated Rastrigin's Function	[−5,5]	[−5,5]	30
	F11: Shifted Rotated Weierstrass Function	[−0.5,0.5]	[−0.5,0.5]	30
	F12: Schwefel's Problem 2.13	[−100,100]	[−100,100]	30
Expanded multimodal	**F13**: Expanded Extended Griewank's plus Rosenbrock's Function (CEC8CEC2)	[−3,1]	[−3,1]	30
	F14: Shifted Rotated Expanded Schaffer's CEC6	[−100,100]	[−100,100]	2
Hybrid composition multimodal	**F15**: Hybrid Composition Function	[−5,5]	[−5,5]	4
	F16: Rotated Hybrid Composition Function	[−5,5]	[−5,5]	2
	F17: Rotated Hybrid Composition Function with Noise in Fitness	[−5,5]	[−5,5]	2
	F18: Rotated Hybrid Composition Function	[−5,5]	[−5,5]	2
	F19: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	[−5,5]	[−5,5]	3
	F20: Rotated Hybrid Composition Function with the Global Optimum on the Bounds	[−5,5]	[−5,5]	6
	F21: Rotated Hybrid Composition Function	[−5,5]	[−5,5]	4
	F22: Rotated Hybrid Composition Function with High Condition Number Matrix	[−5,5]	[−5,5]	4
	F23: Non-Continuous Rotated Hybrid Composition Function	[−5,5]	[−5,5]	4

Table 3
Results of Unimodal test functions.

Fun		GA	PSO	GSA	TLBO	GOA	SCA	GWO	MPA	RSA	AOA	AO	**SRS**
F_1	Best	0.7055	0.0000	0.0000	0.0000	0.7499	0.0006	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	1.2661	0.0002	0.0000	0.0000	3.4298	4.5613	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000
	Median	1.2305	0.0000	0.0000	0.0000	3.1620	0.7123	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	2.0955	0.0017	0.0000	0.0000	9.5710	51.6348	0.0000	0.0000	0.0099	0.0000	0.0000	0.0000
	SD	0.3146	0.0005	0.0000	0.0000	1.7259	10.9361	0.0000	0.0000	0.0018	0.0000	0.0000	0.0000
F_2	Best	0.1975	0.0174	0.0000	0.0000	0.2339	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	0.3185	28.4296	0.0000	0.0000	5.0726	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Median	0.3201	30.0010	0.0000	0.0000	4.1077	0.0080	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	0.4544	70.2805	0.0000	0.0000	18.882	0.0217	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	SD	0.0625	16.4240	0.0000	0.0000	3.6940	0.0061	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F_3	Best	1746.001	10 948.64	300.9936	0.0000	642.135	609.561	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	4084.474	29 089.94	510.4635	0.0000	2166.472	7014.34	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000
	Median	3761.977	30 574.31	435.6213	0.0000	1827.364	5733.96	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	6559.695	48 642.88	992.5921	0.0000	7260.670	20 654.64	0.0000	0.0000	0.0000	0.0233	0.0000	0.0000
	SD	1289.396	8186.67	176.1126	0.0000	1503.522	5500.36	0.0000	0.0000	0.0000	0.0006	0.0000	0.0000
F_4	Best	1.3210	0.0000	1.1910	0.0000	3.1914	9.2477	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	2.2717	0.0005	3.1984	0.0000	8.3863	25.9075	0.0000	0.0000	0.0000	0.0274	0.0000	0.0000
	Median	2.2671	0.0001	3.0377	0.0000	8.1238	26.1810	0.0000	0.0000	0.0000	0.0224	0.0000	0.0000
	Worst	3.6279	0.0040	6.6885	0.0000	13.992	43.8575	0.0000	0.0000	0.0000	0.0438	0.0000	0.0000
	SD	0.5544	0.0008	1.3998	0.0000	3.1190	9.3782	0.0000	0.0000	0.0000	0.0197	0.0000	0.0000
F_5	Best	266.3652	25.5947	26.3088	21.8927	222.1	350.2	28.7801	23.7360	0.0000	28.0789	0.0002	25.4339
	Mean	730.2725	41.5368	406.1650	23.3968	27 488.8	2 285 262.0	28.9138	24.5124	16.3657	28.6416	0.0068	26.5503
	Median	596.5280	41.9533	215.5538	23.5471	68 215.8	78 815.8	28.9390	24.4335	28.7615	28.7412	0.0021	26.2340
	Worst	2226.672	58.1886	2579.527	24.7573	196 441.7	3.10e+07	28.9967	25.6733	28.9905	28.9562	0.0525	28.5298
	SD	458.7533	8.1190	571.9077	0.6601	493 035.5	6.61e+06	0.0588	0.4215	14.5569	0.2427	0.0111	0.6395

Capuchin Search Algorithm (CapSA) [68], Sparrow Search Algorithm (SSA) [69], Spotted Hyena Optimizer (SHO) [70], Sail-Fish Optimizer (SFO) [71], Ebola Optimization Search Algorithm (EOSA) [72], Dwarf Mongoose Optimization (DMO) [73], Marine Predators Algorithm (MPA) [74], Reptile Search Algorithm (RSA) [75] as many others.

Human-Based (HB) algorithms: human-based algorithms are another type of metaheuristic algorithm that solve optimization problems by providing a mathematical model for social phenomena. For example, the Teaching–Learning Based Optimization (TLBO) algorithm [76], inspired by the teaching–learning process in a classroom, provides a model for solving optimization problems. Examples of this algorithm include: Mine Blast Algorithm (MBA) [77], Social Engineering Optimizer (SEO) [78], Human Behavior-Based Optimization (HBBO) [79], Volleyball Premier League (VPL) [80], Exchange Market Algorithm (EMA) [81], Taboo Search (TS) [82], Imperialist Competitive Algorithm (ICA) [83], Dynastic Optimization Algorithm (DOA) [84], Ali Baba and the Forty Thieves (AFT) [85], Coronavirus Herd Immunity Optimizer (CHIO) [86], Gaining Sharing Knowledge based Algorithm (GSK) [87], Human Mental Search (HMS) [88], Human Urbanization Algorithm (HUA) [89], Social Network Search (SNS) [90], Cooperation Search Algorithm (CSA) [91], Group Teaching Optimization Algorithm (GTOA) [92] .

Metaheuristic algorithms have their own specific capabilities and characteristics. Each algorithm has many advantages and disadvantages, and these methods have strengths and weaknesses in the face of solving various optimization problems. The idea of

Table 4
Wilcoxon Signed Ranks Test.

		SRS–GA	SRS–PSO	SRS–GSA	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–AOA	SRS–AO	SRS–RSA
F_1	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−2.366[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.018
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_2	Z	−4.782[a]	−4.783[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	0.000[c]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS = AOA	SRS < AO	SRS < RSA
F_3	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.000	.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_4	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−3.621[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.000	.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_5	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[b]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[b]	−4.782[a]	−4.782[b]	−1.635[b]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.102
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS** > **TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS** > **MPA**	SRS < AOA	**SRS** > **AO**	**SRS** > **RSA**

[a]Based on positive ranks. (SRS is winner).
[b]Based on negative ranks. (SRS is failure).
[c]The sum of ranks is equal.

Table 5
Results of Basic Multimodal test functions.

Fun		GA	PSO	GSA	TLBO	GOA	SCA	GWO	MPA	RSA	AOA	AO	SRS
F_6	Best	0.5831	0.0000	0.0000	0.0000	0.0000	3.9571	0.0000	5.1514	0.0000	2.4228	0.0000	0.0000
	Mean	1.2330	990.0253	0.2666	0.0000	0.0000	8.2489	0.4832	97.5575	0.0000	2.8963	0.0000	0.0000
	Median	1.2083	0.0000	0.0000	0.0000	0.0000	5.5744	0.5016	6.5411	0.0000	2.9148	0.0000	0.0000
	Worst	2.0468	9900.250	2.0000	0.0000	0.0000	33.9162	1.0102	2739.248	0.0000	3.3689	0.0002	0.0000
	SD	0.3508	3020.849	0.5208	0.0000	0.0000	6.9819	0.2335	498.9359	0.9147	0.2556	0.0000	0.0000
F_7	Best	2110.162	12.2529	1.91e+09	0.0004	0.0158	3148.664	0.0005	0.0001	0.0000	1.43e+16	0.0000	0.0000
	Mean	10 968.19	5.61e+10	4.98e+09	0.0009	0.1046	1.66e+08	0.0018	0.0010	0.0001	3.97e+16	0.0000	0.0000
	Median	9494.89	5.51e+03	4.58e+09	0.0009	0.0907	9.46e+06	0.0014	0.0010	0.0001	4.16e+16	0.0000	0.0000
	Worst	26 272 461	5.18e+11	1.09e+10	0.0010	0.2878	3.42e+09	0.0051	0.0022	0.0003	6.50e+16	0.0002	0.0002
	SD	5821.06	1.16e+11	2.36e+09	0.0030	0.0692	6.31e+08	0.0010	0.0005	0.0000	1.46e+16	0.0000	0.0000
F_8	Best	−581.5036	−600.0064	−409.4639	−702.3511	−240.8296	−526.1717	−722.4774	−722.4888	−540.3788	−334.7942	−675.8682	−722.4888
	Mean	−493.9803	−519.8518	−255.7350	−649.7367	−231.5243	−463.3456	−235.9229	−629.0986	−438.6517	−236.0310	−534.7667	−721.8175
	Median	−480.8187	−524.9758	−259.3224	−662.0758	−240.8296	−462.5429	−240.8285	−631.3192	−447.2215	−232.4297	−534.7667	−722.4888
	Worst	−380.1249	−433.9693	−158.6344	−581.5252	−117.4384	−398.4212	−216.7090	−549.2913	−282.9983	−150.0611	−416.9614	−702.3511
	SD	44.8652	42.5732	52.0459	32.2362	24.8952	30.6274	9.0735	46.9121	67.7683	43.0289	54.9212	3.6766
F_9	Best	3.4439	93.4153	10.9445	0.0000	5.9697	0.0172	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	11.1611	140.4428	18.4067	11.8229	19.7333	27.6424	2.0395	0.0000	0.0000	0.0000	0.0000	0.0000
	Median	10.9724	129.3958	17.9092	11.9945	17.4117	16.1374	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	17.3786	241.2685	31.8386	19.8995	35.8184	114.250	18.5895	0.0000	0.0000	0.0000	0.0000	0.0000
	SD	2.4815	36.2102	5.0242	3.7487	8.5232	33.1806	4.6338	0.0000	0.0000	0.0000	0.0000	0.0000
F_{10}	Best	0.0306	0.0001	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	0.0414	1.5217	0.0000	0.0000	0.0993	0.0177	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Median	0.0405	1.3448	0.0000	0.0000	0.0000	0.0106	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	0.0595	5.05811	0.0000	0.0000	3.4041	0.1502	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	SD	0.0068	1.5423	0.0000	0.0000	1.1492	0.0276	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F_{11}	Best	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	0.0000	0.0056	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Median	0.0000	0.0046	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Worst	0.0000	0.0222	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	SD	0.0000	0.0057	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F_{12}	Best	0.0087	0.9367	2.0196	0.0000	0.0002	0.8159	0.0131	0.0026	0.5092	0.1566	0.0000	0.0000
	Mean	0.0148	9257.363	4.8823	0.0000	0.7576	2.20e+05	0.0328	0.0202	0.0000	0.2600	0.0000	0.0000
	Median	0.0121	11.3534	4.8369	0.0000	0.4098	12.9826	0.0290	0.0171	0.0000	0.2593	0.0000	0.0000
	Worst	0.0337	2.77e+05	9.1955	0.0000	3.1887	5.04e+06	0.0802	0.0496	0.0000	0.3547	0.0000	0.0000
	SD	0.0061	5.06e+04	1.7208	0.0000	0.9725	9.33e+05	0.0178	0.0124	0.3823	0.0442	0.0000	0.0000

Table 6
Wilcoxon Signed Ranks Test.

		SRS–GA	SRS–PSO	SRS–GSA	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–AOA	SRS–AO	SRS–RSA
F_6	Z	−4.782[a]	−4.741[a]	−.895[a]	−4.103[b]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.371	.000	.000	.000	.000	.000	.000	.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS** > **TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_7	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−0.936[b]	−1.676[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.349	.094
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	**SRS** > **AO**	SRS < RSA
F_8	Z	−4.782[a]	−4.785[a]	−4.785[a]	−4.813[a]	−4.782[a]	−4.782[a]	−4.782[a]	−2.828[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005	.000	.000	0.000
	Comparison	SRS > GA	SRS > PSO	SRS > GSA	SRS > TLBO	SRS > GOA	SRS > SCA	SRS > GWO	SRS > MPA	SRS < AOA	SRS < AO	SRS > RSA
F_9	Z	−4.782[a]	−4.782[a]	−4.785[a]	−4.704[a]	−4.782[a]	−4.782[a]	−4.254[a]	.000[c]	.000[c]	.000[c]	.000[c]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000	1.000	1.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS** = **MPA**	**SRS** = **AOA**	**SRS** = **AO**	**SRS** = **RSA**
F_{10}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.932[a]	−4.782[a]	−4.782[a]	−4.901[a]	−4.783[a]	.000[c]	.000[c]	−1.000[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	1.000	1.000	.317
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS** = **AOA**	**SRS** = **AO**	SRS < RSA
F_{11}	Z	−4.782[a]	−4.783[a]	.000[c]	.000[c]	−4.782[a]	−4.782[a]	.000[c]	.000[c]	.000[c]	.000[c]	.000[c]
	Sig. (2-tailed)	.000	.000	1.000	1.000	.000	.000	1.000	1.000	1.000	1.000	1.000
	Comparison	SRS < GA	SRS < PSO	**SRS** = **GSA**	**SRS** = **TLBO**	SRS < GOA	SRS < ScA	**SRS** = **GWO**	**SRS** = **MPA**	**SRS** = **AOA**	**SRS** = **AO**	**SRS** = **RSA**
F_{12}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.720[b]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.000	.000	0.000
	Comparison	SRS > GA	SRS > PSO	SRS > GSA	**SRS** > **TLBO**	SRS < GOA	SRS > SCA	SRS > GWO	SRS > MPA	SRS < AOA	**SRS** < **AO**	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

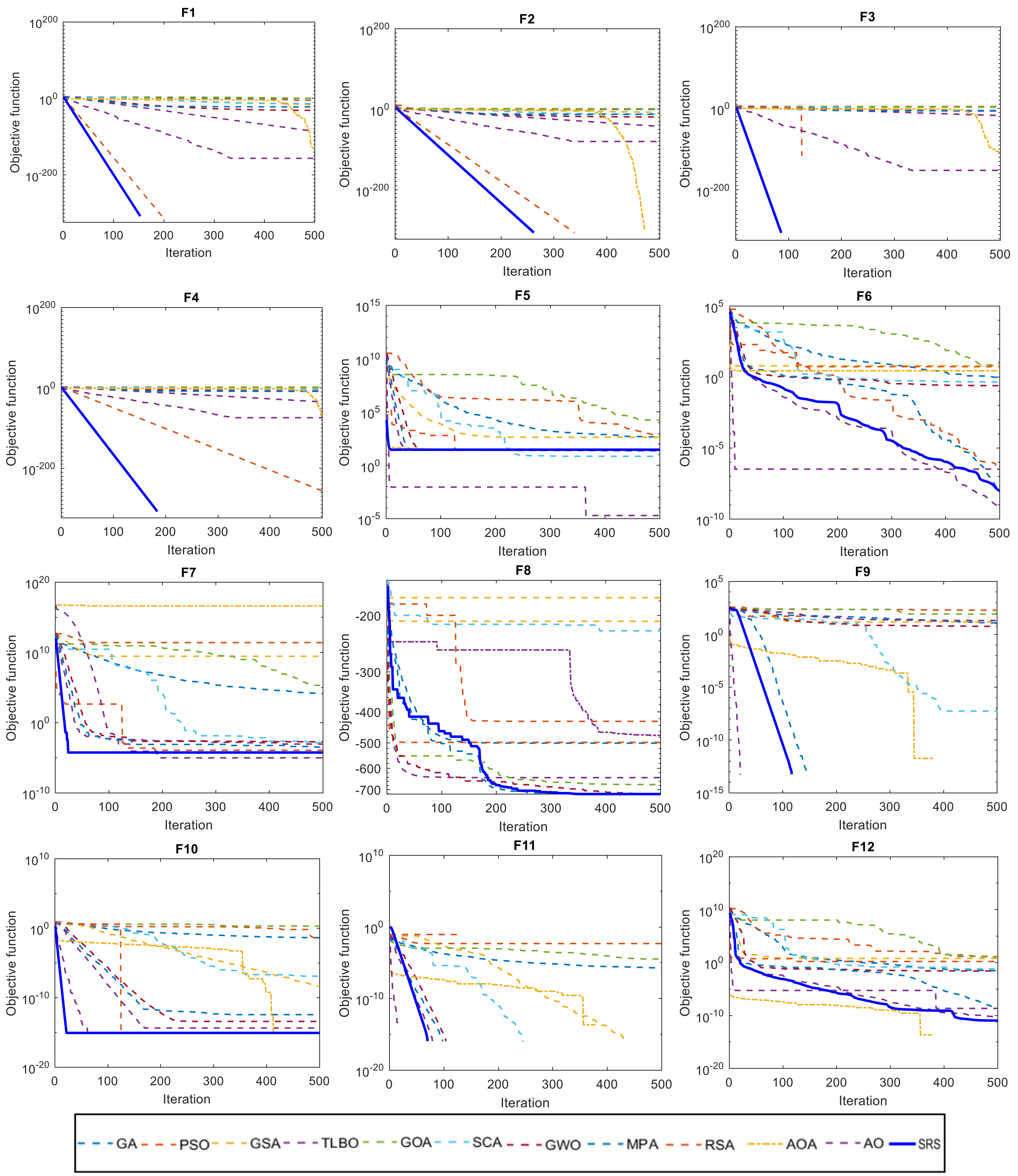

Fig. 6. Convergence history of functions 1 through 12.

hybridization has been formed due to the functional and structural differences of different optimization algorithms. Its purpose is to create hybrid algorithms that can use the strengths of hybrid algorithms to perform better in solving various optimization problems. The Hybrid Evolutionary Algorithm (HEA) [93] is one of the most recent hybrid algorithms which integrates a crossover operator based on solution reconstruction, perturbation-based K-nearest neighbors and densest imputation for collaborative filtering (KDI-KNN) [94]. Surrogate-Assisted Hybrid Optimization (SAHO) [95] algorithm, which is a combination of TLBO [76] and DE [16], Hybrid Cuckoo Search [50] and adaptive Gaussian quantum behaved particle swarm optimization (AGQPSO) [96]. Chaotic maps have been embedded into Gravitational Search Algorithms

Table 7
Results of Expanded Multimodal test functions.

Fun		GA	PSO	GSA	TLBO	GOA	SCA	GWO	MPA	RSA	AOA	AO	SRS
F_{13}	Best	0.0000	0.0000	0.5424	0.0000	0.0000	0.8284	0.0000	0.0000	0.0000	0.0820	0.0000	0.0000
	Mean	0.0000	0.8000	0.8274	0.0061	0.0000	1.1281	0.1162	0.1064	0.0000	1.4401	0.0002	0.0000
	Median	0.0000	0.0000	0.8439	0.0000	0.0000	1.1540	0.0988	0.0000	0.0000	1.4659	0.0000	0.0000
	Worst	0.0007	3.2000	1.0325	0.1183	0.0000	1.5145	0.5033	1.8511	0.0000	1.8022	0.0018	0.0000
	SD	0.0001	1.0916	0.1368	0.0218	0.0000	0.1921	0.1402	0.3123	0.0000	0.2228	0.0004	0.0000
F_{14}	Best	0.9980	0.9980	1.0006	0.9980	0.9980	0.9980	0.9980	0.9980	0.9993	0.9980	0.9980	0.9980
	Mean	0.9980	0.9980	4.2550	0.9980	0.9980	1.4614	2.9613	0.9980	3.2561	8.9954	2.9209	0.9980
	Median	0.9980	0.9980	3.3587	0.9980	0.9980	0.9981	0.9980	0.9980	2.9821	10.7631	0.9980	0.9980
	Worst	0.9980	0.9980	11.4794	0.9980	0.9980	2.9821	10.7631	0.9980	10.7631	12.6705	12.6705	0.9980
	SD	0.0000	0.0000	2.4831	0.0000	0.0000	0.8532	3.2438	0.0000	2.2179	3.5207	3.7776	0.0000

Table 8
Wilcoxon Signed Ranks Test.

