



Artificial Electric Field Algorithm with Greedy State Transition Strategy for Spherical Multiple Traveling Salesmen Problem

Jian Bi¹ · Guo Zhou² · Yongquan Zhou^{1,3} · Qifang Luo^{1,3} · Wu Deng⁴

Received: 15 September 2021 / Accepted: 20 December 2021
© The Author(s) 2021

Abstract

The multiple traveling salesman problem (MTSP) is an extension of the traveling salesman problem (TSP). It is found that the MTSP problem on a three-dimensional sphere has more research value. In a spherical space, each city is located on the surface of the Earth. To solve this problem, an integer-serialized coding and decoding scheme was adopted, and artificial electric field algorithm (AEFA) was mixed with greedy strategy and state transition strategy, and an artificial electric field algorithm based on greedy state transition strategy (GSTAEFA) was proposed. Greedy state transition strategy provides state transition interference for AEFA, increases the diversity of population, and effectively improves the accuracy of the algorithm. Finally, we test the performance of GSTAEFA by optimizing examples with different numbers of cities. Experimental results show that GSTAEFA has better performance in solving SMTSP problems than other swarm intelligence algorithms.

Keywords Artificial electric field algorithm · State transfer strategy · Spherical geometry · Spherical multiple traveling salesman problem · Metaheuristic optimization

1 Introduction

Combinatorial optimization problem is a kind of optimization problem. Optimization problems can be divided into two categories: one is continuous variable problems, the other is discrete variable problems. Problems with discrete variables are called combinatorial. In the case of continuous variables, it is generally a set of real numbers, or a function; In combinatorial problems, an object is found from an infinite or countable infinite set—typically an integer, a set, a permutation, or a graph. In general, these two types of problems have quite different characteristics, and the methods of solving them are quite different. Metaheuristic

algorithms have been widely studied in recent years because of their advantages in multi-modal, large-scale, and highly constrained problems.

The traveling salesman problem (TSP) is a classical combinatorial optimization problem [1]. Many practical problems can be modeled as TSP, such as route planning [2], production scheduling [3], and emergency management [4]. In recent years, there are many scholars have made continuous exploration on solving TSP in two-dimensional space. There are some precise algorithms and heuristic algorithms that can effectively solve TSP. Gouveia, Leitner, and Ruthmair used a branch cutting algorithm [5]. Kinable, Smeulders, Delcour, and Spieksma used an accurate algorithm [6]. Asadpour et al. used Approximation Algorithm [7]. Both Genetic Algorithm [8] and Memetic Algorithm [9] are applied to asymmetric traveling salesman problem. Algorithms to solve the TSP problem also include Evolutionary Algorithm [10], Artificial Bee Colony Algorithm [11], Ant Colony Algorithm [12], Water Cycle Algorithm [13], Discrete Bat Algorithm [14], Random-key Cuckoo Search [15], Black Hole Algorithm [16], Discrete Comprehensive Learning Particle Swarm Optimization Algorithm [17], Simulated Annealing Algorithm [18], Fruit-fly Optimization Algorithm [19], and so on.

✉ Yongquan Zhou
yongquanzhou@126.com

¹ College of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, China
² Department of Science and Technology Teaching, China University of Political Science and Law, Beijing 100088, China
³ Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning 530006, China
⁴ College of Electronic Information and Automation, Civil Aviation University of China, Tianjin, Hebei 300300, China