		SRS–GA	SRS–PSO	SRS–GSA	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–AOA	SRS–AO	SRS–RSA
F_{13}	Z	−4.782[a]	−3.217[a]	−4.782[a]	−4.704[a]	−4.703[a]	−4.782[a]	−4.782[a]	−4.015[a]	−4.782[a]	−4.782[a]	−4.784[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	.000	.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{14}	Z	−6.162[a]	−2.658[a]	−21.436[a]	−6.183[a]	−17.545[a]	−21.435[a]	−21.434[a]	−9.878[a]	−21.435[a]	−16.581[a]	−10.576[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).

Table 9
Results of Hybrid Composition Multimodal test functions.

Fun		GA	PSO	GSA	TLBO	GOA	SCA	GWO	MPA	RSA	AOA	AO	SRS
F_{15}	Best	0.0005	0.0003	0.0012	0.0003	0.0006	0.0005	0.0003	0.0003	0.0006	0.0003	0.0003	0.0004
	Mean	0.0023	0.0048	0.0041	0.0004	0.0075	0.0040	0.0044	0.0003	0.0011	0.0095	0.0004	0.0039
	Median	0.0014	0.0007	0.0034	0.0003	0.0008	0.0041	0.0003	0.0003	0.0011	0.0028	0.0004	0.0031
	Worst	0.0204	0.0203	0.0128	0.0012	0.0626	0.0015	0.0203	0.0003	0.0020	0.0910	0.0016	0.0135
	SD	0.0037	0.0079	0.0028	0.0002	0.0133	0.0003	0.0080	0.0000	0.0003	0.0178	0.0002	0.0032
F_{16}	Best	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Mean	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0310	−1.0316	−1.0313	−1.0316
	Median	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0313	−1.0316	−1.0314	−1.0316
	Worst	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0315	−1.0316	−1.0316	−1.0284	−1.0316	−1.0303	−1.0316
	SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	0.0000	0.0002	0.0000
F_{17}	Best	0.3978	0.3978	0.3978	0.3978	0.3978	0.3978	0.3978	0.3978	0.3979	0.3978	0.3978	0.3978
	Mean	0.3978	0.3978	0.3978	0.3978	0.3978	0.3988	0.3978	0.3978	0.4062	0.3978	0.3979	0.3978
	Median	0.3978	0.3978	0.3978	0.3978	0.3978	0.3985	0.3978	0.3978	0.4040	0.3978	0.3979	0.3978
	Worst	0.3978	0.3978	0.3978	0.3978	0.3978	0.4016	0.3978	0.3978	0.4373	0.3978	0.3984	0.3978
	SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0009	0.0000	0.0000	0.0084	0.0000	0.0001	0.0000
F_{18}	Best	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0005	3.0000
	Mean	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	4.9043	27.2380	3.0180	3.0000
	Median	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	30.0000	3.0154	3.0000
	Worst	3.0000	3.0000	3.0000	3.0000	3.0000	3.0003	3.0000	3.0000	33.0109	73.6486	3.0783	3.0000
	SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	7.2567	26.0398	0.0168	0.0000
F_{19}	Best	−3.8628	−3.8627	−3.8627	−3.8627	−3.8627	−3.8615	−3.8627	−3.8627	−3.8594	−3.8627	−3.8621	−3.8627
	Mean	−3.8628	−3.8627	−3.8627	−3.8627	−3.6792	−3.8523	−3.8612	−3.8627	−3.8376	−2.3801	−3.8585	−3.8627
	Median	−3.8628	−3.8627	−3.8627	−3.8627	−3.8501	−3.8528	−3.8627	−3.8627	−3.8370	−2.0452	−3.8590	−3.8627
	Worst	−3.8628	−3.8627	−3.8627	−3.8627	−2.6533	−3.8253	−3.8525	−3.8627	−3.7950	−1.0008	−3.8508	−3.8627
	SD	0.0000	0.0000	0.0000	0.0000	0.0347	0.0075	0.0033	0.0000	0.0153	1.4155	0.0027	0.0000
F_{20}	Best	−3.3219	−3.3219	−3.3219	−3.3219	−3.3219	−3.1835	−3.3219	−3.3219	−3.1353	−3.3219	−3.3152	−3.3219
	Mean	−3.2506	−3.1619	−3.1941	−3.2662	−2.8543	−3.0052	−3.2443	−3.3219	−2.4380	−3.2992	−3.1821	−3.3219
	Median	−3.2030	−3.2031	−3.2030	−3.3176	−3.1921	−3.0174	−3.2030	−3.3219	−2.6315	−3.3218	−3.2006	−3.3219
	Worst	−3.2030	−0.0305	−1.4406	−3.2031	−0.9493	−2.6386	−3.1355	−3.3219	−1.0701	−3.1383	−2.8097	−3.3219
	SD	0.0592	0.0542	0.3367	0.0600	0.7629	0.1113	0.0712	0.0730	0.6486	0.0587	0.1107	0.0000
F_{21}	Best	−10.1532	−10.1532	−5.0551	−10.1532	−10.1532	−8.3773	−10.1529	−10.1532	−5.0551	−10.1531	−10.1531	−10.1532
	Mean	−5.9048	−6.6034	−5.0551	−8.7602	−6.7545	−5.2045	−7.7734	−8.9636	−5.0551	−7.3908	−10.1321	−5.0048
	Median	−5.0551	−5.0551	−5.0551	−10.1532	−5.0551	−4.8983	−10.1498	−10.1532	−5.0551	−5.0551	−10.1379	−5.0551
	Worst	−5.0551	−1.0900	−5.0551	−5.0551	−5.0551	−3.6232	−5.0551	−5.0551	−5.0551	−5.0551	−10.0776	−5.0551
	SD	1.9323	3.4469	0.0000	2.1979	2.4443	0.9313	2.5861	2.1930	0.0000	2.5502	0.0210	0.0000
F_{22}	Best	−10.4029	−10.4029	−10.4029	−10.4029	−10.4029	−10.1006	−10.4028	−10.4029	−5.0876	−10.4029	−10.4027	−10.4029
	Mean	−6.5050	−4.4344	−5.6191	−9.6763	−8.7629	−5.6410	−8.8073	−9.6942	−5.0876	−6.7903	−10.3854	−8.5692
	Median	−5.0876	−3.7236	−5.0876	−10.4029	−10.4029	−5.2626	−10.4010	−10.4029	−5.0876	−5.0876	−10.3948	−6.0219
	Worst	−5.0876	−3.7236	−5.0876	−4.9223	−3.7240	−3.5913	−5.0876	−5.0876	−5.0876	−3.4921	−10.2891	−6.7155
	SD	2.3906	1.6851	1.6218	1.8485	2.5591	1.4592	2.4767	1.8377	0.0000	2.6708	0.0265	1.0666
F_{23}	Best	−10.5364	−10.5364	−10.5364	−10.5364	−10.5364	−9.6070	−10.5360	−10.5364	−5.1284	−10.5363	−10.5360	−10.5364
	Mean	−8.0636	−4.5480	−6.6452	−9.9915	−8.5889	−5.6939	−9.2732	−10.5364	−5.1284	−6.9016	−10.5207	−8.5383
	Median	−10.5363	−3.8347	−5.1284	−10.5363	−10.5364	−5.3316	−10.5343	−10.5364	−5.1284	−5.1284	−10.5312	−9.2423
	Worst	−3.8354	−3.8347	−5.1284	−5.1284	−3.8354	−3.6411	−5.1284	−10.5364	−5.1284	−1.6985	−10.3619	−3.8646
	SD	2.8990	2.0438	2.4209	1.6488	2.6499	1.2881	2.3256	0.0000	0.0000	2.9063	0.0325	1.8490

Table 10
Wilcoxon Signed Ranks Test.

		SRS–GA	SRS–PSO	SRS–GSA	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–AOA	SRS–AO	SRS–RSA
F_{15}	Z	−19.377[a]	−21.656[a]	−19.377[a]	−15.526[b]	−19.546[a]	−22.104[a]	−18.923[b]	−17.954[b]	−19.392[a]	−2.546[b]	−17.209[b]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.011	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	**SRS > MPA**	SRS < AOA	**SRS > AO**	**SRS > RSA**
F_{16}	Z	−4.540[b]	−2.447[b]	−4.540[b]	−3.991[b]	−5.013[a]	−4.080[a]	−4.475[b]	−3.817[b]	−2.353[a]	−19.590[b]	−2.414[b]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.019	0.000	0.016
	Comparison	**SRS > GA**	**SRS > PSO**	**SRS > GSA**	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	**SRS > MPA**	SRS < AOA	**SRS > AO**	**SRS > RSA**
F_{17}	Z	−4.485[b]	−4.474[b]	−10.083[a]	−4.464[b]	−14.682[a]	−8.539[a]	−4.464[b]	−2.299[a]	−3.545[a]	−19.139[a]	−10.090[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.022	0.000	0.000	0.000
	Comparison	**SRS > GA**	**SRS > PSO**	SRS < GSA	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{18}	Z	−2.044[a]	−4.459[a]	−11.056[a]	−3.227[a]	−15.536[a]	−11.547[a]	−11.685[a]	−16.106[a]	−19 388[a]	−19.690[a]	−6.513[a]
	Sig. (2-tailed)	0.041	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{19}	Z	−8.401[b]	−5.260[b]	−12.847[a]	−8.101[b]	−17.488[a]	−13.923[a]	−7.744[b]	−3.176[a]	−12.532[a]	−19.321[a]	−20.328[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	**SRS > GA**	**SRS > PSO**	SRS < GSA	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{20}	Z	−17.379[a]	−20.815[a]	−11.446[a]	−13.400[a]	−18.952[a]	−19.352[a]	−13.400[a]	−15.272[a]	−18.060[a]	−19.789[a]	−19.434[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{21}	Z	−8.585[a]	−19.812[a]	−19.989[a]	−19.996[a]	−18.757[a]	−19.996[a]	−5.331[a]	−5.331[a]	−9.564[a]	−13.183[a]	−19.983[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{22}	Z	−19.260[a]	−20.092[a]	−11.260[a]	−11.108[a]	−17.584[a]	−19.983[a]	−16.906[a]	−16.951[a]	−9.247[a]	−13.584[a]	−11.365[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA
F_{23}	Z	−19.582[a]	−19.594[a]	−7.337[a]	−11.861[b]	−17.195[a]	−19.555[a]	−19.582[a]	−8.110[b]	−19.978[a]	−11.457[a]	−11.860[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < AOA	SRS < AO	SRS < RSA

[a] Based on positive ranks (SRS is winner).
[b] Based on negative ranks (SRS is failure).

Table 11
Mathematical 2019 test functions.

No.	Functions	F_i^*	D	Search range
F_1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[−8192,8192]
F_2	Inverse Hilbert Matrix Problem	1	16	[−16 384,16 384]
F_3	Lennard-Jones Minimum Energy Cluster	1	18	[−4,4]
F_4	Rastrigin's Function	1	10	[−100,100]
F_5	Griewank's Function	1	10	[−100,100]
F_6	Weierstrass Function	1	10	[−100,100]
F_7	Modified Schwefel's Function	1	10	[−100,100]
F_8	Expanded Schaffer's F6 Function	1	10	[−100,100]
F_9	Happy Cat Function	1	10	[−100,100]
F_{10}	Ackley's Function	1	10	[−100,100]

(CGSA) [97]. Local Escaping Operator and Marine Predators Algorithm (LCOMPA) [98]. Although these methods have a high convergence speed due to use of some special extra and multi-phase equipments, they do not perform well in solving some optimization problems. Therefore, researchers are still trying to come up with a new, and more efficient optimization method. In this study, a new metaheuristic algorithm called the Special Relativity Search (SRS) algorithm is proposed. One of the most significant points in the development of metaheuristic algorithms is the choice of a phenomenon as a source of inspiration. Therefore, the phenomena that are most compatible with optimization problems will perform better. In this research, using electromagnetic concepts, the particles in a magnetic field are selected as the optimal possible answers, and based on this, a mathematical model is proposed to simulate the interaction of particles with each other. Unlike other metaheuristic methods that have used Newtonian physics to develop equations, the framework of the algorithm has been developed based on the theory of Special relativity physics. In order to evaluate the performance of the SRS algorithm, 89 benchmark functions, from a wide range of numerical problems, have been selected and optimized. To compare the performance of the new algorithm with other metaheuristic methods, GA, PSO, GSA, TLBO, GOA, SCA, GWO, MPA, RSA, AOA, AO, and DMOA algorithms have been selected from the literature and their results compared with the new algorithm. In addition, the results of the other two new algorithms, i.e. extended versions of the GSA (CGSA) and MPA (LCOMPA) algorithms, are compared with the novel algorithm. To make a fair comparison the number of population and iteration of all algorithms are set equal, and Wilcoxon Signed-Ranked (WSR) statistical test is performed to rank the best algorithm.

The rest of this paper is organized as follows: Section 2 describes inspiration notions along with the basic concepts of the process of the model as a metaheuristic algorithm. Section 3 presents the problems, parameters tunning, statistical test, Qualitative evaluation of the SRS, and discusses the results. Section 4 presents the main conclusions about the advantage and disadvantages or weaknesses of the proposed algorithm. A summary of the main findings and the potential of the new algorithm to solve various optimization problems in future work are presented in the last section.

2. Special relativity search(SRS)

This section describes the source of inspiration for the SRS algorithm and how to develop equations based on the theory of Special relativity physics. For example, if a particle with an electric charge moves in a magnetic field, the force exerted on the particle in the magnetic field depends on the velocity. The direction of this force will be obtained using the right-hand rule. Suppose a charged particle with mass m in a constant magnetic field. If we consider the direction of the magnetic field B along the z-axis, we can write the equations of motion along the three axes x, y, and z. Since field B is in the direction of the z-axis, it cannot exert a force on the particle along the z-axis. Thus, the particle will move in a circular path. More details are presented in the following subsections.

Table 12
Results of mathematical test Functions (CEC2019).

Fun		GA	PSO	GSA	CGSA	TLBO	GOA	SCA	GWO	MPA	LEOMPA	RSA	SRS
	Best	2.43e+09	1.46e+09	2.30e+11	1.35e+11	9.59e+06	2.08e+08	1.18e+07	3.65e+05	3.19e+04	3.59e+04	1.11e+05	4.88e+04
	Mean	2.02e+10	1.80e+10	2.38e+12	2.02e+12	1.37e+08	8.43e+09	6.33e+09	1.68e+08	1.14e+05	1.06e+05	7.72e+10	6.80e+04
F_1	Median	1.44e+10	7.20e+09	1.93e+12	1.73e+12	1.07e+08	7.52e+09	3.04e+09	7.89e+07	4.64e+04	4.88e+04	1.71e+05	6.05e+04
	Worst	7.87e+10	1.08e+11	8.47e+12	8.36e+12	4.13e+08	2.14e+10	2.41e+10	1.22e+09	5.63e+05	4.50e+05	2.31e+12	1.30e+05
	SD	1.86e+10	2.57e+10	1.91e+12	1.54e+12	1.02e+08	7.02e+09	7.71e+09	2.72e+08	1.20e+05	1.43e+06	4.22e+11	1.96e+04
	Best	17.3582	17.3428	5850.60	5660.37	17.7751	17.3626	17.3964	17.3431	17.3428	1.73428	17.6132	17.3428
	Mean	20.1676	17.3428	13 861.46	12 477.17	46.5292	17.3954	17.4561	17.3437	17.3428	1.73428	18.0382	17.3428
F_2	Median	18.1931	17.3428	13 700.04	12 396.36	18.7061	17.3797	17.4523	17.3437	17.3428	1.73428	17.9547	17.3428
	Worst	33.5765	17.3428	22 175.51	21 619.87	853.147	17.5971	17.6966	17.3445	17.3428	1.73428	18.8095	17.3428
	SD	4.19 678	0.0000	4285.73	4402.75	152.346	0.0446	0.0531	0.0002	0.0000	0.0000	0.3896	0.0000
	Best	12.7032	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024	12.7024
	Mean	12.7053	12.7024	12.7024	12.7026	12.7024	12.7025	12.7025	12.7024	12.7024	12.7024	12.7025	12.7024
F_3	Median	12.7056	12.7024	12.7024	12.7024	12.7024	12.7024	12.7025	12.7024	12.7024	12.7024	12.7025	12.7024
	Worst	12.7078	12.7048	12.7028	12.7044	12.7028	1.27039	12.7026	12.7024	12.7024	12.7031	12.7025	12.7024
	SD	0.0011	0.0000	0.0000	0.0005	0.0000	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Best	11.0122	7.2604	152.2242	0.9955	5.9708	10.9609	444.4445	15.3864	9162.801	5.9698	5105.197	2.9923
	Mean	31.6414	449.6538	431.5752	5.4395	20.7674	29.7214	1133.196	56.0880	25 964.34	11.8139	10 052.070	12.0780
F_4	Median	30.2226	406.8602	337.2596	4.9751	18.9267	25.8697	1084.600	55.3235	22 728.25	10.9667	9440.306	11.9813
	Worst	82.8788	2197.9010	1422.2470	11.9396	57.3392	81.5862	2100.504	92.6843	53 885.75	24.8780	20 058.870	26.8647
	SD	16.1666	563.9228	278.0081	2.6993	11.9957	14.4471	386.9216	18.0953	10 423.95	4.7233	3747.868	5.8112
	Best	1.0203	1.0467	1.0000	1.0000	1.0147	1.0737	1.9110	1.0974	2.8869	1.0172	3.4443	1.0074
	Mean	1.1059	1.3165	1.0552	1.0056	1.0636	1.2637	2.1693	1.3571	5.8168	1.0665	4.3835	1.0529
F_5	Median	1.0849	1.1883	1.0418	1.0000	1.0554	1.2620	2.1803	1.2464	5.9660	1.9616	4.3374	1.0516
	Worst	1.3488	2.1481	1.1868	1.0319	1.1944	1.5090	2.3683	1.7819	8.3204	1.1478	5.8077	1.1107
	SD	0.0731	0.2751	0.0520	0.0084	0.0378	0.1071	0.0911	0.2513	1.2954	0.0294	0.5930	0.0235
	Best	2.6659	6.0697	10.7235	1.2971	8.9419	3.2661	8.9877	7.6529	1.1483	1.2672	8.5047	1.0000
	Mean	4.4326	9.7915	13.8003	1.5229	10.3149	5.7707	10.7065	10.682	2.2648	2.1506	10.8171	1.0000
F_6	Median	4.5005	9.8411	14.0909	1.5190	10.2878	5.5890	10.7334	10.7617	1.9430	1.9983	10.8355	1.0000
	Worst	5.6048	11.6501	16.5644	1.7876	11.3829	8.3782	11.7885	11.6551	4.8336	4.4239	12.1676	1.0001
	SD	0.7631	1.2345	1.3354	0.1197	0.5652	1.4678	0.6047	0.8149	0.9295	0.7861	0.8473	0.0000
	Best	−12.3995	−106.9460	881..2014	79.3216	82.8644	−18.2576	433.0359	−42.8896	−216.8731	36.5651	588.9466	146.1416
	Mean	164.6369	300.7631	1856.367	149.7309	478.6274	350.1502	691.4016	395.8728	−15.9519	186.9589	879.2429	239.5503
F_7	Median	154.7605	268.2821	1915.269	147.9299	463.7414	341.9610	678.3666	323.7804	−24.3506	166.6350	899.4373	241.3146
	Worst	451.6187	831.7658	2642.320	239.9051	808.3885	726.8113	988.7094	1059.0940	189.0830	299.4521	1289.085	671.0371
	SD	160.6008	237.9680	421.7118	41.3594	165.5650	178.9194	132.7630	288.7870	85.3857	39.8596	166.6643	94.6047
	Best	3.3055	3.1961	4.8453	4.5877	2.7168	4.1912	424.5453	2.6736	6.0948	2.5944	5.3687	2.1175
	Mean	4.4459	5.2037	5.8576	5.3106	4.1738	5.3621	741.3140	4.7360	7.7018	3.9799	6.3279	3.9402
F_8	Median	4.5579	5.4666	5.8366	5.2310	4.3075	5.4773	748.1647	4.6388	7.7569	4.0982	6.3755	4.0768
	Worst	5.5445	6.7294	6.3702	6.6818	5.8330	6.3823	988.1529	6.7421	8.8435	5.0083	6.8634	4.8631
	SD	0.5769	0.9463	0.3237	0.4592	0.8085	0.5819	154.9204	1.1305	0.6371	0.6766	0.3425	0.6519
	Best	2.5030	2.3741	3.1192	2.4141	2.3393	2.3618	8.2810	2.4233	1680.392	2.5412	704.3149	2.3435
	Mean	2.8343	2.4660	4.3418	2.6915	2.3452	2.5111	72.6577	4.2553	4096.009	2.6844	1296.2140	2.3971
F_9	Median	2.8226	2.4522	4.2183	2.6916	2.3444	2.4913	67.2750	4.1867	4233.031	2.4726	1345.1430	2.3872
	Worst	3.3642	2.5946	5.8881	3.3473	2.3574	2.8435	326.5745	5.8656	6404.370	2.4925	1950.5610	2.5070
	SD	0.1729	0.0655	0.8314	0.1885	0.0042	0.1218	57.1843	0.8139	1429.761	0.5123	316.8512	0.0426
	Best	20.4616	20.0687	0.0957	0.0016	2.0414	20.0000	20.2924	20.3879	0.0041	2.5799	20.2104	20.0011
	Mean	20.8616	20.3052	19.3441	19.2970	18.2533	20.0466	20.4783	20.4835	16.9194	1.9421	20.4165	20.0131
F_{10}	Median	20.8942	20.2995	20.0068	19.9836	20.3727	20.0148	20.4862	20.4904	20.0000	2.0000	20.4330	20.0079
	Worst	21.2444	20.5708	20.0271	19.9979	20.5523	20.2513	20.5790	20.5813	20.0599	2.0053	20.5651	20.0526
	SD	0.2059	0.1409	3.6354	3.6447	5.6309	0.0697	0.0699	0.0524	7.0365	3.1808	0.0900	0.0119