However, the classic TSP cannot be modeled in some special cases. For example, suppose that a company has multiple salespeople living in different cities, and the company expects them to visit the city no less than a certain number of times to meet their minimum wage [20]. Therefore, the MTSP is introduced on the basis of the current problem [21]. ACO has been widely used in the research field of MTSP and achieved good results. In particular, Ghafurian and Javadian [22] designed an ant-based system to solve the MTSP problem at a fixed destination. Yousefkhoshbakht et al. [23] solved the MTSP problem by mixing the insertion, switching, and 2-opt operators with ant colony algorithm. Changdar et al. [24] proposed a genetic ACO algorithm for solving MTSP. However, it is difficult to optimize the parameters of ant colony optimization system because of the large number of inherent parameters. This further leads to its disadvantages such as slow computation speed and easy local convergence. When more complex problems are solved, unacceptable performance occurs. Other metaheuristic algorithms have been proposed to solve MTSP. Malmborg [25] proposed a genetic algorithm based on two chromosomes. Carter and Ragsdale [26] designed a genetic algorithm for the two-part chromosome technique. Venkatesh et al. [27] mixed artificial bee colony algorithm with weed invasion algorithm. Chen et al. [28] proposed an improved two-part Wolf pack search algorithm. Zhou et al. [29] proposed a improved PSO to solve MTSP.

Compared to the standard TSP, the spherical space TSP is more practical. In recent years, many scholars have studied TSP in three-dimensional space. Literature [30] solves 3D-TSP with multi-dimensional city location. Literature [31] adopts genetic algorithm to solve the TSP problem on the sphere. Literature [32] describes the distance distribution between random points on a sphere. The ant colony algorithm [33], firefly algorithm [34], an improved flower pollination algorithm based on greedy strategy [35], and the discrete improved cuckoo algorithm [36] are used to solve STSP. As we know, the surface of the earth on which we live is very close to a sphere. In many fields of study, such as chemistry, biology, and physics, structures such as atoms, molecules, and proteins are represented by spheres. Therefore, the solution of spherical MTSP can be directly applied to micro and macro systems such as path planning, robot path planning, picking, and placing parts. Therefore, it is of great significance to study spherical MTSP problem.

AEFA is a physics-based optimization algorithm designed by Yadav et al. in 2019 to solve continuous optimization problems. Its main characteristics include less optimization parameters, rapid convergence in the process of optimization, high accuracy, and low complexity of the algorithm [37]. Since its inception, AEFA has been widely used in continuous optimization problems [38], practical engineering problems [39], image matching problems [40],

multi-objective optimization [41], nine parameters triode PV model best estimate [42], cancer detection [43], white blood cells [44], the optimal purchasing problem [45], feature classification [46], assembly line balancing problem [47], and wind turbines active loss in distribution network and the distribution of the voltage deviation problems [48] made a wide range of applications. In this paper, an improved AEFA is proposed. The greedy state transition strategy is introduced on the basis of AEFA, and the improved algorithm is used to solve SMTSP. Similar to STSP, the city coordinates of the SMTSP are extended to the surface of a sphere, and the coordinates of the points and the minimum spherical distance between the points are known. The improved AEFA is used to solve SMTSP. Experiments show that the proposed algorithm is feasible and effective in solving SMTSP.

The rest of this paper is organized as follows: “[Spherical Geometry and Definition of the Problem](#)” gives a brief introduction to spherical geometry and the definition of the problem. “[Related Works](#)” introduces the related work, including solution encoding and decoding, greedy state transfer strategy, AEFA, and GSTAEFA. In “[Experimental Setup and Discussion of Results](#),” the simulation experiment and numerical results are analyzed. Finally, “[Conclusions and Further Works](#)” gives summary and future prospects.

2 Spherical Geometry and Definition of the Problem

Figure 1 shows a sphere. The radius r of the sphere is the distance from the point on the sphere to the center. In three-dimensional space, the coordinates of each point on the sphere need to satisfy the relationship as shown in Eq. (1):

$$x^2 + y^2 + z^2 = r^2, \quad (1)$$

where r is the distance from a point on the sphere to the center, and it is the radius of the sphere, (x, y, z) are the coordinates of the points on the sphere in the three dimensions [34] (Fig. 2).

2.1 Representation of Points on a Sphere

Points on a sphere can be represented in cartesian coordinates so that the shortest distance between two points on the sphere can be calculated using vector methods. The Cartesian vector point function can represent that point [35]:

$$P(u) = (x(u, v), y(u, v), z(u, v)). \quad (2)$$

In general, the standard equation for a sphere can be expressed in parametric form, and a spherical coordinate equation can be expressed with two parameters u and v

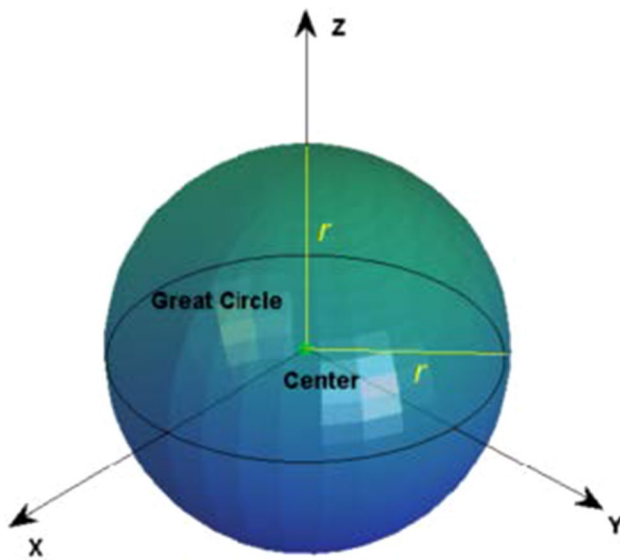


Fig. 1 Sphere

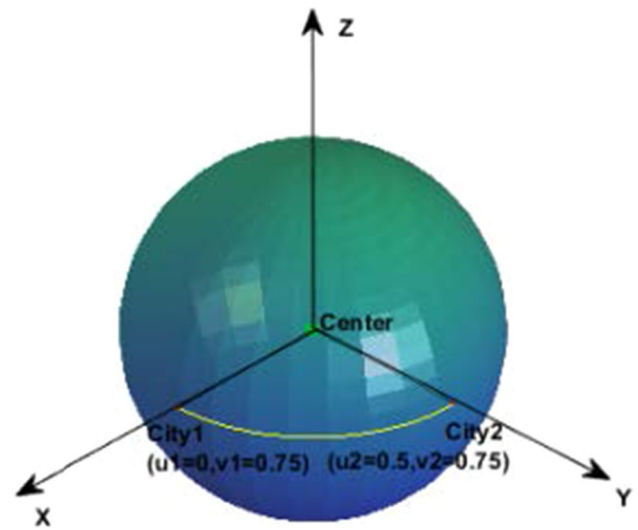


Fig. 3 The u and v of different cities

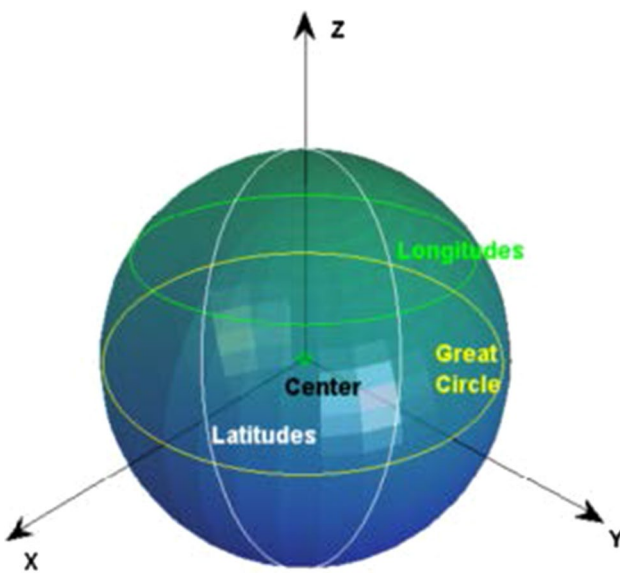


Fig. 2 Latitude and longitude on the sphere

between 0 and 1. A sphere of radius r with its center at the $(0,0,0)$ can be expressed as [36]

$$x(u, v) = r \cdot \cos(2\pi u) \cdot \sin(\pi v) \tag{3}$$

$$y(u, v) = r \cdot \sin(2\pi u) \cdot \sin(\pi v) \tag{4}$$

$$z(u, v) = r \cdot \cos(\pi v), \tag{5}$$

where the two parameters u and v represent the longitude and latitude lines, respectively. In this paper, we use a sphere