2.1. The magnetic field

The magnetic interaction is created through an intermediary of the magnetic field B. The magnetic force between two particles in a magnetic field occurs when they are moving at an initial velocity v [99]. Thus, when charged particle Q_i is moving at speed v_i, it produces magnetic field $\mathrm{B_i}$ at distance r from the charged particle Q_j. This process is shown in Fig. 1, the magnetic field B which produced by particle Q_i is defined using Eq. (1). it is easy to understand that two charged particles exert magnetic forces on each other. Consider two particles Q_i and Q_j separated by a distance r and in the same direction, as illustrated in Fig. 2. The force exerted on particle Q_j due to the magnetic field set up by the particle Q_i. Also, Q_j creates a magnetic field $\mathrm{B_j}$ at the location of Q_i. The direction of $\mathrm{B_i}$ is perpendicular to Q_i, as shown in Fig. 2. The direction of $\mathrm{F_i}$ is toward Q_j because $\mathrm{B_j}$ is in that direction. If the field set up at Q_j by Q_i is calculated, the force $\mathrm{F_j}$ acting on Q_j is found to be equal in magnitude and opposite in direction to $\mathrm{F_i}$. This is based on Newton's third law which must be obeyed. When the particles are in opposite directions, the forces are reversed and the particles repel each other. Hence, we find that parallel particles carrying charges in the same direction attract each other, and parallel particles carrying charges in opposite directions repel each other [99].

The magnetic force included of electric force (QE) and magnetic force (QVB). The total force acting on the charged particle Eq. (2) is the sum of the electric and magnetic force known as the Lorentz force. Given that in electric fields E the particles are at rest and their initial velocity is zero ($v = 0$). Therefore, the laws of classical physics still govern the equations of motion. However, in magnetic fields where the particles have an initial velocity, we use magnetic force, which is a function of the velocity of the particles, to measure the force acting on the particles. Eq. (2), known as the Lorentz force, consists of two parts: electric and magnetic. The electric force is omitted due to the uniformity of the magnetic field, and only the magnetic force between the

Table 13
Wilcoxon Signed Ranks Test (CEC2019).

		SRS–GA	SRS–PSO	SRS–GSA	SRS–CGSA	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–ELOMPA	SRS–RSA
F_1	Z	−19.381[a]	−19.380[a]	−19.375[a]	−19.375[a]	−19.445[a]	−19.000[a]	−19.024[a]	−19.396[a]	−15.967[b]	−15.348[b]	−15.899[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS > MPA**	**SRS > MPA**	SRS < RSA
F_2	Z	−19.418[a]	−5.118[a]	−19.396[a]	−19.355[a]	−16.839[a]	−19.223[a]	−14.974[a]	−13.861[a]	−10.068[a]	−13.655[a]	−16.863[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.016
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_3	Z	−19.391[a]	−3.964[a]	−6.598[a]	−11.957[a]	−3.640[a]	−19.391[a]	−12.123[a]	**.000**[c]	−3.833[a]	−5.055[a]	−17.513[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**	0.022	0.022	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	**SRS = GWO**	SRS < MPA	SRS < MPA	SRS < RSA
F_4	Z	−18.720[a]	−19.013[a]	−18.477[a]	−13.418[a]	−16.660[a]	−18.357[a]	−18.015[a]	−17.496[a]	−19.321[a]	−19.018[a]	−19.317[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_5	Z	−15.023[a]	−13.456[a]	−16.084[a]	−15.485[b]	−9.895[a]	−16.563[a]	−18.704[a]	−16.887[a]	−20.178[a]	−20.178[a]	−19.546[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS > GSA**	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_6	Z	−16.218[a]	−19.331[a]	−19.381[a]	−19.374[a]	−19.138[a]	−19.357[a]	−19.357[a]	−19.348[a]	−19.310[a]	−18.878[a]	−19.203[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_7	Z	−17.386[a]	-.093[a]	−19.372[a]	−18.762[b]	−18.587[a]	−19.345[a]	−18.987[a]	−14.340[a]	−1.203[a]	−**6.375**[b]	−19.351[a]
	Sig. (2-tailed)	0.000	0.926	0.000	0.000	0.000	0.000	0.000	0.000	0.229	**0.000**	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	**SRS > GSA**	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS > MPA**	SRS < RSA
F_8	Z	−14.764[a]	−19.359[a]	−19.361[a]	−19.356[a]	−19.265[a]	−19.356[a]	−19.363[a]	−19.209[a]	−19.363[a]	−19.344[a]	−19.363[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_9	Z	−17.716[a]	−17.262[a]	−17.408[a]	−17.704[a]	−5.005[b]	−17.707[a]	−19.392[a]	−16.807[a]	−19.439[a]	−17.368[a]	−19.086[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA
F_{10}	Z	−19.321[a]	−19.344[a]	−8.193[a]	−17.757[a]	−19.180[a]	−19.197[a]	−19.360[a]	−19.267[a]	−18.169[a]	−19.379[a]	−19.380[a]
	Sig. (2-tailed)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Comparison	SRS < GA	SRS < PSO	SRS < GSA	SRS < GSA	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < MPA	SRS < RSA

[a] Based on positive ranks (SRS is winner).
[b] Based on negative ranks (SRS is failure).
[c] The sum of ranks is equal.

particles is utilized.

$$B_i = \mu . Q_i . \frac{v_i}{{r_{ij}}^3} \vec{r} \tag{1}$$

$$F_j = Q_j [E_i + v_j \times B_i] \tag{2}$$

where μ is the magnetic permeability factor, v_i and v_j are the initial velocity of the charged particles i and j, $\vec{r}$ is the unit vector, and r_{ij} is the separate distance between particles i and j.

In Eq. (2) The electric force is ignored $Q_j E = 0$ due to the uniformity of the magnetic field and only the magnetic force between the particles is utilized. Eq. (2) is rewritten as follows:

$$F_m = Q_j v_j B_i \tag{3}$$

2.2. The trajectory of the particle in a magnetic field based on angular frequency

The magnetic force between the particles causes the particles to move in a circular direction [96]. How the particles move in the magnetic field is depicted in Fig. 3. The magnetic force is perpendicular to the velocity vector, so it changes direction in a circular motion but does not change velocity. The magnetic force is always perpendicular to the velocity and does nothing on the particle. For this reason, the kinetic energy and velocity of the particle will remain constant. In such special conditions, the particle velocity is constant, but the direction of motion is variable. The particle moves in a circular path under the influence of a force in the direction of the center of the circle. The reason for this is that the force is radial. The angle between the velocity vector and the force vector is 90 degrees.

As shown in Fig. 4, consider a particle with relative mass m and charge Q moving at a velocity v at an angle of $90°$ to the magnetic field. The amount of Lorentz force that causes the particle to rotate is defined according to Eq. (3). Because the Lorentz force is perpendicular to the velocity of motion, it causes the particle to begin to rotate in a circular path. Therefore, the Lorentz force is defined according to Eq. (4) as a radial force.

$$F_j = m\,a = \frac{mv^2}{r} = Q_j \ v_j \, B_i \tag{4}$$

where m is the mass of the particle, a is the acceleration, and r is the cyclotron radius $r = \frac{mv_j}{Q_j B_i}$.

2.2.1. Calculation of the cyclotron frequency

The charged Q_j begins to move in a uniform magnetic field called a cyclotron resonance. The number of cycles that the particle makes per second around its circular circuit is called the cyclotron frequency (f_j). The cyclotron frequency is defined in Eq. (5). By substituting the value of r from Eq. (4) we have:

$$f_j = \frac{v_j}{2\pi r} = \frac{Q_j B_i}{2\pi m_j} \tag{5}$$

Since the natural period (T) corresponds to the inverse of the cyclotron frequency $T = \frac{1}{f} = \frac{2\pi m}{QB}$. With a natural period, we can easily obtain the cyclotron frequency in radians per second by Eq. (6).

$$T = \frac{2\pi}{\omega_n} = \frac{2\pi m}{QB} \quad \Rightarrow \quad \omega_n = \frac{QB}{m} \tag{6}$$

2.2.2. Motion of the particle bias to the magnetic field

Generally, the velocity v of a charged particle has an arbitrary angle with the vector of the magnetic field B, which is at an acute angle $\theta, 0 < \theta < \pi/2$. To study this type of motion, the velocity vector must be analyzed. By analyzing the velocity vector, two components are obtained, one parallel and the other perpendicular to the magnetic field B (Eq. (7)). According to the hypotheses previously stated, the component perpendicular to the magnetic field B produces a centripetal force that causes the charged particle to move in a circular path. In other words, the parallel component of the velocity vector does not produce magnetic force, and the particle moves in the direction of the magnetic field B at a constant speed without being accelerated. Accordingly, we use the vertical component Eq. (9) of the velocity vector to move the particle along a circular path. Fig. 5 shows the path of a particle in a circular path and the magnetic field B perpendicular to the plane.

$$m\dot{v}_x = Q\, v_y\, B\,, \qquad m\dot{v}_y = -Q\, v_x B \tag{7}$$

$$m\ddot{v}_x = \frac{QB}{m}\dot{v}_y = -(\frac{QB}{m})^2 v_x = -\omega_n^2 v_x \tag{8}$$

where $\dot{v}_x$ and $\ddot{v}_x$ are the velocity acceleration, respectively in x direction.

The horizontal component of velocity is $v_x = v\cos(\omega_n t)$. By substituting in Eq. (8), we will have:

$$v_y = \frac{m}{QB}\dot{v}_x = -\frac{|Q|}{Q} v \sin(\omega_n\, t) \tag{9}$$

By integrating Eq. (9), the new coordinate of a particle at any given time will be determined by Eq. (10).

$$x_{new} = x_{old} + \frac{v}{\omega_n}\sin(\omega_n\, t) \tag{10}$$

2.3. The theory of special relativity physics

The theory of special relativity was first introduced by Albert Einstein in 1905 [96]. Newton accepted the center of the sun as an inertial frame, but the Ether inertial frame was intended to emit light. Light as a wave from a mechanical point of view needs a medium to propagate. Therefore, the space between the stars should not be empty and it is filled with a substance called Ether. Accordingly, in classical physics, the velocity of particles is measured based on Ether. However, in special relativity physics, particle velocities are measured based on the speed of light, and for the first time in this work equations of motion are developed based on the special relativity physics. In special relativity physics, there are two significant phenomena called length contraction and time dilation, which increase the accuracy of the equations. In metaheuristic algorithms, the most important part is developing an efficient equation for the main step of the algorithm that includes all the characteristics of motion. Special relativity physics is a good option for developing a new equation that includes the position, direction, and velocity of a particle. However, to obtain a new equation, it is first necessary to introduce and formulate the two phenomena of length contraction and time dilation using Lorentz transformations. Then develop the original equation using inverse Lorentz transformations. As mentioned previously, in special relativity physics, equations based on the speed of light are developed. The reason for this assumption is that the speed of light in all coordinates is absolute and higher than the speed of any phenomenon in the universe. In the physics of special relativity, as closely as the velocity of the particles to the speed of light, the equations are more accurate. Therefore, in this work, it is assumed that the particle's speed is very high and close to the speed of light.

2.3.1. Length contraction

Consider a bar in the frame o' parallel to the x' axis whose length is L'. To measure, the observer must measure and subtract the coordinates of the two ends of the bar at the same time (Eq. (12)).

$$x_i{}' = \frac{x_i - vt_i}{\sqrt{1-\frac{v^2}{c^2}}}, \qquad x_j{}' = \frac{x_j - vt_j}{\sqrt{1-\frac{v^2}{c^2}}} \tag{11}$$

$$x'_j - x'_i = \frac{x_j - x_i - v(t_j - t_i)}{\sqrt{1-\beta^2}}, \qquad t_i = t_j,\ \ x_j - x_i = L, \tag{12}$$
$$L' = L_0, \quad L = L_0\sqrt{1-\beta^2}$$

where $x_i{}'$ and $x_j{}'$ are the initial and end point, respectively. L_0 represents the actual length and L is the contracted length. v is the velocity of the moving observer, and c denotes the speed of light.

2.3.2. Time dilation

Consider two events in o' at a fixed location x' whose time interval is t, the t' will be determined using Eq. (13).

$$\Delta t = \frac{\Delta t' + \frac{v}{c^2}\Delta x'}{\sqrt{1-\beta^2}}, \qquad \Delta x' = 0,\ \Delta t' = \Delta t_0,\ \Delta t = \frac{\Delta t_0}{\sqrt{1-\beta^2}} \tag{13}$$

that is, moving clocks work slower. Special time is the minimum time calculated by the resident observer.

2.3.3. Inverse Lorentz transformations

In this section uses previous information and reverse Lorentz transformations, the main equation of motion is derived. In summary, the conversion of coordinates in an inertia frame that is relativistically correct was determined by the following two governing postulates:

(1) In all inertia frames, light travels isotropically at a constant speed c,
(2) All inertial reference frames are equally valid in expressing physical laws.

By applying the above two postulates, Lorentz transformations can be obtained preliminarily. Therefore, if we consider the two orthogonal coordinates E_i and E_j, which move at a constant relative speed U along their x-axis. In this case, if we show the coordinates of an event in the first frame with x_i, y_i, z_i, and t_i the coordinates of the same event in the second frame with x_j, y_j, z_j, and t_j, then the Lorentz conversion which represents the conversion relations between the coordinates of the event when going from a frame to another frame is defined using Eqs. (14) and (15).

$$t_j = \frac{t_i - \frac{v}{c^2}x_i}{\sqrt{1-(\frac{v}{c})^2}} \Rightarrow t_i - (\frac{v}{c^2})x_i = t_j\sqrt{1-(\frac{v}{c})^2}$$
$$\Rightarrow t_1 = (\frac{v}{c^2})x_i + t_j\sqrt{1-(\frac{v}{c})^2} \tag{14}$$

$$x_j = \frac{x_i - vt}{\sqrt{1-(\frac{v}{c})^2}} \Rightarrow x_i - vt_i = x_j\sqrt{1-(\frac{v}{c})^2}$$
$$\Rightarrow x_i = vt_i + x_j\sqrt{1-(\frac{v}{c})^2} \tag{15}$$

By substituting Eqs. (14) to (15), the vector motion of the answers in each step is obtained by Eq. (16).

$$x_i = (\frac{v}{c})^2 x_i + v\, t_j\sqrt{1-(\frac{v}{c})^2} + x_j\sqrt{1-(\frac{v}{c})^2} \tag{16}$$

where the velocity vector v is a function of the mass and charge of the particles, which is determined by Eq. (17). By substituting the values of v and x_j (the value of x_j is equal $R_{ij}\sin(\omega_n\, t)$ Eq. (24)) in Eq. (16), the equation of the main step of the algorithm is defined by Eq. (18).

$$v = \omega_n\, r = \frac{QB}{m}r = \mu\frac{Q_iQ_j}{m{r_{ij}}^3}v_j\, r_{ij} = \mu\frac{Q_iQ_j}{m{r_{ij}}^2}v_j \tag{17}$$

$$x_i = (\frac{v}{c})^2 x_i + \left[\mu\frac{Q_iQ_j}{m{r_{ij}}^2}v_j\right] t_j\sqrt{1-(\frac{v}{c})^2} + R_{ij}\sin(\omega_n)\sqrt{1-(\frac{v}{c})^2} \tag{18}$$

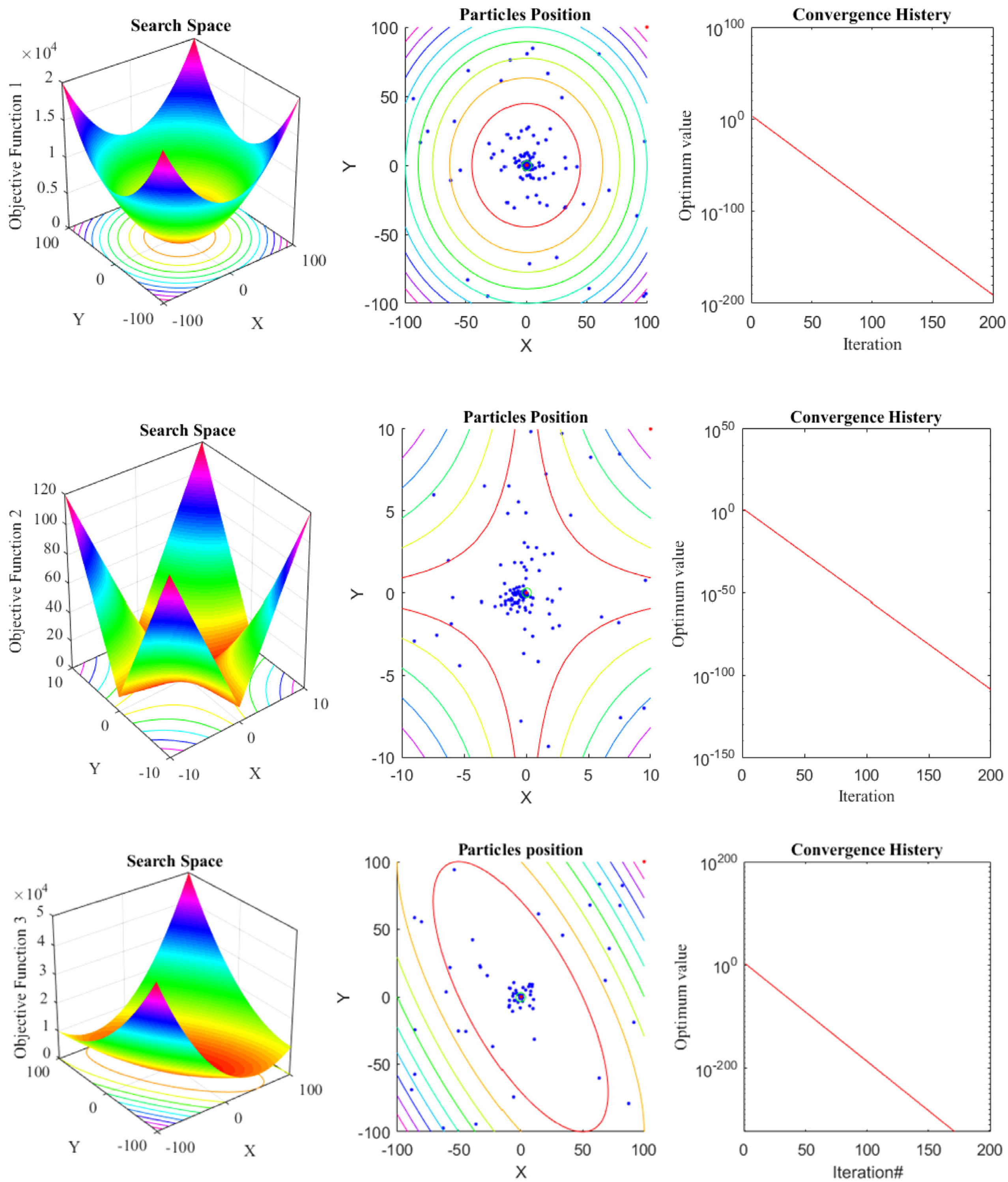

Fig. 7. Qualitative evaluation of functions 1 through 3.

2.4. The SRS mathematical simulation

This section describes the proposed mathematical model for simulating the SRS algorithm. The SRS algorithm is a single-objective algorithm that searches for the local and global optimal answer in the feasible search space. Like other metaheuristic methods, SRS is a population-based algorithm. In this way, a population is initialized first. Then, a new population of optimal possible answers is generated. Using an iteration process, the new population is merged into the current population using some selection procedures. The search process stops when the max iteration is obtained, which is called the "stop criterion". In the following, the different parts of the SRS algorithm are formulated and explained through steps 1 to 7.

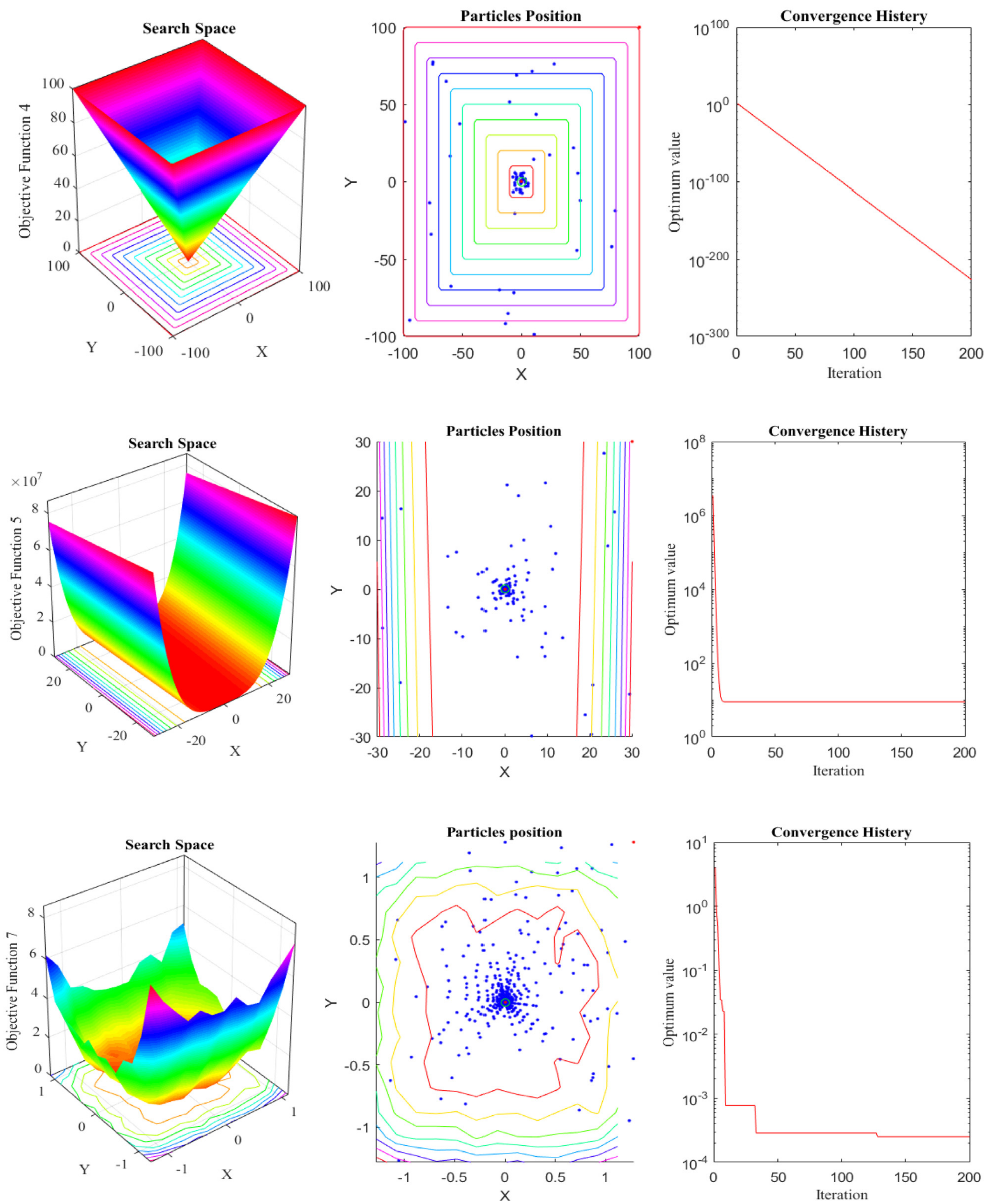

Fig. 8. Qualitative evaluation of functions 4, 5, and 7.

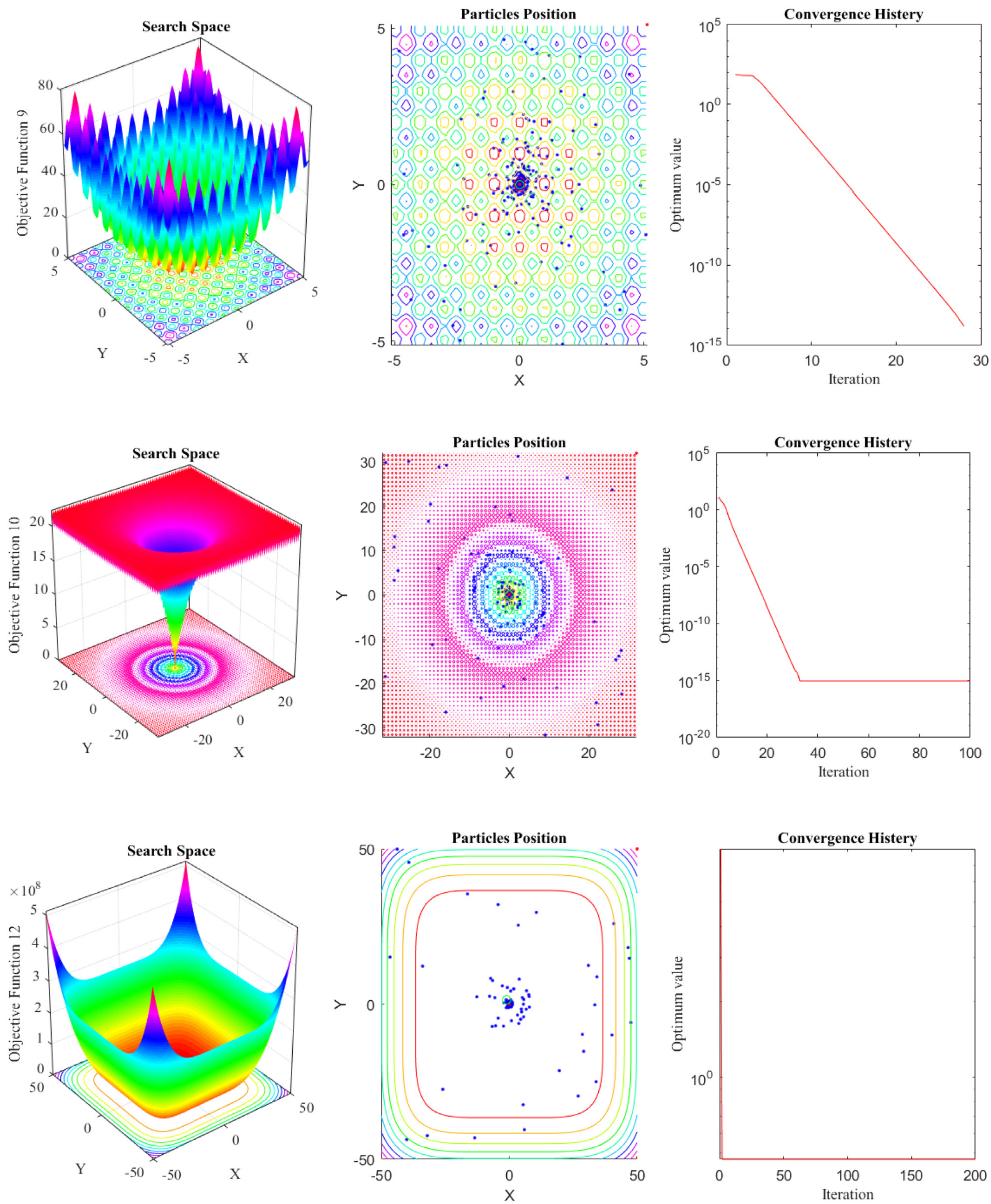

Fig. 9. Qualitative evaluation of functions 9, 10, and 12.

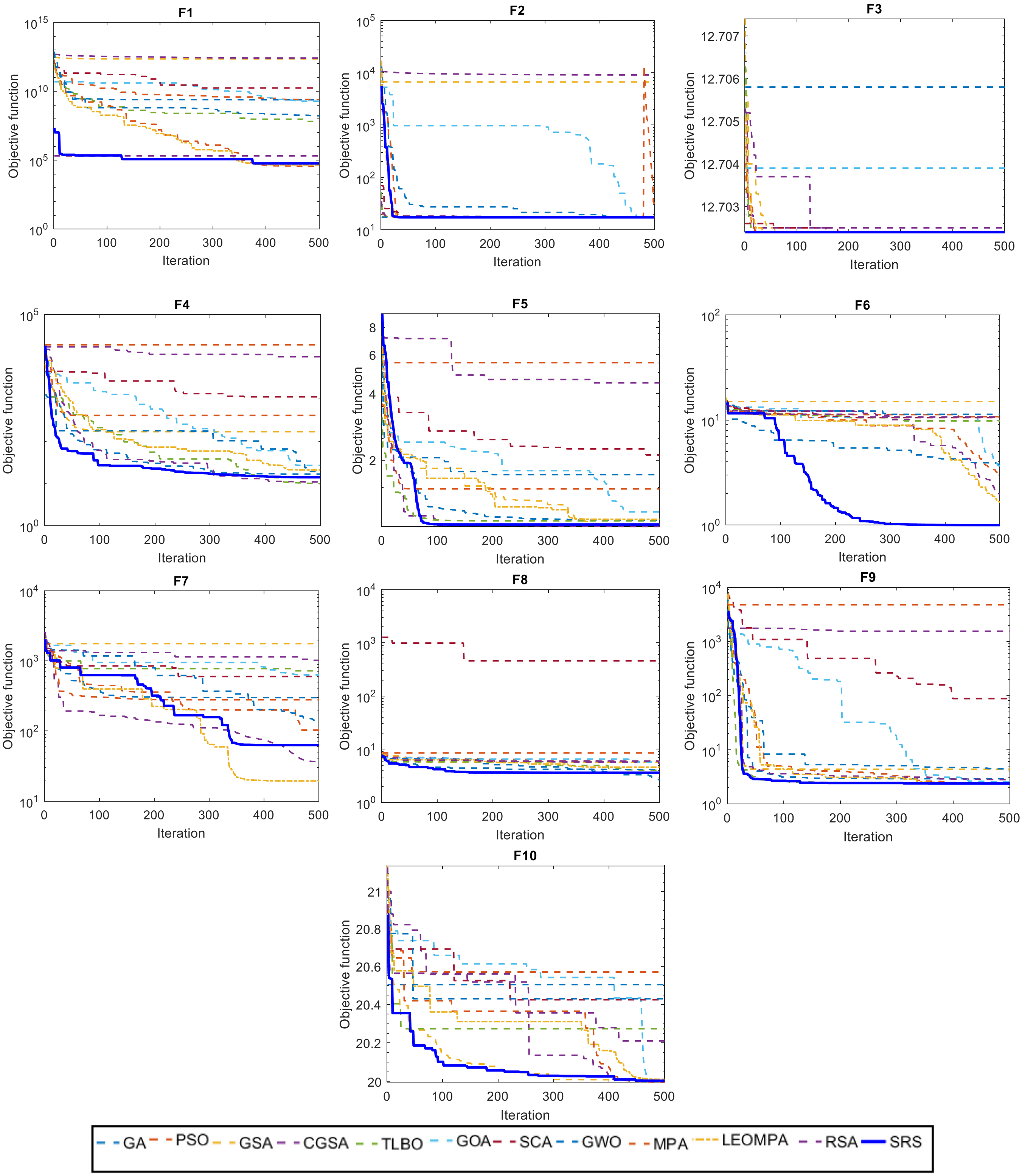

Fig. 10. Convergence history of functions 1 through 10 (CEC 2019).

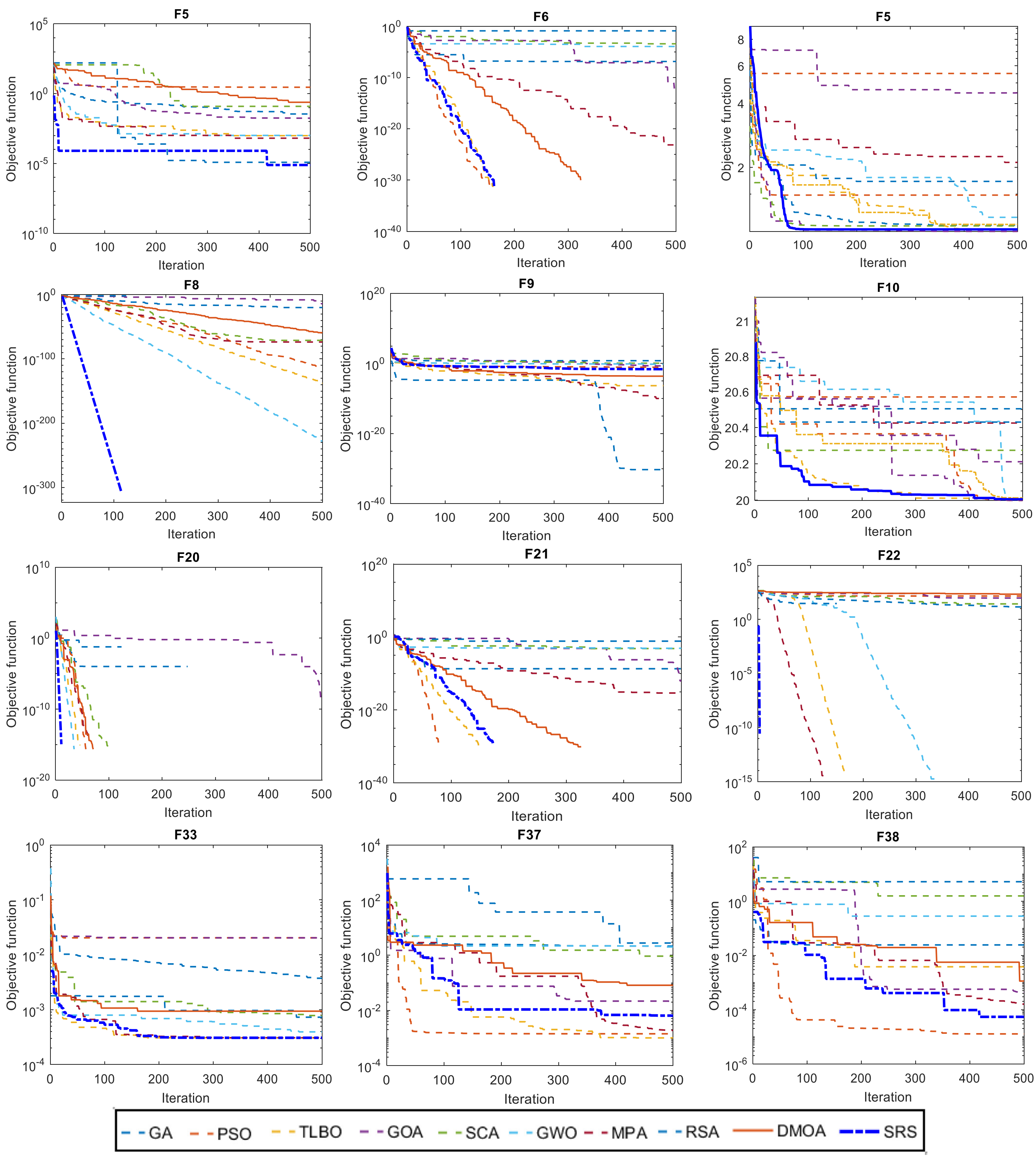

Fig. 11. Convergence history of the test Functions.

Table 14
Information of benchmark functions.