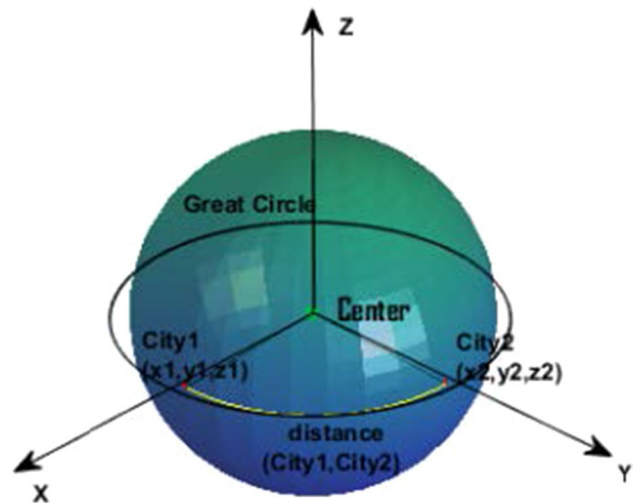


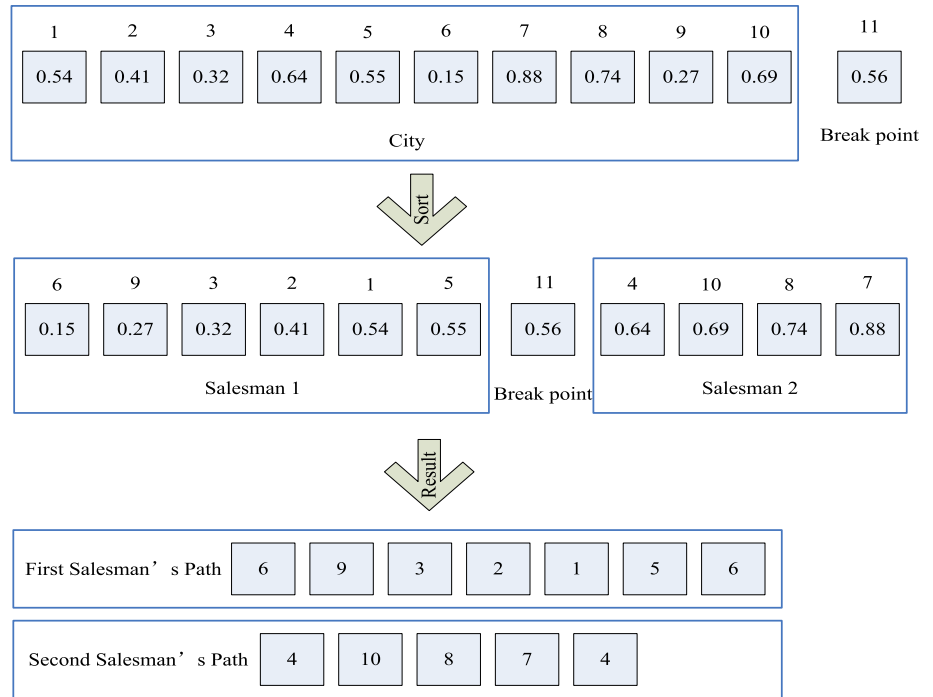
Fig. 4 The minimum distance between two cities

with radius 1 for simulation. Therefore, it is easy to calculate the length of the path between two cities (Fig. 3).