Type	Function	Initialization range	D	Minimum answer
Unimodal separable	$\mathbf{F_1}$: Stepint	[−5.12, 5.12]	5	0
	$\mathbf{F_2}$: Step	[−100, 100]	30	0
	$\mathbf{F_3}$: Sphere	[−100, 100]	30	0
	$\mathbf{F_4}$: SumSquares	[−10, 10]	30	0
	$\mathbf{F_5}$: Quartic	[−1.28, 1.28]	30	0
Unimodal non-separable	$\mathbf{F_6}$: Beale	[−4.5, 4.5]	2	0
	$\mathbf{F_7}$: Easom	[−100, 100]	2	−1
	$\mathbf{F_8}$: Matyas	[−10, 10]	2	0
	$\mathbf{F_9}$: Colville	[−10, 10]	4	0
	$\mathbf{F_{10}}$: Trid6	$[-D^2, D^2]$	6	−50
	$\mathbf{F_{11}}$: Trid10	$[-D^2, D^2]$	10	−210
	$\mathbf{F_{12}}$: Zakharov	[−5, 10]	10	0
	$\mathbf{F_{13}}$: Powell	[−4, 5]	24	0
	$\mathbf{F_{14}}$: Schwefel 2.22	[−10, 10]	30	0
	$\mathbf{F_{15}}$: Schwefel 1.2	[−100, 100]	30	0
	$\mathbf{F_{16}}$: Rosenbrock	[−30, 30]	30	0
	$\mathbf{F_{17}}$: Dixon-Price	[−10, 10]	30	0
Multimodal separable	$\mathbf{F_{18}}$: Foxholes	[−65.536, 65.536]	2	0.998
	$\mathbf{F_{19}}$: Branin	[−5, 10] × [0, 15]	2	0.398
	$\mathbf{F_{20}}$: Bohachevsky1	[−100, 100]	2	0
	$\mathbf{F_{21}}$: Booth	[−10, 10]	2	0
	$\mathbf{F_{22}}$: Rastrigin	[−5.12, 5.12]	30	0
	$\mathbf{F_{23}}$: Schwefel	[−500, 500]	30	−12,569.5
	$\mathbf{F_{24}}$: Michalewicz2	$[0, \pi]$	2	−1.8013
	$\mathbf{F_{25}}$: Michalewicz5	$[0, \pi]$	5	−4.6877
	$\mathbf{F_{26}}$: Michalewicz 10	$[0, \pi]$	10	−9.6602
Multimodal non-separable	$\mathbf{F_{27}}$: Schaffer	[−100, 100]	2	0
	$\mathbf{F_{28}}$: Six Hump Camel Back	[−5, 5]	2	−1.03163
	$\mathbf{F_{29}}$: Bohachevsky2	[−100, 100]	2	0
	$\mathbf{F_{30}}$: Bohachevsky3	[−100, 100]	2	0
	$\mathbf{F_{31}}$: Shubert	[−10, 10]	2	−186.73
	$\mathbf{F_{32}}$: GoldStein-Price	[−2, 2]	2	3
	$\mathbf{F_{33}}$: Kowalik	[−5, 5]	4	0.00031
	$\mathbf{F_{34}}$: Shekel5	[0, 10]	4	−10.15
	$\mathbf{F_{35}}$: Shekel7	[0, 10]	4	−10.4
	$\mathbf{F_{36}}$: Shekel10	[0, 10]	4	−10.53
	$\mathbf{F_{37}}$: Perm	[−D, D]	4	0
	$\mathbf{F_{38}}$: Powersum	[0, 1]	4	0
	$\mathbf{F_{39}}$: Hartman3	[0, D]	3	−3.86
	$\mathbf{F_{40}}$: Hartman6	[0, 1]	6	−3.32
	$\mathbf{F_{41}}$: Griewank	[−600, 600]	30	0
	$\mathbf{F_{42}}$: Ackley	[−32, 32]	30	0
	$\mathbf{F_{43}}$: Penalized	[−50, 50]	30	0
	$\mathbf{F_{44}}$: Penalized2	[−50, 50]	30	0
	$\mathbf{F_{45}}$: Langermann2	[0, 10]	2	−1.08
	$\mathbf{F_{46}}$: Langermann5	[0, 10]	5	−1.5
	$\mathbf{F_{47}}$: Langermann 10	[0, 10]	10	NA
	$\mathbf{F_{48}}$: Fletcher Powell2	$[-\pi, \pi]$	2	0
	$\mathbf{F_{49}}$: Fletcher Powell5	$[-\pi, \pi]$	5	0
	$\mathbf{F_{50}}$: FletcherPowell10	$[-\pi, \pi]$	10	0

Step 1: A population of optimal possible answers is generated using Eq. (19). Then, using Eq. (20), we stochastically select a set of candidates between the upper and lower bound.

$$X_{ij} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{P1} & X_{P2} & \cdots & X_{Pj} & \cdots & X_{Pn} \end{bmatrix} \tag{19}$$

$$X_{ij}(t+1) = lb + rand \times (ub - lb), \quad j = 1, 2, \ldots, n \tag{20}$$

where X_{ij} is a population of possible answers consisting of particles with position *jth* and solution *ith* at iteration time t, n denotes the number of variables, P represents the number of population. lb and ub also indicate the lower and upper bound, respectively.

Step 2: The distance between the two particles X_i and X_j in the magnetic field is determined using the Euclidean norm. (Eq. (21)).

$$R_{ij}(t) = norm(X_i(t) - X_j(t)) \tag{21}$$

where R_{ij} is the separate distance between particles X_i and X_j at iteration time t.

Step 3: Calculation of the charge of Q_i and Q_j particles in a magnetic field based on particle fitness is carried out by Eq. (22).

$$Q_i(t) = Q_j(t) = \frac{Fit_i(t) - Worst(t)}{Global(t) - Worst(t)} \tag{22}$$

Table 15
Results of mathematical test Functions.

Type	Fun		GA	PSO	TLBO	GOA	SCA	GWO	MPA	RSA	DMOA	SRS
Unimodal separable	F_1	Best	−25.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	−1.09e+75	0.0000
		Mean	−25.0000	0.0000	0.0000	1.2666	0.0000	0.0000	0.0000	1.1666	−4.87e+73	0.0000
		Median	−25.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	−2.61e+70	0.0000
		Worst	−25.0000	0.0000	0.0000	9.0000	0.0000	0.0000	0.0000	5.0000	−8.30e+58	0.0000
		SD	0.0000	0.0000	0.0000	2.0330	0.0000	0.0000	0.0000	2.0356	2.01e+74	0.0000
	F_2	Best	0.0000	1.0000	0.0000	8.0000	0.0000	0.0000	0.0000	0.0000	3.0000	0.0000
		Mean	0.4666	671.33	0.0000	31.7000	4.1666	0.0000	0.0000	0.0000	10.4333	0.0000
		Median	0.0000	4.0000	0.0000	30.0000	2.0000	0.0000	0.0000	0.0000	10.5000	0.0000
		Worst	3.0000	1e+04	0.0000	65.0000	22.0000	0.0000	0.0000	0.0000	17.0000	0.0000
		SD	0.6814	2537.31	0.0000	16.2292	2.6938	0.0000	0.0000	0.0000	3.1914	0.0000
	F_3	Best	0.7461	0.0000	0.0000	0.4520	0.0015	0.0000	0.0000	0.0000	3.7581	0.0000
		Mean	1.2378	333.3333	0.0000	3.1816	4.6323	0.0000	0.0000	0.0000	5.7970	0.0000
		Median	1.1477	0.0000	0.0000	2.6547	0.7301	0.0000	0.0000	0.0000	5.5908	0.0000
		Worst	1.9707	1e+04	0.0000	8.6161	53.5518	0.0000	0.0000	0.0000	8.5128	0.0000
		SD	0.3430	1825.7420	0.0000	1.9002	11.3578	0.0000	0.0000	0.0000	1.4015	0.0000
	F_4	Best	0.0824	0.0000	0.0000	0.2388	0.0001	0.0000	0.0000	0.0000	0.2896	0.0000
		Mean	0.1690	393.3385	0.0000	9.4212	0.5028	0.0000	0.0431	0.0000	0.6947	0.0000
		Median	0.1667	350.0000	0.0000	8.9535	0.1282	0.0000	0.0000	0.0000	0.7446	0.0000
		Worst	0.3593	1000.0000	0.0000	20.7431	3.0240	0.0000	1.2949	0.0000	1.0172	0.0000
		SD	0.0605	335.2068	0.0000	6.4939	0.7987	0.0000	0.2364	0.0000	0.1746	0.0000
	F_5	Best	0.0223	0.1102	0.0004	0.0121	0.0097	0.0004	0.0003	0.0000	0.1668	0.0000
		Mean	0.0416	0.8862	0.0008	0.0290	0.0730	0.0010	0.0011	0.0000	0.2659	0.0000
		Median	0.0426	0.2400	0.0008	0.0293	0.0555	0.0009	0.0011	0.0000	0.2573	0.0000
		Worst	0.0555	0.1095	0.0015	0.0413	0.2244	0.0026	0.0023	0.0001	0.4679	0.0001
		SD	0.0094	2.1791	0.0002	0.0075	0.0603	0.0004	0.0005	0.0000	0.0751	0.0000

where Fit_i is the fitness value of particles *ith* at iteration time t, *Worst* and *Global* is the maximum and minimum values of the objective function at iteration time t, respectively.

Step 4: The velocity of the particles are calculated based on Eq. (17).

Step 5: As mentioned before, the frequency of the cyclotron is a function of the magnetite field B, the charge of Q, and the mass of the particles. We use Eq. (6) to calculate the value of ω_n Eq. (23).

$$\omega_n = \frac{Q_i B}{m} \xrightarrow{B=\mu \frac{Q_j v_j}{R^3{}_{ij}}} \mu \frac{Q_i Q_j v_j}{m R^3{}_{ij}} \tag{23}$$

Step 6: Using Eq. (10), the new coordinate of the particle is determined, but to use Eq. (10) in the SRS algorithm, we need to set $v = \omega_n R_{ij}$, and develop a new equation (Eq. (24)).

$$X_j = X_j + \frac{v}{\omega_n}\sin(\omega_n) \xrightarrow{v=\omega_n R_{ij}} \frac{\omega_n R_{ij}}{\omega_n}\sin(\omega_n) = R_{ij}\sin(\omega_n) \tag{24}$$

Step 7: The velocity and new position of the particle are obtained using Eq. (25), and by substituting the values obtained from steps 1 to 6 we have:

$$X_{ij}(t+1) = \beta^2 X_j(t) + V_j(t)\sqrt{1-\beta^2} + X_j(t)\sqrt{1-\beta^2} \tag{25}$$

where V_j is calculated in step 4 at iteration time t, β is the ratio of the particle velocity to the speed of light ($3e{+}8$ m/s), $\beta = \frac{v}{c}$. According to the special relativity physics, the velocity of any particle is not faster than the speed of light [96]. Therefore, we set the value of β equal to the random value between [0, 1].

After determining the new position and velocity of the particles in step 7, we must make sure that the optimal answer is in the feasible space. Therefore, the optimum vector must be between the upper and lower bounds. The initial particle velocity is a random value that needs to be updated at each iteration. We now calculate the value of the objective function. In order to avoid the local optimal point, it is necessary to update the local and global optimal vectors.

2.5. Time and space complexity

The SRS is easy to implement and its algorithmic complexity is easy to calculate. The time complexity of the SRS in worst case is $\Theta(Pop \times D)$, in which the *Pop* is the population size related to the number of Particles and D is the dimension of the problem. The computational complexity of the objective function (F(x)) evaluation in the initialization stage of the SRS is calculated as $\Theta(Pop) \times \Theta(F(x))$. In the next stage, the main loop of the algorithm is started based on maximum number of iterations (T). In this loop, the charge and the coordinate of each particle are calculated, so the computational complexity of the position updating is $\Theta(T \times Pop \times D)$. Consequently, the objective function evaluation in the main loop has computational complexity of $\Theta(T \times Pop) \times \Theta(F(x))$. The SRS complexities are calculated on 10, 30 and 50 dimensions, to show the algorithm complexity's relationship with dimension. The computational complexity is calculated as described in [100]. The results of computational complexity are presented in Table 1.

T_1 is the computing time for Function 3 of CEC2005 for 200,000 evaluations of a certain dimension D, while T_2 is the complete running time of the SRS algorithm of the same D dimensional for Function 3, with a total budget of 200,000 evaluations. T_2 is evaluated five times, and the mean for T_2 is denoted as $\overline{T_2}$. Finally, the algorithm complexity is shown as $(\overline{T}_2 - T_1)/T_0$

3. Numerical problems and discussions

In recent years, many different kinds of metaheuristic algorithms have been developed to solve optimization problems. Performance evaluation of these methods requires a systematic approach because some metaheuristic algorithms are compatible with problems and work well. Therefore, to evaluate the overall performance of the algorithm, it is necessary to examine a wide range of problems. In this study, 83 mathematical benchmark functions with hard, medium, and simple complexity levels have been investigated to evaluate the performance of the SRS algorithm, including 23 CEC2005 [9], which are divided

Table 16
Wilcoxon Signed Ranks Test (high dimentional).

		SRS–GA	SRS–PSO	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–DMOA	SRS–RSA
	Z	−5.477[a]	.000[c]	.000[c]	−3.321[a]	.000[c]	.000[c]	.000[c]	-	−2.588[a]
F_1	Sig. (2-tailed)	.000	1.000	1.000	.001	1.000	1.000	1.000	-	.010
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	**SRS = SCA**	**SRS = GWO**	**SRS = MPA**	NA	SRS < RSA
	Z	−3.357[a]	−4.792[a]	.000[c]	−4.783[a]	−3.928[a]	.000[c]	.000[c]	−4.789[a]	.000[c]
F_2	Sig. (2-tailed)	.001	.000	1.000	.000	.000	1.000	1.000	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	**SRS = GWO**	**SRS = MPA**	SRS < DMOA	**SRS = RSA**
	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.789[a]	−3.059[a]
F_3	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.002
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
	Z	−4.782[a]	−4.785[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−2.803[a]	−4.789[a]	.000[c]
F_4	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.005	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS = RSA**
	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.789[a]	−1.923[a]
F_5	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.054
	Comparison	SRS < GA	SRS < PSO	SRS > TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

into four groups: unimodel, basic multimodal, expanded multimodal, and hybrid composition multimodal. 10 difficult problems CEC2019 [10] and 50 benchmark functions [10] which are a wide range of problems are also evaluated.

3.1. Parameter tuning

To fairly compare the results of the SRS algorithm with GA, PSO, GSA, CGSA, TLBO, GOA, SCA, GWO, MPA, LEOMPA, RSA, DMOA, AOA, and AO all problems are solved under the same conditions and population size and the iteration is set to 50 and 500, respectively. To evaluate the efficiency and capability of the SRS and other algorithms, all of the 83 benchmark functions are run 30 times. The best mean, median, and SD results are recorded. Then, the best algorithm is distinguished using the WRS statistical test.

3.2. Statistical test

The Wilcoxon Signed-Ranked (WSR) test was used for the statistical analysis of results and selection of the superior algorithm. The WSR is a non-parametric method based on rankings and their distribution. The WSR test is used to examine two dependent samples or the compatibility between two samples. It should be noted that this test is considered for the average of the abnormal community. However, this test is non-parametric and does not depend on the distribution of data. To perform this test, the following hypotheses must be considered:

(a) The data should be arranged in pairs.
(b) Each component of pairs is randomly selected and is independent of other samples.
(c) The type or scale of data should be distance or relative to distance so that the difference between their values can be obtained and these differences can be ranked.

The results of the SRS algorithm for each function are compared in pairs with other methods. In this test, there are three main parameters that determine the status of SRS compared to other methods, which are: 1- b = Based on positive ranks, ie the results obtained from the SRS algorithm are superior to other methods. 2- c = Based on negative ranks, i.e. the results of the SRS algorithm are worse than other methods. 3- d = The sum of ranks is equal, i.e. the results of the SRS are equal to other methods and not many changes have been made. Tables 4, 6, 8, 10, 13, 16, 18, 20, 22, and 24 show the results of the WSR test for all 83 numerical problems.

3.3. Evaluation of the CEC2005 test functions

In this section, 23 benchmark functions which include unimodel, basic multimodal, expanded multimodal, and hybrid composition multimodal are presented and experiments are conducted on SRS, GA, PSO, GSA, TLBO, GOA, SCA, GWO, MPA, AOA, AO, and RSA. Table 2 provides the type of function, upper bound, lower bound, and other details. For more information about these functions, refer to the reference Ref. [9].

The first class of problems are unimodal functions, which include F_1 to F_5. These functions have only one global answer and are a good option for testing the exploration ability of the SRS algorithm and other metaheuristic methods. The results of the SRS and other algorithms are provided in Table 3. According to the results presented for F_1 to F_4, the SRS obtain the best answer compared to other metaheuristic methods. The results of RSA, AO, MPA, and TLBO algorithms for F_5 are better than other methods. Compared to the SRS, there is not much difference between the optimal answer, but the Std value of SRS is better than RSA, and TLBO, which indicates the stability of SRS in solving function 5. The GA and GOA algorithms did not perform well in solving unimodal functions and almost obtained the worst answers compared to other methods. This indicates that some metaheuristics may not perform well even in solving problems with only one global optimal answer. However, it cannot be concluded on this basis, so to ensure the efficiency of metaheuristics, a wide range of problems must be solved to evaluate their overall performance. In general, the exploration ability of the SRS to find the global answer is superior to other methods.

The second class of problems is the functions of the basic multimodal type, which include functions F_6 to F_{12}. In these problems, which have a huge local answer and a global answer, it is difficult to achieve a global answer by increasing the dimensions of the problem. Using these problems, two important features of exploration and exploitation of the SRS and other methods can be evaluated. The results of SRS and other methods are shown in Table 5. The SRS results for F_8, are superior to other metaheuristic methods. The SRS is optimally responding in F_6, but it has got the second place next to TLBO. In solving function 7, SRS has obtained the optimal global answer, but with a slight difference is in second place after AO. In solving F_9 the SRS, RSA, AOA, AO, and MPA performed the same job. The results of AOA and AO for F_{10} are equal to SRS. Also, SRS, RSA, MPA, GWO, TLBO, AOA, AO, and GSA have the same rank in solving F_{11}. The TLBO algorithm obtain a good result for $F_{12}$and SRS despite reaching the global

Table 17
Results of mathematical test Functions.

Type	Fun		GA	PSO	TLBO	GOA	SCA	GWO	MPA	RSA	DMOA	SRS
Unimodal non-separable	F_6	Best	0.0000	0.0196	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
		Mean	0.0000	1.0707	0.0000	0.1016	0.0001	0.0000	0.0000	0.1525	0.0000	0.0000
		Median	0.0000	0.7268	0.0000	0.0000	0.0000	0.0000	0.0000	0.0405	0.0000	0.0000
		Worst	0.0004	3.8847	0.0000	0.7620	0.0005	0.0000	0.0000	0.6562	0.0000	0.0000
		SD	0.0000	1.0701	0.0000	0.2634	0.0001	0.0000	0.0000	0.1981	0.0000	0.0000
	F_7	Best	−1.0000	−0.9655	−1.0000	−1.0000	−0.9999	−1.0000	−1.0000	−0.9998	−1.0000	−1.0000
		Mean	−0.8000	−0.2927	−1.0000	−0.8333	−0.9985	−0.9999	−1.0000	−0.9950	−0.7259	−1.0000
		Median	−1.0000	−0.8819	−1.0000	−1.0000	−0.9991	−0.9999	−1.0000	−0.9969	−0.9881	−1.0000
		Worst	−0.0000	−0.0000	−1.0000	0.0000	−0.9931	−0.9999	−1.0000	−0.9721	−9.83e−09	−1.0000
		SD	0.4068	0.3585	0.0000	0.3790	0.0016	0.0000	0.0000	0.0060	0.4380	0.0000
	F_8	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Worst	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	F_9	Best	3.0305	0.0007	0.0000	0.0000	0.3933	0.0000	0.0000	0.0000	0.0000	0.0051
		Mean	16.8900	0.0657	0.0000	0.7469	1.2324	1.2379	0.0000	0.0000	0.0217	0.8362
		Median	16.5553	0.0536	0.0000	0.0616	1.3644	1.0767	0.0000	0.0000	0.0112	0.3471
		Worst	34.6028	0.1929	0.0000	7.8769	2.7266	7.8710	0.0000	0.0000	0.0102	5.8595
		SD	8.8151	0.0503	0.0000	1.7348	0.5149	1.6653	0.0000	0.0000	0.0247	1.5827
	F_{10}	Best	−49.9985	−50.0000	−50.0000	−50.0000	−48.5233	−49.9999	−50.0000	−38.9073	−50.0000	−49.9996
		Mean	−49.8175	−43.6000	−50.0000	−50.0000	−43.9925	−49.9999	−50.0000	−28.1843	−49.9407	−49.9109
		Median	−49.8567	−50.0000	−50.0000	−50.0000	−44.3664	−49.0000	−50.0000	−27.7323	−49.9790	−49.9550
		Worst	−49.4599	142.000	−50.0000	−50.0000	−38.2997	−49.0000	−50.0000	−15.9428	−49.2808	−49.5118
		SD	0.1591	35.0542	0.0000	0.0000	2.7994	0.0001	0.0000	6.7666	0.1309	0.1264
	F_{11}	Best	−208.701	−209.998	−210.000	−209.999	−113.655	−209.995	−210.000	−75.9463	−195.1554	−209.999
		Mean	−168.260	52.511	−210.000	−209.512	29.3376	−193.203	−210.000	−41.9597	−152.2941	−209.925
		Median	−174.501	−206.619	−210.000	−209.860	−40.7119	−209.985	−210.000	−39.9119	−152.7635	−209.944
		Worst	−107.455	125.266	−209.999	−207.008	1202.60	−77.1409	−210.000	−24.9143	−102.5935	−209.745
		SD	29.2256	531.730	0.0000	8.2148	255.962	34.0261	0.0000	11.6065	20.6677	0.06618
	F_{12}	Best	0.1057	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000
		Mean	1.2389	2.7791	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0363	0.0000
		Median	1.0343	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0193	0.0000
		Worst	3.2297	31.5802	0.0000	0.0005	0.0000	0.0000	0.0000	0.0000	0.1699	0.0000
		SD	0.8470	8.5251	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0445	0.0000
	F_{13}	Best	0.0178	0.2808	0.0000	0.3750	0.0001	0.0000	0.0000	0.0000	15.9165	0.0000
		Mean	0.6239	301.283	0.0000	2.3184	4.2457	0.0000	0.0000	0.0000	82.5071	0.0000
		Median	0.6021	211.005	0.0000	1.4888	0.2454	0.0000	0.0000	0.0000	65.6777	0.0000
		Worst	1.6808	1315.23	0.0000	8.2895	47.635	0.0003	0.0000	0.0000	187.6275	0.0000
		SD	0.3313	287.928	0.0000	2.0452	11.647	0.0000	0.0000	0.0000	48.8776	0.0000
	F_{14}	Best	0.2469	0.0042	0.0000	1.4849	0.0001	0.0000	0.0000	0.0000	0.8387	0.0000
		Mean	0.3197	25.6922	0.0000	8.8058	0.0066	0.0000	0.0000	0.0000	21.3705	0.0000
		Median	0.3138	20.1092	0.0000	4.1679	0.0033	0.0000	0.0000	0.0000	15.0856	0.0000
		Worst	0.3850	80.0000	0.0000	82.8386	0.0387	0.0000	0.0000	0.0000	76.3884	0.0000
		SD	0.0376	18.4457	0.0000	17.3797	0.0091	0.0000	0.0000	0.0000	19.7966	0.0000
	F_{15}	Best	9.8236	0.0000	0.0000	33.2532	0.0286	0.0000	0.0000	0.0000	25.8686	0.0000
		Mean	16.2297	41 000.0	0.0000	437.936	26.6780	0.0000	0.0000	0.0000	68.5379	0.0000
		Median	16.0200	40 000.0	0.0000	226.381	3.5705	0.0000	0.0000	0.0000	66.9468	0.0000
		Worst	25.9729	120 000	0.0000	3578.99	222.803	0.0000	0.0000	0.0000	124.3002	0.0000
		SD	3.7122	35 559.5	0.0000	767.638	53.5268	0.0000	0.0000	0.0000	23.1129	0.0000
	F_{16}	Best	72.7803	77.4995	20.0414	150.718	81.2064	26.0434	23.8077	0.0000	5.44e+04	28.9374
		Mean	273.916	18 812.4	23.0998	634.208	7281.77	26.9104	24.4214	19.3063	1.46e+05	28.9653
		Median	165.916	305.242	23.2629	589.795	1532.10	26.9823	24.3790	28.9759	1.35e+05	28.9613
		Worst	1602.26	90 246.7	24.6482	1537.33	84 601.1	28.7479	25.2704	28.9940	2.88e+05	29.0000
		SD	361.795	363 035	0.9775	405.677	17 082.2	0.83547	0.40880	13.8851	6.79e+04	0.01823
	F_{17}	Best	0.9016	1.5966	0.9870	0.7134	0.74034	0.6666	0.6666	0.6666	307.7312	0.6666
		Mean	4.2146	7365.34	0.5500	12.1337	89.0508	0.6666	0.6666	0.7570	683.6395	0.6666
		Median	3.8560	87.5414	0.9947	7.5941	14.1085	0.6666	0.6666	0.6666	651.1792	0.6666
		Worst	8.7861	72 583.1	1.6502	48.5344	1188.50	0.6667	0.6667	1.0000	1580.592	0.6666
		SD	2.0428	22 079.6	0.0030	13.2376	244.951	0.0000	0.0000	0.1491	294.3677	0.0000

optimal ranked second place after TLBO. In this class of problems, the GA algorithm still obtains the worst results. While GOA has performed better in solving this class of problems that are more difficult than the unimodal functions.

The third class of problems is the expanded multimodal functions, which include F_{13} and F_{14}. Using a *2-D* function $F(x, y)$ as a starting function, corresponding expanded function Eq. (26) [9]. The results of SRS and other metaheuristic methods are provided in Table 7. The results show that SRS is superior to other methods in solving expanded functions. The results show that GSA, SCA, AOA, and AO algorithms did not perform efficiently in solving these problems and converged to the local optimal. As long as these methods have achieved good results in solving unimodal and basic multimodal problems. But the interesting thing is that

Table 18
Wilcoxon Signed Ranks Test (high dimentional).

		SRS–GA	SRS–PSO	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–DMOA	SRS–RSA
F_6	Z	−4.782[a]	−4.782[a]	−1.000[b]	−4.783[a]	−4.782[a]	−4.782[a]	−4.782[a]	−1.000[b]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.317	.000	.000	.000	.000	0.317	.000
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS > DMOA**	SRS < RSA
F_7	Z	−2.333[a]	−4.785[a]	.000[c]	−2.236[a]	−4.703[a]	.000[c]	.000[c]	−3.623[a]	−4.782[a]
	Sig. (2-tailed)	.020	.000	1.000	.025	.000	1.000	1.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	**SRS = GWO**	**SRS = MPA**	SRS < DMOA	SRS < RSA
F_8	Z	-.768[a]	-.768[a]	-.768[a]	-.768[a]	-.768[a]	-.768[a]	-.768[a]	−4.165[a]	−2.366[b]
	Sig. (2-tailed)	.443	.443	.443	.443	.443	.443	.443	0.000	.018
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS > RSA**
F_9	Z	−4.782[a]	−3.898[b]	−4.782[b]	−1.882[b]	−2.869[a]	−1.594[a]	−4.782[b]	−4.659[b]	−4.782[b]
	Sig. (2-tailed)	.000	.000	.000	.060	.004	.111	.000	0.000	.000
	Comparison	SRS < GA	**SRS > PSO**	**SRS > TLBO**	**SRS > GOA**	SRS < SCA	SRS < GWO	**SRS > MPA**	**SRS > DMOA**	**SRS > RSA**
F_{10}	Z	−2.170[a]	−4.165[b]	−4.782[b]	−4.782[b]	−4.782[a]	−4.782[b]	−4.782[b]	−1.697[b]	−4.782[a]
	Sig. (2-tailed)	.030	.000	.000	.000	.000	.000	.000	0.090	.000
	Comparison	SRS < GA	**SRS > PSO**	**SRS > TLBO**	**SRS > GOA**	SRS < SCA	**SRS > GWO**	**SRS > MPA**	**SRS > DMOA**	SRS < RSA
F_{11}	Z	−4.782[a]	−4.412[a]	−4.782[b]	−2.808[a]	−4.782[a]	-.113[b]	−4.782[b]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.000	.005	.000	.910	.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	**SRS > MPA**	SRS < DMOA	SRS < RSA
F_{12}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	.000[c]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS = RSA**
F_{13}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−2.521[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.012
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{14}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−3.296[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.001
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{15}	Z	−4.782[a]	−4.784[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−1.604[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.109
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{16}	Z	−4.782[a]	−4.782[a]	−4.782[b]	−4.782[a]	−4.782[a]	−4.782[b]	−4.782[b]	−4.782[a]	−1.964[b]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.049
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	**SRS > GWO**	**SRS > MPA**	SRS < DMOA	**SRS > RSA**
F_{17}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−1.000[a]	.000[c]	−4.782[a]	−3.064[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.317	1.000	0.000	.002
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	SRS < DMOA	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

GA and GOA algorithms, which did not perform efficiently in solving unimodal and basic multimodal problems, have achieved the global optimal solution to these types of problems.

$$EF(x_1, x_2, \ldots, x_D) = F(x_1, x_2) + F(x_2, x_3) + \cdots + F(x_{D-1}, x_D) + F(x_D, x_1) \tag{26}$$

In order to prove the SRS convergence to the global optimum, some of the most challenging problems in Section 3.3 have been selected and the SRS convergence history has been compared with other methods. Given the continuity of the problems, the importance of proving convergence is an significant challenge to demonstrate the algorithm's capabilities in exploration and exploitation. The convergence diagram of F_1 to F_{12} for SRS and other metaheuristic methods is shown in Fig. 6. Problems 1 to 5 have only one global optimum, and the convergence rate relative to the global optimum is one of the important factors in proving superiority. In solving F_1, F_2, F_3, and F_4, the SRS has the highest convergence rate relative to the global optimum. As shown in the figure, it converges with minimal iteration and high speed relative to the global optimum. The AO algorithm convergence rate for F_5 is better than other methods. Competition for convergence to optimal answer is more important in F_6 to F_{12}. Moreover, there is a huge number of local optimums in these functions and they are very likely to converge with respect to the local answer. However as shown in Fig. 6, the SRS converges faster than other methods to the optimal global answer. The convergence rate of the TLBO algorithm in F_6 is slightly better than that of SRS.

The fourth class of problems are hybrid composition multimodal functions, which include F_{15} to F_{23}. This type of problem is a combination of unimodal, multimodal, expanded multimodal problems, which have the most local optimum. Many of the algorithms that have been developed so far have not solved these problems well, and only a limited number of them have been able to solve these problems. Algorithms that have high power of exploration and exploitation succeed in solving these functions. The results of SRS and other methods are presented in Table 9. The SRS has performed well in solving functions F_{15} and F_{16} such as MPA, RSA, GWO, and TLBO, also GA, PSO, GSA, GOA, SCA, AOA, and AO did the same performance and has been able to solve these functions with a slight difference. SRS results for functions F_{18}, F_{20}, F_{21}, and F_{22} are superior and outperformed other methods. The results show that SRS is in the second place after TLBO in solving F_{23} with a slight difference. Interestingly, the GA algorithm, which did not perform well in the first and two classes of problems, performed better than the SRS in solving F_{17}.

3.3.1. Qualitative evaluation of the SRS

To show the SRS convergence rate in solving benchmark problems, several functions are selected and analyzed graphically. Figs. 7, 8, and 9 include 3 columns. The first column shows the

Table 19
Results of mathematical test Functions.

Type	Fun		GA	PSO	TLBO	GOA	SCA	GWO	MPA	RSA	DMOA	SRS
Multimodal separable	F_{18}	Best	0.9980	0.9980	0.9980	1.9956	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
		Mean	3.2034	0.9980	0.9980	9.0156	1.5291	3.9405	0.9980	4.1169	0.9980	0.9980
		Median	1.9920	0.9980	0.9980	12.670	0.9983	2.9821	0.9980	2.9821	0.9980	0.9980
		Worst	6.9033	0.9980	0.9980	12.670	2.9821	12.670	0.9980	10.789	0.9980	0.9980
		SD	2.1577	2.1577	0.0000	4.3008	0.8911	3.7527	0.0000	3.2523	0.0000	0.0000
	F_{19}	Best	0.3980	0.4351	0.3978	0.3980	0.3979	0.3978	0.3978	0.3982	0.3978	0.3978
		Mean	4.5161	0.7181	0.3978	0.4135	0.3992	0.3978	0.3978	0.4079	0.3978	0.3978
		Median	1.2627	0.6953	0.3978	0.4082	0.3987	0.3978	0.3978	0.4021	0.3978	0.3978
		Worst	19.421	1.5902	0.3978	0.4965	0.4021	0.3978	0.3978	0.4523	0.3978	0.3978
		SD	6.2630	0.2874	0.0000	0.0213	0.0011	0.0000	0.0000	0.0121	0.0000	0.0000
	F_{20}	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Mean	0.0072	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Median	0.0016	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Worst	0.0812	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		SD	0.0162	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	F_{21}	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000
		Mean	0.0015	0.0000	0.0000	0.0000	0.0010	0.0024	0.0000	0.1025	0.0000	0.0000
		Median	0.0000	0.0000	0.0000	0.0000	0.0007	0.0000	0.0000	0.0742	0.0000	0.0000
		Worst	0.0086	0.0000	0.0000	0.0000	0.0053	0.0729	0.0000	0.7128	0.0000	0.0000
		SD	0.0027	0.0000	0.0000	0.0000	0.0012	0.1332	0.0000	0.1572	0.0000	0.0000
	F_{22}	Best	2.3213	76.0572	0.0000	10.9814	0.4145	0.0000	0.0000	0.0000	187.6144	0.0000
		Mean	8.3248	125.304	11.224	74.9804	42.987	1.1511	0.0000	0.0000	227.8316	0.0000
		Median	8.1164	117.216	11.445	78.3460	37.255	0.0000	0.0000	0.0000	228.5329	0.0000
		Worst	18.040	173.909	21.923	11.8615	17.880	8.2789	0.0000	0.0000	259.9036	0.0000
		SD	3.5162	25.0911	6.3006	25.7569	41.009	2.3151	0.0000	0.0000	15.4847	0.0000
	F_{23}	Best	−8333.31	−10 506.0	−9597.07	−8901.16	−4813.06	−7731.86	−9751.26	−5257.61	−2.74e+63	−8926.53
		Mean	−7329.67	−8896.04	−7857.59	−7191.52	−3539.53	−5692.11	−8989.79	−5127.28	−1.30e+62	−7502.35
		Median	−7386.73	−8874.58	−7797.67	−7006.69	−3463.64	−5710.19	−9000.38	−5174.81	−1.42e+60	−7527.83
		Worst	−5673.21	−7018.62	−5391.06	−5931.16	−2935.16	−3183.63	−7603.71	−4831.86	−4.09e+56	−5456.13
		SD	562.957	762.192	973.5353	816.006	406.504	826.9787	441.9186	125.209	5.07e+62	919.238
	F_{24}	Best	−1.8013	−1.8013	−1.8013	−1.8013	−1.8008	−1.8013	−1.8013	−1.8008	−1.9992	−1.8013
		Mean	−1.8013	−1.8013	−1.8013	−1.6090	−1.7454	−1.8013	−1.8013	−1.7560	−1.9428	−1.8013
		Median	−1.8013	−1.8013	−1.8013	−1.8013	−1.7995	−1.8013	−1.8013	−1.7743	−1.9879	−1.8013
		Worst	−1.8013	−1.8013	−1.8013	−1.0000	−1.0000	−1.8012	−1.8013	−1.6543	−1.8013	−1.8013
		SD	0.0000	0.0000	0.0000	0.3288	0.20266	0.0000	0.0000	0.0453	0.0786	0.0000
	F_{25}	Best	−4.6876	−4.6876	−4.6876	−4.6875	−3.8593	−4.6874	−4.6876	−4.2901	−4.8742	−4.6876
		Mean	−4.5727	−4.4691	−4.6257	−3.4304	−2.9248	−4.4150	−4.6876	−3.4397	−4.7803	−4.6385
		Median	−4.6458	−4.6458	−4.6458	−3.4928	−2.8020	−4.5165	−4.6876	−3.4166	−4.8009	−4.6458
		Worst	−4.2113	−3.7491	−4.4831	−1.9634	−1.9332	−3.5367	−4.6876	−2.7210	−4.6536	−4.4958
		SD	0.1206	0.1806	0.0703	−0.6878	0.4097	0.3549	0.0000	0.3859	0.015	0.0643
	F_{26}	Best	−9.6201	−9.5150	−9.4258	−7.5539	−5.3194	−9.2098	−9.5758	−6.6706	−8.3590	−3.9827
		Mean	−9.0099	−8.0929	−9.0880	−6.1125	−4.2772	−7.7043	−8.9797	−5.5160	−6.8609	−3.2931
		Median	−9.1114	−7.9260	−9.1254	−6.1700	−4.1878	−7.7580	−8.9324	−5.4734	−6.8218	−3.2434
		Worst	−7.8603	−6.1894	−8.5437	−4.6405	−3.2807	−5.2883	−8.5187	−4.5684	−5.8300	−2.3893
		SD	0.4465	0.9455	0.2102	0.9375	0.5494	0.8969	0.3068	0.4571	0.0523	0.4441

search space contour view. The second column shows a two-dimensional view of the position of the particles in the search space. The third column shows the convergence history for the optimal answer.

Fig. 7 shows the F_1, F_2, and F_3 functions that only have one global optimum and the SRS convergence behavior to the optimal answer. Because there is only one optimal answer in these functions, the particles converge to the optimal answer at a high speed. Fig. 8 shows two types of problems that have local and global optimums. F_4 and F_5 have only one global optimum, and F_7 has the local optimum. As it can be seen, in F_4 and F_5, the particles converge to the optimal answer at high speed. F_7 shows the equilibrium between exploration and exploitation for convergence to the optimal answer. Fig. 9 shows the F_9, F_{10}, and F_{12} functions, which are known for evaluating exploration and exploitation. F_9 has a huge number of local optimum, but SRS has converged to the optimal response at a high rate. The diagrams of F_{10} and F_{12} also show that SRS made an improvement in both exploration and exploitation.

3.4. Evaluation of the CEC2019 test functions

In this section, 10 mathematical functions [10] are selected to evaluate the capabilities of the SRS algorithm and other metaheuristic methods. The development of single-objective metaheuristic algorithms is the basis for solving more complex problems. Hence single-objective mathematical problems are the best option to prove the capabilities of population-based algorithms. In general, single-objective problems are divided into several categories: dynamic, niching composition, computational expensive, etc. In order to better understand the SRS algorithm and other metaheuristic methods, 10 single-objective mathematical problems are optimized. Recently, the idea of developing original algorithms to introduce a superior method has been very much welcomed by researchers. Therefore, in this research, two of the newly developed algorithms CGSA and LEOMPA selected and their results are compared with the SRS algorithm. The results of other metaheuristics methods listed in Table 12 are compared with the new algorithm. The details of these functions are provided in Table 11.

Table 20
Wilcoxon Signed Ranks Test (high dimentional).

		SRS–GA	SRS–PSO	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–DMOA	SRS–RSA
F_{18}	Z	−4.237[a]	.000[c]	.000[c]	−4.871[a]	−4.641[a]	−4.093[a]	.000[c]	0.000[c]	−4.727[a]
	Sig. (2-tailed)	.000	1.000	1.000	.000	.000	.000	1.000	1.000	.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{19}	Z	−4.782[a]	−4.782[a]	.000[c]	−4.782[a]	−4.782[a]	−1.732[a]	.000[c]	0.000[c]	−4.782[a]
	Sig. (2-tailed)	.000	.000	1.000	.000	.000	.083	1.000	1.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{20}	Z	−4.782[a]	.000[c]	.000[c]	−4.782[a]	.000[c]	.000[c]	.000[c]	0.000[c]	.000[c]
	Sig. (2-tailed)	.000	1.000	1.000	.000	1.000	1.000	1.000	1.000	1.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	**SRS = SCA**	**SRS = GWO**	**SRS = MPA**	**SRS = DMOA**	**SRS = RSA**
F_{21}	Z	−4.782[a]	.000[c]	.000[c]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	0.000[c]	−4.782[a]
	Sig. (2-tailed)	.000	1.000	1.000	.000	.000	.000	.000	1.000	.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS = DMOA**	SRS < RSA
F_{22}	Z	−4.782[a]	−4.782[a]	−4.541[a]	−4.782[a]	−4.782[a]	−3.180[a]	.000[c]	−4.782[a]	.000[c]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.001	1.000	.000	1.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	SRS < DMOA	**SRS = RSA**
F_{23}	Z	-.895[a]	−4.268[b]	−1.594[b]	−1.162[a]	−4.782[a]	−4.720[a]	−4.741[b]	-	−4.782[a]
	Sig. (2-tailed)	.371	.000	.111	.245	.000	.000	.000	-	.000
	Comparison	SRS < GA	**SRS > PSO**	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS > MPA**	NA	SRS < RSA
F_{24}	Z	.000[c]	.000[c]	.000[c]	−2.598[a]	−4.784[a]	.000[c]	.000[c]	−4.298[b]	−4.782[a]
	Sig. (2-tailed)	1.000	1.000	1.000	.009	.000	1.000	1.000	.000	.000
	Comparison	**SRS = GA**	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	**SRS = GWO**	**SRS = MPA**	**SRS > DMOA**	SRS < RSA
F_{25}	Z	−2.386[a]	−2.335[a]	-.540[a]	−4.638[a]	−4.782[a]	−4.228[a]	−4.054[b]	−4.623[a]	−4.782[a]
	Sig. (2-tailed)	.017	.020	.589	.000	.000	.000	.000	.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS > MPA**	SRS < DMOA	SRS < RSA
F_{26}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.618[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

The results of the SRS algorithm and other metaheuristic methods are presented in Table 12. F_1, F_2, and F_3 are multimodal with one global optimum, very highly conditioned, and fully parameter-dependent, which is for testing the exploration of SRS and other methods. The results of SRS for F_2 and F_1 are superior to other methods, but in the third function, the SRS and GWO results are equal. F_4, F_5, and F_6 are multimodal and non-separable with a large number of local optimizations that challenge both the exploration and exploitation properties of the algorithms. The results of SRS for F_4, F_5, and F_6 are superior to other metaheuristic methods. The SRS results for F_8 and F_{10}, which have a large number of local optimizations, are superior to other metaheuristic methods. The best results for F_7 belong to RSA and PSO, and SRS is in the third place. In solving F_9, the SRS algorithm is in the second place after TLBO. However the Std results of the SRS are better than TLBO, which indicates the stability of the new algorithm. The results show that the developed CGSA and LEOMPA algorithms have performed better than their initial version only in some cases. It can be concluded that the NFL theory is also true for this type of algorithm. Therefore, the developed algorithms, like the original methods, are incapable of solving some optimization problems. The convergence history of SRS and other methods are shown in Fig. 10.