2.2 Shortest Path Between Two Cities on Sphere

On the surface of the earth, a geodesic is the curve that shows the shortest distance between two cities. The point at which the tangent plane through the center intersects the earth forms a great circle. The length of the shortest line between two points on the sphere is the length of the lower arc between two points in the great circle that passes through them [34]. In Fig. 4, there are two cities City1 and City2, in order to obtain the shortest distance between them. Take the scalar product of two vectors [34]:

Fig. 5 Encoding and decoding processes



$$\vec{v}_1 \cdot \vec{v}_2 = |\vec{v}_1| |\vec{v}_2| \cos(\theta). \tag{6}$$

The smaller angle between v_1 and v_2 is θ . Scalar values are computed as follows [34]:

$$\vec{v}_1 \cdot \vec{v}_2 = x_1x_2 + y_1y_2 + z_1z_2. \tag{7}$$

The formula for the shortest distance between City1 and City2 on the sphere is [31]

$$\hat{d}_{City_1, City_2} = r\theta. \tag{8}$$

Combining formulas (6) to (8), the distance formula is deduced as follows [31]:

$$\hat{d}_{City_1, City_2} = r \arccos \left(\frac{x_1x_2 + y_1y_2 + z_1z_2}{r^2} \right). \tag{9}$$

2.3 Definition of the Problem

In MTSP, there are multiple traveling agents visiting different cities, and each city can only be visited once and only by one person. Given a set of n cities that a salesperson will visit, TSP looks for the shortest possible trip for the salesperson, which visits each city only once. MTSP can be briefly described as follows: Give an undirected graph $G=(V, A)$, It consists of a set of vertices V and a set of arcs A , where m represents the total number of salespeople. The objective function is to divide V into m nonempty subsets $\{S_i\}_{i=1}^m$ and

calculate the minimum cost sum of all sets. Then the objective function value of MTSP can be obtained [49] as follows:

$$\text{Minimize } \sum_{i=1}^m \left(x_{n^i,1}^i + \sum_{j=1}^{n^i-1} x_{j,j+1}^i \right), \tag{10}$$

where the first sum represents the path through m salespeople, and the second sum represents the cycle of all the cities visited by the i th salesperson (the index of the first city visited by the i th salesperson is 1, and the index of the last city is n^i); $x_{j,j+1}^i$ represents the distance between the city j visited by the i th salesperson and $j + 1$; $x_{n^i,1}^i$ represents the distance between the last city visited by the i th salesperson and the first city; n^i value should not be less than the minimum number of cities specified by each salesperson [49].

3 Related Works

3.1 Motivation

At present, for many practical application problems such as TSP problem, MST problem, MTSP problem, people cannot effectively obtain the optimal solution in a reasonable time range. Therefore, metaheuristic optimization algorithms are being applied by more and more researchers from various fields to obtain satisfactory solutions such as hybrid PSO–GA algorithm for constrained optimization problems [61], a hybrid GSA–GA algorithm for constrained optimization problems [59], GA–GSA hybrid