3.5. Evaluation of the 50 benchmark functions

To investigate the performance of the SRS algorithm in solving various problems with real conditions, 50 mathematical functions have been chosen [11]. These functions are divided into several categories such as unimodal separable, unimodal non-separable, multimodal separable, and multimodal non-separable. More information about these functions is provided in Table 14. The convergence history of SRS and other methods are shown in Fig. 11.

Functions F_1 to F_5 are of the unimodal separable type, which is not dependent on specific conditions and has only one global optimum. The results of SRS, GA, PSO, TLBO, GOA, SCA, GWO, MPA, RSA, and DMOA other metaheuristic methods are presented in Table 15. The SRS algorithm has got the best answer in solving these functions with good performance. This indicates that SRS has a high exploration ability to find the optimal answer. Other methods have shown good performance in some cases but have not been good in others. This is why the SRS algorithm with zero SD has reached the best optimal answer. RSA, MPA, GWO, and TLBO were able to achieve good results only in some cases. The GA, PSO, and DMOA algorithms did not perform well in solving F_1, F_2, F_3, F_4, and F_5. For example, the GA algorithm obtained a completely skewed result for the F_1, and the DMOA algorithm did not perform well in solving this function. GA performed better than DMOA in solving F_2, but both of these methods failed to achieve the optimal global answer.

Functions F_6 to F_{17} are unimodal non-separable and are completely dependent on conditions and parameters. The results of SRS and other methods are presented in Table 17. In F_9, the SRS algorithm could not reach the optimal answer with a small difference and ranked in the third place after TLBO, MPA, and RSA algorithms. In F_{10}, GOA, TLBO, DMOA, and MPA algorithms achieved the optimal answer and the SRS algorithm was in the fifth place. In F_{11}, the TLBO and MPA algorithms managed to achieve the optimal response and the SRS algorithm came in the third place with a slight difference. In other functions, the SRS algorithm performed well and achieved the optimal answer. The DMOA performs better than the SRS algorithm in solving F_9 and F_{10} functions. But as you can see, the GA, PSO, and GOA

Table 21
Results of mathematical test Functions.

	Fun		GA	PSO	TLBO	GOA	SCA	GWO	MPA	RSA	DMOA	SRS
Multimodal non-separable	F_{27}	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Mean	0.0379	0.0014	0.0000	0.0000	0.0000	0.0014	0.0000	0.0000	0.0000	0.0000
		Median	0.0436	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Worst	0.0437	0.0436	0.0000	0.0000	0.0000	0.0436	0.0000	0.0000	0.0000	0.0000
		SD	0.0149	0.0079	0.0000	0.0000	0.0000	0.0079	0.0000	0.0000	0.0000	0.0000
	F_{28}	Best	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Mean	−1.0316	−1.0316	−1.0316	−1.0316	−1.0315	−1.0316	−1.0316	−1.0309	−1.0316	−1.0316
		Median	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0314	−1.0316	−1.0316
		Worst	−1.0316	−1.0316	−1.0316	−1.0316	−1.0314	−1.0316	−1.0316	−1.0264	−1.0316	−1.0316
		SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0012	0.0000	0.0000
	F_{29}	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Mean	0.0007	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Worst	0.0144	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		SD	0.0026	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	F_{30}	Best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Mean	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Worst	0.0036	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		SD	0.0006	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	F_{31}	Best	−186.73	−186.73	−186.73	−186.730	−186.730	−186.730	−186.73	−186.70	−186.73	−186.73
		Mean	−186.73	−186.73	−186.73	−182.123	−186.293	−186.706	−186.73	−180.87	−186.73	−186.73
		Median	−186.73	−186.73	−186.73	−186.730	−186.587	−186.730	−186.73	−184.52	−186.73	−186.73
		Worst	−186.72	−186.73	−186.73	−48.5068	−183.738	−186.307	−186.73	−146.05	−186.73	−186.73
		SD	0.0002	0.0000	0.0000	25.2361	0.6419	0.0828	0.0000	9.9479	0.0024	0.0000
	F_{32}	Best	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
		Mean	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0002	3.0000	3.0000
		Median	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
		Worst	3.0000	3.0000	3.0000	3.0000	3.0002	3.0000	3.0000	3.0020	3.0000	3.0000
		SD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0004	0.0000	0.0000
	F_{33}	Best	0.0005	0.0004	0.0003	0.0006	0.0004	0.0003	0.0003	0.0004	0.0007	0.0003
		Mean	0.0032	0.0069	0.0004	0.0056	0.0010	0.0037	0.0003	0.0010	0.0009	0.0141
		Median	0.0013	0.0016	0.0004	0.0012	0.0011	0.0003	0.0003	0.0009	0.0010	0.0038
		Worst	0.0203	0.0225	0.0004	0.0203	0.0015	0.0203	0.0003	0.0021	0.0010	0.1665
		SD	0.0058	0.0090	0.0001	0.0083	0.0003	0.0075	0.0000	0.0004	0.0000	0.0315
	F_{34}	Best	−10.153	−10.153	−10.153	−10.153	−7.0421	−10.152	−10.153	−5.0551	−10.153	−10.153
		Mean	−8.3998	−6.1345	−10.153	−6.2238	−3.7618	−9.4767	−10.153	−5.0551	−10.153	−10.153
		Median	−10.153	−5.0779	−10.153	−5.0779	−4.5675	−10.151	−10.153	−5.0551	−10.153	−10.153
		Worst	−2.6828	−2.6304	−10.153	−2.6304	−0.4972	−5.0551	−10.153	−5.0551	−10.150	−10.153
		SD	3.0189	3.2588	0.0000	3.3915	1.7756	1.7507	0.0000	0.0000	0.0000	0.0000
	F_{35}	Best	−10.402	−10.402	−10.402	−10.402	−6.2235	−10.402	−10.402	−5.0876	−10.4029	−10.402
		Mean	−7.8987	−7.9020	−10.223	−7.0548	−3.6358	−10.401	−10.402	−5.0876	−10.4029	−10.402
		Median	−10.402	−10.402	−10.402	−7.7658	−4.7229	−10.401	−10.402	−5.0876	−10.4029	−10.402
		Worst	−2.7519	−1.8375	−5.0126	−1.8375	−0.5239	−10.400	−10.402	−5.0876	−10.4027	−10.402
		SD	3.4047	3.4086	0.9841	3.5217	1.9977	0.0006	0.0006	0.0000	0.0000	0.0006
	F_{36}	Best	−10.536	−10.536	−10.536	−10.536	−8.2256	−10.536	−10.536	−5.1284	−10.536	−10.536
		Mean	−8.9045	−7.4136	−10.536	−7.8732	−4.7485	−10.534	−10.536	−5.1284	−10.536	−10.536
		Median	−10.536	−10.536	−10.536	−10.536	−4.8493	−10.535	−10.536	−5.1284	−10.536	−10.536
		Worst	−2.4217	−1.8594	−10.536	−1.8594	−0.9457	−10.532	−10.536	−5.1284	−10.536	−10.536
		SD	3.0366	3.4847	0.0000	3.6529	1.5404	0.0008	0.0000	0.0000	0.0000	0.0000
	F_{37}	Best	0.0016	0.0000	0.0003	0.0009	0.1323	0.0041	0.0000	0.0337	0.0048	0.0000
		Mean	1.2856	0.0962	0.0211	0.0450	0.8792	2.8213	0.0010	5.7726	0.0947	0.0052
		Median	0.6188	0.0067	0.0058	0.0129	0.6758	1.0779	0.0006	2.6648	0.0590	0.0006
		Worst	5.3057	0.4723	0.1306	0.3758	2.1876	45.225	0.0049	26.2277	0.7754	0.1306
		SD	1.5301	0.1579	0.0356	0.0754	0.6185	8.3096	0.0014	6.6533	0.1429	0.0237
	F_{38}	Best	0.0001	0.0000	0.0000	0.0000	0.0347	0.0003	0.0000	0.0000	0.0009	0.0000
		Mean	0.0376	0.0041	0.0010	0.0090	2.1321	0.1463	0.0001	0.0000	0.0061	0.0000
		Median	0.0219	0.0022	0.0005	0.0015	0.9943	0.0064	0.0000	0.0000	0.0048	0.0000
		Worst	0.2133	0.0213	0.0087	0.0898	20.753	0.8827	0.0003	0.0000	0.0210	0.0000
		SD	0.0489	0.0053	0.0017	0.0183	5.0164	0.2842	0.0001	0.0000	0.0049	0.0000

algorithms, which in most cases did not get good results, obtained acceptable results in solving these types of functions.

Functions F_{18} to F_{26} are multimodal separable that have no dependence on conditions and parameters. The results of the SRS algorithm and other methods are compared in Table 19. In general, the SRS and MPA algorithms perform better than other methods in solving multimodal separable functions. According to the results presented in Table 20, these two algorithms have the same performance. The TLBO, RSA, and GWO algorithms are performed better than other methods in solving these problems. The results of all the algorithms listed in Table 19 for F_{18}, F_{19}, F_{20}, and F_{21} are almost the same. But the SD value of the SRS algorithm is less than other methods. Algorithms DMOA, SCA, GOA, PSO, and GA in solving the F_{22} could not achieve the global

Table 22
Wilcoxon Signed Ranks Test (high dimentional).

		SRS–GA	SRS–PSO	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–DMOA	SRS–RSA
F_{27}	Z	−4.933[a]	−1.000[a]	.000[c]	−4.782[a]	.000[c]	−1.000[a]	.000[c]	.000[c]	.000[c]
	Sig. (2-tailed)	.000	.317	1.000	.000	1.000	.317	1.000	1.000	1.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	**SRS = RSA**
F_{28}	Z	.000[c]	.000[c]	.000[c]	.000[c]	−1.732[a]	.000[c]	.000[c]	.000[c]	−3.834[a]
	Sig. (2-tailed)	1.000	1.000	1.000	1.000	.083	1.000	1.000	1.000	.000
	Comparison	**SRS = GA**	**SRS = PSO**	**SRS = TLBO**	**SRS = GOA**	SRS < SCA	**SRS = GWO**	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{29}	Z	−4.782[a]	.000[c]	.000[c]	−4.782[a]	.000[c]	.000[c]	.000[c]	.000[c]	.000[c]
	Sig. (2-tailed)	.000	1.000	1.000	.000	1.000	1.000	1.000	1.000	1.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	**SRS = SCA**	**SRS = GWO**	**SRS = MPA**	**SRS = DMOA**	**SRS = RSA**
F_{30}	Z	−4.623[a]	.000[c]	.000[c]	−4.782[a]	.000[c]	.000[c]	.000[c]	.000[c]	.000[c]
	Sig. (2-tailed)	.000	1.000	1.000	.000	1.000	1.000	1.000	1.000	1.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	**SRS = SCA**	**SRS = GWO**	**SRS = MPA**	**SRS = DMOA**	**SRS = RSA**
F_{31}	Z	−1.342[a]	.000[c]	.000[c]	−1.000[a]	−4.782[a]	−4.132[a]	.000[c]	.000[c]	−4.782[a]
	Sig. (2-tailed)	.180	1.000	1.000	.317	.000	.000	1.000	1.000	.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{32}	Z	.000[c]	.000[c]	.000[c]	.000[c]	−2.762[a]	−1.414[a]	.000[c]	.000[c]	−3.686[a]
	Sig. (2-tailed)	1.000	1.000	1.000	1.000	.006	.157	1.000	1.000	.000
	Comparison	**SRS = GA**	**SRS = PSO**	**SRS = TLBO**	**SRS = GOA**	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{33}	Z	−2.952[b]	−1.286[b]	−4.762[b]	−1.471[b]	−4.227[b]	−2.664[b]	−4.782[b]	−4.227[b]	−4.062[b]
	Sig. (2-tailed)	.003	.199	.000	.141	.000	.008	.000	1.000	.000
	Comparison	**SRS > GA**	**SRS > PSO**	**SRS > TLBO**	**SRS > GOA**	**SRS > SCA**	**SRS > GWO**	**SRS > MPA**	**SRS > DMOA**	**SRS > RSA**
F_{34}	Z	−3.097[a]	−3.858[a]	.000[c]	−3.769[a]	−4.782[a]	−4.784[a]	.000[c]	.000[c]	−5.477[a]
	Sig. (2-tailed)	.002	.000	1.000	.000	.000	.000	1.000	1.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{35}	Z	−2.989[a]	−2.947[a]	−1.000[a]	−3.420[a]	−4.782[a]	−4.785[a]	.000[c]	−1.000[a]	−5.477[a]
	Sig. (2-tailed)	.003	.003	.317	.001	.000	.000	1.000	0.317	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS < DMOA**	SRS < RSA
F_{36}	Z	−2.680[a]	−3.311[a]	.000[c]	−2.944[a]	−4.782[a]	−4.783[a]	.000[c]	.000[c]	−5.477[a]
	Sig. (2-tailed)	.007	.001	1.000	.003	.000	.000	1.000	1.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{37}	Z	−4.782[a]	−4.350[a]	−4.535[a]	−4.782[a]	−4.782[a]	−4.782[a]	-.360[b]	−4.515[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.719	0.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS > RSA**
F_{38}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	.000[c]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS = RSA**

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

optimal answer. But SRS achieved the optimal answer with the lowest SD. The PSO, MPA, and TLBO results are better for F_{23} than SRS. In solving F_{24}, GA, PSO, TLBO, GWO, and MPA algorithms had the same performance. But the DMOA algorithm got the best optimal answer. SRS is also more successful than other methods in solving F_{26}.

The results of the SRS algorithm and other metaheuristic methods for F_{27} to F_{50} are compared in (Tables 21 and 23). In general, due to the high power of exploration and exploitation of the SRS algorithm, it has been able to get the best results for F_{27} to F_{50}. The MPA and RSA algorithms also performed better in solving this category than other metaheuristic methods. Further details on the results are provided in (Tables 22 and 24).

4. Conclusions

In this study, a new metaheuristic algorithm called Special Relativity Search (SRS) was proposed to optimize optimization problems. The SRS algorithm was inspired by the concepts of electromagnetism and the application of the theory of Special Relativity physics. All the algorithms presented in the literature use Newtonian physics to develop the required equations. But in this study, for the first time, the equations were developed using Special relativity physics. In other words, in Special relativity physics, unlike Newtonian physics, the equations are based on the speed of light. The SRS utilizes two phenomena of length contraction and time dilation, plus Lorentz inverse transformations. This equation includes the position vector and the velocity vector in the magnetic field. Particles in the magnetic field are considered as optimal possible answers that move under the effect of the force known as the Lorentz force. Then the equations of motion are developed using the angular frequency. During the optimization of 83 functions that included three general categories of CEC2005, CEC2019, and 50 high dimensional, significant results were obtained. GA, PSO, GSA, and GOA do not perform well in solving problems with a local optimal. But they are better at solving problems with only a global optimal. The GOA algorithm requires more time to solve problems and its SD value is higher than other methods. But these methods worked well on some other problems and were able to discover the global optimal. MPA, GWO, and RSA methods performed better than the other methods. But even these methods could not solve the whole problem. The proposed SRS method also failed to solve all the problems. For example, CEC2005 includes four categories of problems that are a good option for testing the capabilities of metaheuristic algorithms. SRS easily solves unimodal functions that have only one global optimal. But it could not solve only 14% of the Basic Multimodal functions due to the huge optimal

Table 23
Results of mathematical test Functions.