Fig. 6 Swap operator

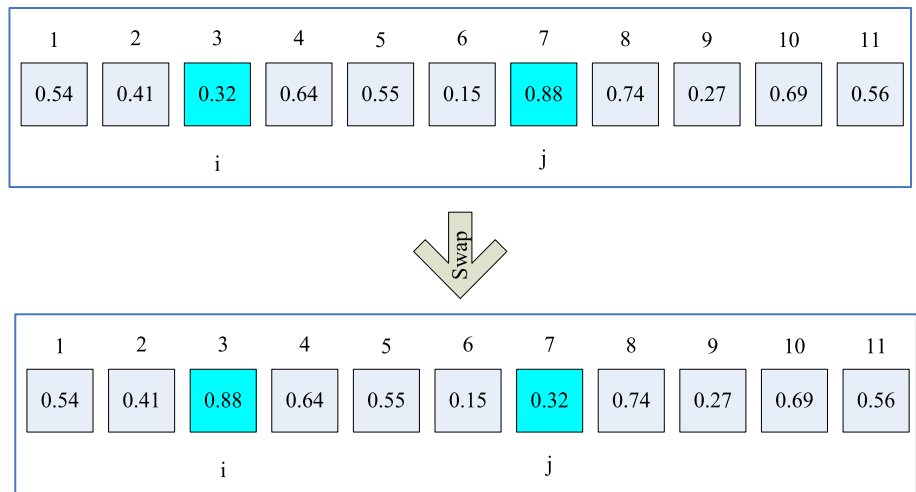


Fig. 7 Reverse operator

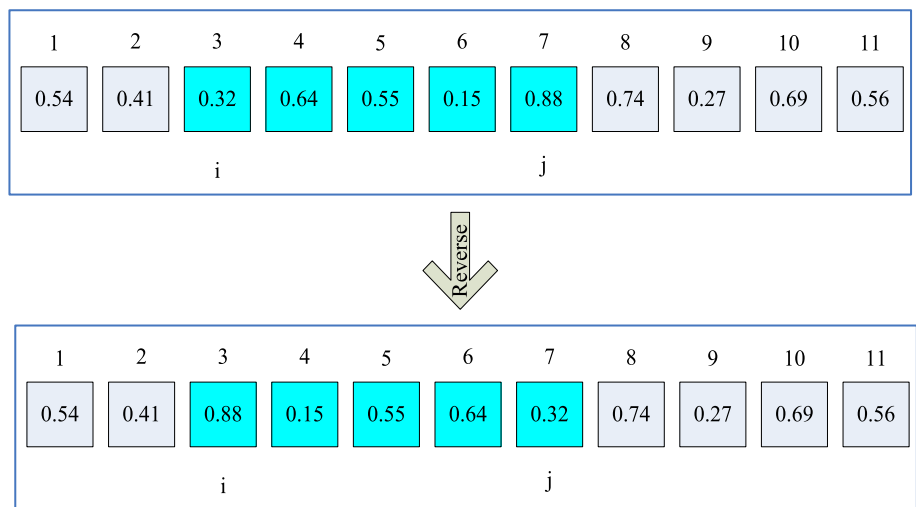


Fig. 8 Insert operator

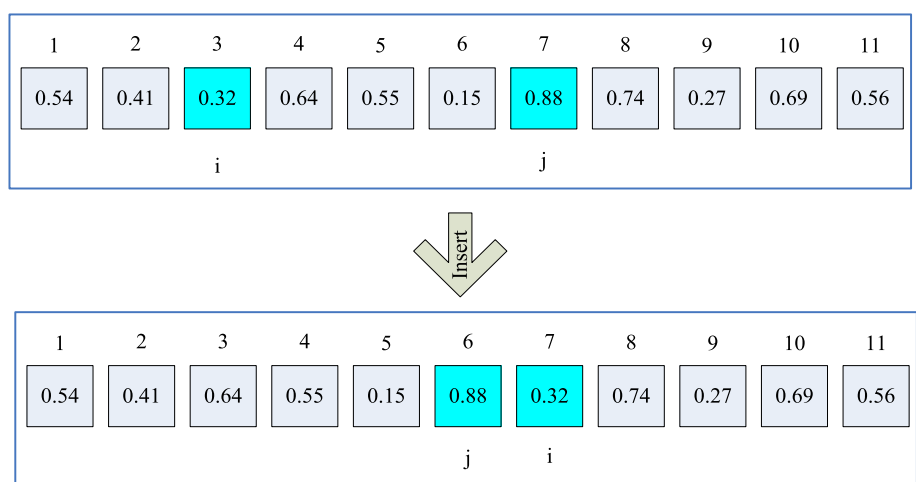
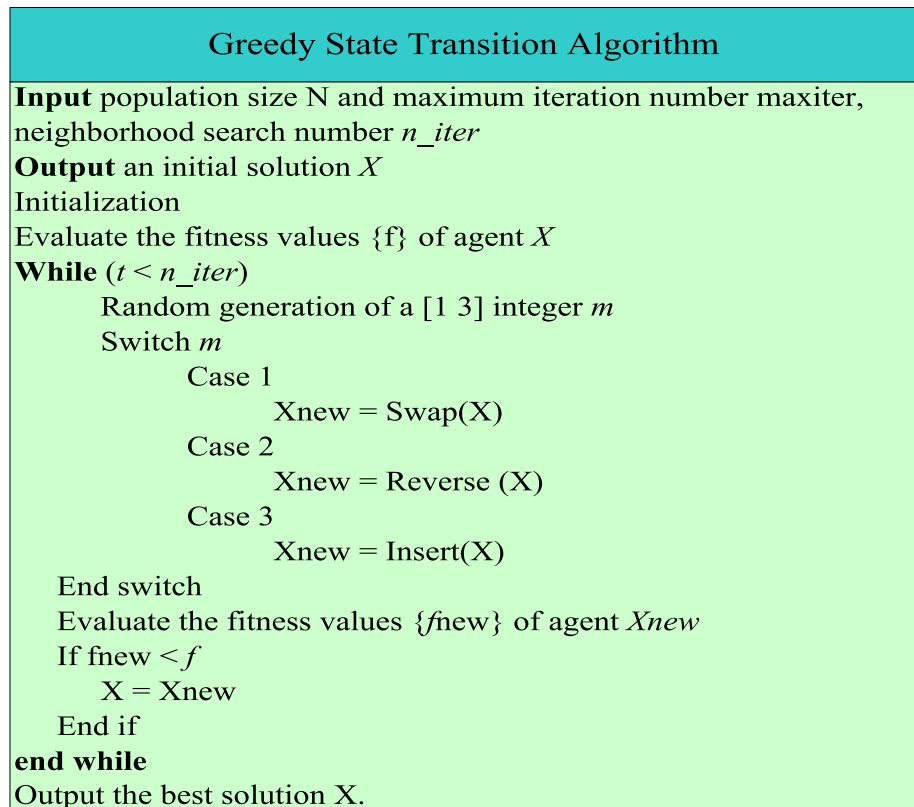


Fig. 9 Proposed greedy state transition algorithm



algorithm for optimizing the performance of industrial systems using uncertain data [60], and a hybrid ITL-HHO algorithm for solving numerical and engineering optimization problems [62]. This paper mainly studies the improvement of artificial electric field algorithm and its application in practical optimization problems. The motive of the improvement is to improve the mining capacity and exploration capacity of the original algorithm, respectively, so as to further improve the overall optimization ability of the algorithm. Artificial electric field algorithm is a new intelligent optimization algorithm based on physical laws proposed in 2019, so its improvement and application is relatively small. Therefore, the purpose of improving the algorithm is to solve the practical optimization problem on one hand, on the other hand, to verify the optimization ability of the artificial electric field algorithm, overcome the shortcomings of the original algorithm, so as to broaden the application scope of the artificial electric field algorithm in practical life.

3.2 Solution Encoding and Decoding

Sequence coding is the simplest and most efficient way to express MTSP solutions, using a series of numbers to represent each solution. There are two common approaches to using the sequential encoding approach to represent an

MTSP solution. One uses a path sequence and a breakpoint sequence, and the other uses a path sequence and a city number sequence. Here, we adopt the real encoding method, which is suitable for most continuous algorithms to solve the MTSP problem. The following example shows how the real-coded approach represents an MTSP solution. For example, suppose two salespeople have to visit 10 cities.

We use a single sequence coding scheme to represent a feasible solution. Compared with double sequence coding, the calculation is simpler and the processing is more convenient. [0.54 0.41 0.32 0.64 0.55 0.15 0.88 0.74 0.27 0.69 0.56] can be described as a feasible solution. After sorting it from small to large, the corresponding sequence is [6 9 3 2 1 5 11 4 10 8 7], and number 11 is the breakpoint. The sequence can be divided into two paths, which are [6 9 3 2 1 5 6] and [4 10 8 7 4] (Fig. 5).

3.3 Greedy State Transition Strategy

The state transition algorithm (STA) [50], a new heuristic search algorithm proposed by Zhou et al. [51], has outstanding performance in solving optimization problems and is widely used in many practical problems [52]. In this paper, we introduce three kinds of state transfer operators, and combine them to propose a state transfer strategy. The three operators are Swap, Reverse, and Insert of three basic transformations.