	Fun		GA	PSO	TLBO	GOA	SCA	GWO	MPA	RSA	DMOA	SRS
Multimodal non-separable	F_{39}	Best	−3.8627	−3.8627	−3.8627	−3.8627	−3.8626	−3.8627	−3.8627	−3.8591	−3.8627	−3.8627
		Mean	−3.8627	−3.8627	−3.8627	−3.8112	−3.8554	−3.8617	−3.8627	−3.8266	−3.8627	−3.8627
		Median	−3.8627	−3.8627	−3.8627	−3.8627	−3.8546	−3.8627	−3.8627	−3.8325	−3.8627	−3.8627
		Worst	−3.8627	−3.8627	−3.8627	−3.0897	−3.8530	−3.8573	−3.8627	−3.7128	−3.8627	−3.8627
		SD	0.0000	0.0000	0.0000	0.1911	0.0024	0.0017	0.0000	0.0315	0.0000	0.0000
	F_{40}	Best	−3.3223	−3.3223	−3.3223	−3.3223	−3.1595	−3.3223	−3.3223	−3.0530	−3.3223	−3.3223
		Mean	−3.2945	−3.2596	−3.3141	−3.2575	−2.9438	−3.2480	−3.3223	−2.7237	−3.3223	−3.3223
		Median	−3.3223	−3.3223	−3.3223	−3.2031	−3.0108	−3.2031	−3.3223	−2.8352	−3.3223	−3.3223
		Worst	−3.2031	−3.0769	−3.2031	−3.1886	−1.9045	−3.0839	−3.3223	−1.4696	−3.3223	−3.3223
		SD	0.0512	0.0788	0.0295	0.0617	0.2547	0.0752	0.0000	−0.3758	0.0000	0.0000
	F_{41}	Best	0.6763	0.0000	0.0000	0.4651	0.0079	0.0000	0.0000	0.0000	0.9946	0.0000
		Mean	0.8903	0.0458	0.0000	0.7197	0.6658	0.0033	0.0000	0.0000	1.0469	0.0000
		Median	0.8858	0.0371	0.0000	0.7398	0.6542	0.0186	0.0000	0.0000	1.0449	0.0000
		Worst	1.0107	0.1941	0.0000	0.9914	1.1962	0.0186	0.0000	0.0000	1.0863	0.0000
		SD	0.0759	0.0445	0.0000	0.1381	0.3438	0.0059	0.0000	0.0000	0.0183	0.0000
	F_{42}	Best	0.2033	0.0175	0.0000	1.6081	0.0066	0.0000	0.0000	0.0000	2.2960	0.0000
		Mean	0.3202	5.1840	0.0000	3.2411	10.719	0.0000	0.0000	0.0000	3.2182	0.0000
		Median	0.3114	2.1784	0.0000	3.0515	13.353	0.0000	0.0000	0.0000	3.2045	0.0000
		Worst	0.4354	19.962	0.0000	5.0498	20.269	0.0000	0.0000	0.0000	4.9014	0.0000
		SD	0.0539	6.7867	0.0000	0.9214	9.6891	0.0000	0.0000	0.0000	0.5057	0.0000
	F_{43}	Best	0.0010	0.0080	0.0000	1.3691	0.6077	0.0063	0.0000	0.5187	281.0347	0.0000
		Mean	0.0028	6.5535	0.0000	6.5115	54 237	0.0283	0.0000	1.1530	6.33e+04	0.0000
		Median	0.0023	5.7787	0.0000	6.0898	4.4310	0.0264	0.0000	1.0725	3.26e+04	0.0000
		Worst	0.0092	13.840	0.0000	13.353	1.56e+06	0.0584	0.0000	1.6688	3.63e+05	0.0000
		SD	0.0016	3.4676	0.0000	3.1126	2.83e+05	0.0130	0.0000	0.3237	8.14e+04	0.0000
	F_{44}	Best	0.0314	2.4828	0.0000	0.1021	2.1909	0.0000	0.0000	0.0000	1.49e+04	0.0000
		Mean	0.0704	22.117	0.0467	15.614	3561.5	0.3827	0.0000	0.3800	2.44e+05	0.0000
		Median	0.0652	19.609	0.0000	10.197	20.377	0.4090	0.0000	0.0000	1.74e+05	0.0000
		Worst	0.1844	129.43	0.2990	54.132	6.04e+04	0.9229	0.0000	2.9000	7.40e+05	0.0000
		SD	0.0308	21.781	0.0774	15.692	1.27e+04	0.2428	0.0000	0.9859	1.99e+05	0.0000
	F_{45}	Best	−1.0809	−1.0809	−1.0809	−1.0809	−1.0809	−1.0809	−1.0809	−1.0786	−1.0809	−1.0809
		Mean	−1.0358	−1.0809	−1.0809	−1.0683	−1.0805	−1.0809	−1.0809	−1.0479	−1.0809	−1.0809
		Median	−1.0809	−1.0809	−1.0809	−1.0809	−1.0807	−1.0809	−1.0809	−1.0475	−1.0809	−1.0809
		Worst	−0.9455	−1.0809	−1.0809	−1.0052	−1.0791	−1.0809	−1.0809	−1.0139	−1.0809	−1.0809
		SD	0.0648	0.0000	0.0000	0.0286	0.0004	0.0000	0.0000	0.0214	0.0000	0.0000
	F_{46}	Best	−0.9390	−1.4999	−1.4999	−1.4999	−0.7884	−1.4999	−1.4999	−0.6894	−1.4999	−1.4999
		Mean	−0.6339	−10.297	−1.2807	−0.8338	−0.4876	−1.1526	−1.4286	−0.4035	−1.4999	−1.4821
		Median	−0.5436	−0.9398	−1.4999	−0.7976	−0.4888	−0.9649	−1.4999	−0.4503	−1.4999	−1.4999
		Worst	−0.2715	−0.4828	−0.9080	−0.1438	−0.1351	−0.5054	−0.9649	−0.0369	−1.4999	−0.9649
		SD	0.1632	0.3105	0.2739	0.42114	0.1555	0.3543	0.1849	0.1385	0.0000	0.0976
	F_{47}	Best	−0.4658	−0.7976	−1.5000	−0.7976	−0.0790	−1.4994	−1.5000	−0.0375	−1.2439	−0.0114
		Mean	−0.1871	−0.5296	−0.4876	−0.5185	−0.0371	−0.5193	−0.7046	−0.0084	−0.7931	−0.0007
		Median	−0.1668	−0.4727	−0.4472	−0.4727	−0.0413	−0.4796	−0.7976	−0.0036	−0.7976	−0.0000
		Worst	−0.0407	−0.2749	−0.2749	−0.1454	−0.0002	−0.2749	−0.4658	−0.0002	−0.4658	−0.0000
		SD	0.09814	0.2179	0.2621	0.2196	0.0221	0.2570	0.2079	0.0100	0.1379	0.2320
	F_{48}	Best	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0278	0.0000	0.0000
		Mean	0.0004	0.0000	0.0000	0.5226	0.0356	0.0003	0.0000	25.789	0.0000	0.0000
		Median	0.0000	0.0000	0.0000	0.0000	0.0177	0.0000	0.0000	0.5907	0.0000	0.0000
		Worst	0.0054	0.0000	0.0000	0.0015	0.1385	0.0021	0.0000	543.84	0.0000	0.0000
		SD	0.0011	0.0000	0.0000	0.0286	0.0393	0.0005	0.0000	99.412	0.0000	0.0000
	F_{49}	Best	0.0001	10.5394	0.0000	39.430	124.976	0.0881	0.0000	341.886	0.0629	0.0000
		Mean	424.51	1447.35	36.225	5598.9	624.520	341.13	0.0000	6532.61	312.0251	0.0000
		Median	263.33	59.0288	9.9720	2499.3	541.169	58.094	0.0000	3987.59	207.1712	0.0000
		Worst	4353.5	14 947.4	677.39	18 738	2163.78	4348.9	0.0000	20 423.9	1186.373	0.0000
		SD	800.83	3079.80	122.81	6339.7	514.211	1095.3	0.0000	6169.52	299.9507	0.0000
	F_{50}	Best	0.0191	10.5394	0.0000	0.0000	74.4697	0.1194	0.0000	544.138	0.0016	0.0000
		Mean	766.95	2465.77	45.120	2512.1	475.583	187.06	0.0000	5535.90	208.4601	0.0000
		Median	348.69	59.0288	10.539	677.39	424.079	59.062	0.0000	3601.10	222.9801	0.0000
		Worst	5074.7	14 947.4	692.45	14 947	1433.86	692.67	0.0000	22 028.2	839.2940	0.0000
		SD	1412.0	4050.49	124.31	4310.1	307.202	282.21	0.0000	4641.10	186.4152	0.0000

answers. Successfully solved expanded multimodal functions that included two functions. And failed to solve 33% of the last class of CEC2005 problems that were a combination of the previous three classes. The CEC2019 consists of 10 functions, three of which are very hard. Functions F_7, F_8, and F_9 which multimodal non-separable. Because SRS showed a good performance in optimizing multimodal non-separable functions from CEC2005, it optimized these three hard functions better than other methods. The SRS algorithm also solved non-separable multimodal functions of 50 high dimensional benchmark functions better than other functions. This indicates that the overall performance of SRS in solving a total of 83 functions has been better than other metaheuristic methods. But for future work, the weaknesses of this algorithm can be addressed to have a better performance in solving different types of problems. This algorithm can also be developed to solve constraint engineering problems, which will be investigated in future work.

Table 24
Wilcoxon Signed Ranks Test (high dimentional).

		SRS–GA	SRS–PSO	SRS–TLBO	SRS–GOA	SRS–SCA	SRS–GWO	SRS–MPA	SRS–DMOA	SRS–RSA
F_{39}	Z	.000[c]	.000[c]	.000[c]	−2.701[a]	−4.786[a]	−3.305[a]	.000[c]	.000[c]	−4.782[a]
	Sig. (2-tailed)	1.000	1.000	1.000	.007	.000	.001	1.000	1.000	.000
	Comparison	**SRS = GA**	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{40}	Z	−2.646[a]	−3.237[a]	−2.366[a]	−3.519[a]	−4.782[a]	−3.627[a]	.000[c]	.000[c]	−4.782[a]
	Sig. (2-tailed)	.008	.001	.018	.000	.000	.000	1.000	1.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{41}	Z	−4.782[a]	−4.782[a]	−4.782[b]	−4.782[a]	−4.782[a]	−2.521[a]	.000[c]	−4.782[a]	.000[c]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.012	1.000	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	**SRS = MPA**	SRS < DMOA	**SRS = RSA**
F_{42}	Z	−4.782[a]	−4.782[a]	−4.949[a]	−4.782[a]	−4.782[a]	−4.834[a]	−4.782[a]	−4.782[a]	.000[c]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	1.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS = RSA**
F_{43}	Z	−4.782[a]	−4.782[a]	−2.890[b]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.004	.000	.000	.000	.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{44}	Z	−4.782[a]	−4.782[a]	−4.371[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−2.437[b]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.015
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	**SRS > RSA**
F_{45}	Z	−2.970[a]	.000[c]	.000[c]	−2.236[a]	−4.389[a]	.000[b]	.000[c]	.000[c]	−4.782[a]
	Sig. (2-tailed)	.003	1.000	1.000	.025	.000	1.000	1.000	1.000	.000
	Comparison	SRS < GA	**SRS = PSO**	**SRS = TLBO**	SRS < GOA	SRS < SCA	**SRS = GWO**	**SRS = MPA**	**SRS = DMOA**	SRS < RSA
F_{46}	Z	−4.787[a]	−4.098[a]	−3.084[a]	−4.624[a]	−4.782[a]	−3.924[a]	−1.732[a]	−1.000[b]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.002	.000	.000	.000	.083	0.317	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS > DMOA**	SRS < RSA
F_{47}	Z	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.659[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.371[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{48}	Z	−4.782[a]	−2.849[b]	−3.692[b]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	.000[c]	−4.782[a]
	Sig. (2-tailed)	.000	.004	.000	.000	.000	.000	.000	1.000	.000
	Comparison	SRS < GA	**SRS > PSO**	**SRS > TLBO**	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	**SRS = DMOA**	SRS < RSA
F_{49}	Z	−4.782[a]	−4.807[a]	−4.250[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA
F_{50}	Z	−4.782[a]	−4.798[a]	−4.585[a]	−4.786[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]	−4.782[a]
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	0.000	.000
	Comparison	SRS < GA	SRS < PSO	SRS < TLBO	SRS < GOA	SRS < SCA	SRS < GWO	SRS < MPA	SRS < DMOA	SRS < RSA

[a]Based on positive ranks (SRS is winner).
[b]Based on negative ranks (SRS is failure).
[c]The sum of ranks is equal.

CRediT authorship contribution statement

Vahid Goodarzimehr: Conceptualization, Methodology, Software, Writing – original draft, Investigation, Writing – review & editing. **Saeed Shojaee:** Conceptualization, Methodology, Visualization, Investigation, Supervision, Validation, Writing – review & editing. **Saleh Hamzehei-Javaran:** Visualization, Investigation, Supervision, Validation. **Siamak Talatahari:** Software, Visualization, Supervision, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Horst, H. Tuy, Global Optimization: Deterministic Approaches, Springer Science & Business Media, 2013.
[2] V. Hajipour, A. Kheirkhah, M. Tavana, N. Absi, Novel pareto-based meta-heuristics for solving multi-objective multi-item capacitated lot-sizing problems, Int. J. Adv. Manuf. Technol. 80 (1) (2015) 31–45.
[3] S.S. Mirjalili, M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[4] L. Abualigah, Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications, Neural Comput. Appl. 32 (2020) 12381–12401.
[5] A.H. Gandomi, X.She. Yang, S. Talatahari, A.H. Alavi, Metaheuristic Applications in Structures and Infrastructures, Elsevier, 2013.
[6] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.
[7] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: overview and con- ceptual comparison, ACM Comput. Surv. 35 (2003) 268–308.
[8] El-Ghazali Talbi, Metaheuristics: From Design to Implementation, Wiley, 2009.
[9] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[10] K.V. Price, N.H. Awad, M.Z. Ali, P.N. Suganthan, The 100-Digit Challenge: Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization, Technical Report, Nanyang Technological University, Singapore, 2018.
[11] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[12] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[13] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[14] E. Bogar, S. Beyhan, Adolescent Identity Search Algorithm (AISA): A novel metaheuristic approach for solving optimization problems, Appl. Soft Comput. 95 (2020) 1568–4946.
[15] J.H. Holland, Adaptation in Natural and Artificial Systems, second ed., University of Michigan Press, Ann Arbor, MI, 1975, 1992.
[16] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[17] I. Rechenberg, Evolutions Strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, 1973.

[18] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[19] J.A. Lozano, L. Pedro, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic, 2002.
[20] D. Dasgupta, Advances in artificial immune systems, IEEE Comput. Intell. Mag. 1 (2006) 40–49.
[21] P. Moscato, C. Cotta, A. Mendes, Memetic algorithms, in: New Optimization Techniques in Engineering, in: Studies in Fuzziness and Soft Computing, vol. 141, Springer, Berlin, Heidelberg, 2004.
[22] M.M. Motevali, A. Mohammadi Shanghooshabad, R. Zohouri Aram, H. Keshavarz, Who: A new evolutionary algorithm bio-inspired by wildebeests with a case study on bank customer segmentation, Int. J. Pattern Recognit. Artif. Intell. 33 (05) (2019) 1959017.
[23] A. Cheraghalipour, M. Hajiaghaei-Keshteli, M.M. Paydar, Tree growth algorithm (tga): A novel approach for solving optimization problems, Eng. Appl. Artif. Intell. 72 (2018) 393–414.
[24] N.S. Jaddi, J. Alvankarian, S. Abdullah, Kidney-inspired algorithm for optimization problems, Commun. Nonlinear Sci. Numer. Simul. 42 (2017) 358–369.
[25] M.H. Salmani, K. Eshghi, A metaheuristic algorithm based on chemotherapy science: Csa, J. Optim. 2017 (2017).
[26] X. Xu, Z. Hu, Q. Su, Y. Li, J. Dai, Multivariable grey prediction evolution algorithm: A new metaheuristic, Appl. Soft Comput. 89 (2020) 106086.
[27] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing., Science 220 (1983) 671–680.
[28] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Gsa: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[29] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[30] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289.
[31] O.K. Erol, I. Eksin, A new optimization method: Big bang–big crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.
[32] A.Y.S. Lam, V.O.K. Li, Chemical-reaction-inspired metaheuristic for optimization, IEEE Trans. Evol. Comput. 14 (3) (2010) 381–399.
[33] A. Kaveh, M. Khayatazad, A new meta-heuristic method: Ray optimization, Comput. Struct. 112–113 (2012) 283–294.
[34] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[35] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[36] B. Javidy, A. Hatamlou, S. Mirjalili, Ions motion algorithm for solving optimization problems, Appl. Soft Comput. 32 (2015) 72–79.
[37] B. Doğan, T. Ölmez, A new metaheuristic for numerical function optimization: Vortex search algorithm, Inform. Sci. 293 (2015) 125–145.
[38] M. Azizi, Atomic orbital search: A novel metaheuristic algorithm, Appl. Math. Model. 93 (2021) 657–683.
[39] S. Talatahari, M. Azizi, M. Tolouei, B. Talatahari, P. Sareh, Crystal structure algorithm (CryStAl): A metaheuristic optimization method, IEEE Access 9 (2021) 71244–71261.
[40] S. Talatahari, M. Azizi, Chaos game optimization: a novel metaheuristic algorithm, Artif. Intell. Rev. 54 (2021) 917–1004.
[41] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, Inform. Sci. 540 (2020) 131–159.
[42] S. Gholizadeh, M. Danesh, C. Gheyratmand, A new Newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames, Comput. Struct. 234 (2020) 106250.
[43] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.
[44] J.L. Junho Pereira, M.B. Francisco, C.A. Diniz, G. Antônio Oliver, Jr S. Simoes Cunha, G.F. Gomes, Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization, Expert Syst. Appl. 170 (2021) 114522.
[45] S. Mirjalili, SCA: A Sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[46] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948.
[47] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[48] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1) (1996) 29–41.
[49] X.-S. Yang, Firefly algorithms for multimodal optimization, in: O. Watanabe, T. Zeugmann (Eds.), Stochastic Algorithms: Foundations and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 169–178.
[50] X. Yang, Cuckoo search via levy flights, in: 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), 2009, pp. 210–214.
[51] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
[52] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[53] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[54] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 65–74.
[55] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[56] W.-T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, Knowl.-Based Syst. 26 (2012) 69–74.
[57] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[58] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (aaa) for nonlinear global optimization, Appl. Soft Comput. 31 (2015) 153–171.
[59] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[60] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50.
[61] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[62] Malik Braik, Abdelaziz Hammouri, Jaffar Atwan, Mohammed Azmi Al-Betar, Mohammed A. Awadallah, White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems, Knowl.-Based Syst. 243 (2022) 108457.
[63] Benyamin Abdollahzadeh, Farhad Soleimanian Gharehchopogh, Seyedali Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021) 107408.
[64] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: New metaheuristic algorithm for solving optimization problems, Math. Comput. Simulation 192 (2022) 84–110.
[65] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020) 103731.
[66] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A.H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.
[67] M.S. Braik, Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Syst. Appl. 174 (2021) 114685.
[68] M. Braik, A. Sheta, H. Al-Hiary, A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm, Neural Comput. Appl. (2020) 1–33.
[69] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci. Control Eng. 8 (1) (2020) 22–34.
[70] Gaurav Dhiman, Vijay Kumar, Spotted hyena optimizer: a novel bioinspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[71] S. Shadravan, H.R. Naji, V.K. Bardsiri, The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Eng. Appl. Artif. Intell. 80 (2019) 20–34.
[72] O.N. Oyelade, A.E.-S. Ezugwu, T.I.A. Mohamed, L. Abualigah, Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm, IEEE Access 10 (2022) 16150–16177.
[73] Jeffrey O. Agushaka, Absalom E. Ezugwu, Laith Abualigah, Dwarf mongoose optimization algorithm, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114570.
[74] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: A nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020) 113377.
[75] L. Abualigah, M.A. Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022) 116158.
[76] R. Rao, V. Savsani, D. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.

[77] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population-based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[78] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, The social engineering optimizer (seo), Eng. Appl. Artif. Intell. 72 (2018) 267–293.
[79] S.-A. Ahmadi, Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems, Neural Comput. Appl. 28 (1) (2017) 233–244.
[80] R. Moghdani, K. Salimifard, Volleyball premier league algorithm, Appl. Soft Comput. 64 (2018) 161–185.
[81] N. Ghorbani, E. Babaei, Exchange market algorithm, Appl. Soft Comput. 19 (2014) 177–187.
[82] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. 13 (5) (1986) 533–549.
[83] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, Ieee, 2007, pp. 4661–4667.
[84] S. Massan, A.I. Wagan, M.M. Shaikh, A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem, Appl. Soft Comput. 90 (2020) 106176.
[85] Malik Braik, Mohammad Hashem Ryalat, Hussein Al-Zoubi, A novel metaheuristic algorithm for solving numerical optimization problems: Ali baba and the forty thieves, Neural Comput. Appl. (2021) 1–47.
[86] Mohammed Azmi Al-Betar, Zaid Abdi Alkareem Alyasseri, Mohammed A. Awadallah, Iyad Abu Doush, Coronavirus herd immunity optimizer (chio), Neural Comput. Appl. 33 (10) (2021) 5011–5042.
[87] A.W. Mohamed, A.A. Hadi, A. Khater Mohamed, Gaining-sharing knowledge-based algorithm for solving optimization problems: a novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern. 11 (7) (2020) 1501–1529.
[88] S.J. Mousavirad, H. Ebrahimpour-Komleh, Human mental search: a new population-based metaheuristic optimization algorithm, Appl. Intell. 47 (3) (2017) 850–887.
[89] H. Ghasemian, F. Ghasemian, H. Vahdat-Nejad, Human urbanization algorithm: A novel metaheuristic approach, Math. Comput. Simulation 178 (2020) 1–15.
[90] Z. Feng, W. Niu, S. Liu, Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems, Appl. Soft Comput. 98 (2021) 106734.
[91] S. Talatahari, H. Bayzidi, M. Saraee, Social network search for global optimization, IEEE Access 9 (2021) 92815–92863.
[92] Y. Zhang, Z. Jin, Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems, Expert Syst. Appl. 148 (2020) 113246.
[93] L. Yongliang, B. Una, W. Qinghua, A highly effective hybrid evolutionary algorithm for the covering salesman problem, Inform. Sci. 564 (2021) 144–162.
[94] R.J. Kuo, Cheng-Kang Chen, Shao-Hong Keng, Application of hybrid metaheuristic with perturbation-based K-nearest neighbores algorithm and densest imputation to collaborative filtering in recommender systems, Inform. Sci. 575 (2021) 90–115.
[95] P. Jeng-Shyang, L. Nengxian, C. Shu-Chuan, L. Taotao, An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems, Inform. Sci. 561 (2021) 304–325.
[96] N. Kumar, A.A. Shaikh, S. Kumar Mahato, A. Kumar Bhunia, Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations, Expert Syst. Appl. 172 (2021) 0957–4174.
[97] S. Mirjalili, A.H. Gandomi, Chaotic gravitational constants for the gravitational search algorithm, Appl. Soft Comput. 53 (2017) 407–419.
[98] M. Oszust, Enhanced marine predators algorithm with local escaping operator for global optimization, Knowl.-Based Syst. 232 (2021) 107467.
[99] T. Weidner, Richard, L. Sells, Robert, Elementary Modern Physics, Allyn and Bacon, Boston, 1973.
[100] D. Halliday, R. Resnick, J. Walker, Fundamentals of Physics, eighth ed., Wiley, New York, 2008.