Fig. 10 AEFA

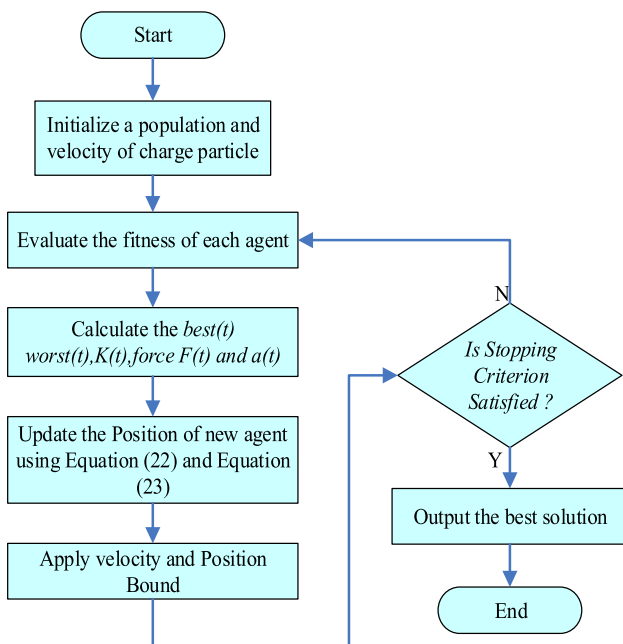
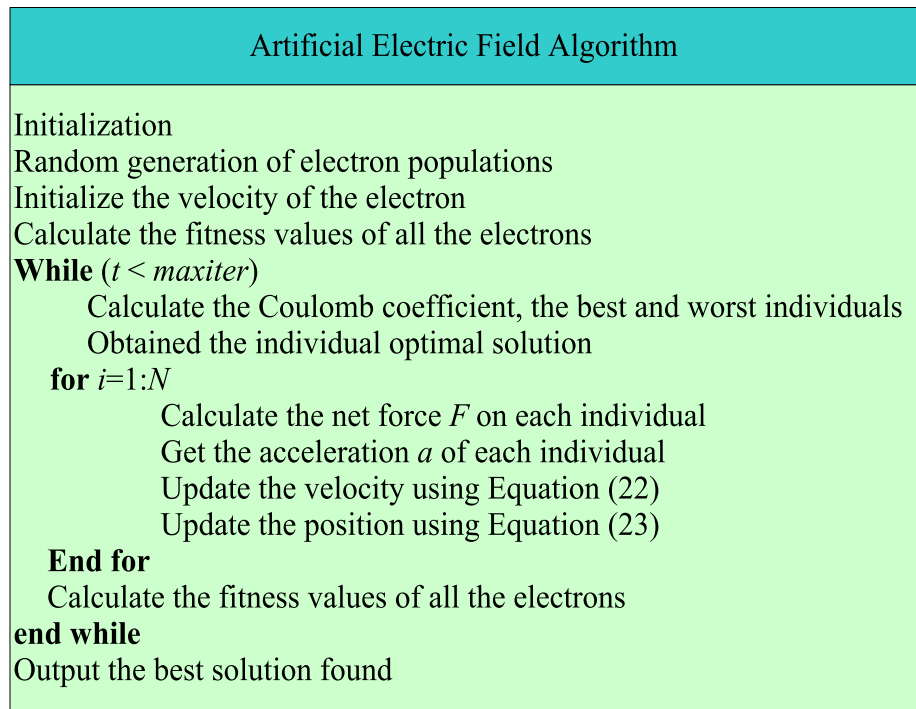


Fig. 11 AEFA flowchart

Greedy strategies always do what is best for the moment. In other words, instead of considering global optimality, the algorithm gets a local optimal solution in a certain sense. In the state transition strategy, we do not consider globally. After any operator produces a new solution, if the result is better, we accept the local change, so that the algorithm

will keep getting closer to the global optimal solution. The MTSP problem is a combinatorial optimization problem, and the solution form is a sequence. Every state transfer will generate a feasible solution. Greedy thought is introduced, assuming that an optimal sequence can be obtained without considering the time complexity.

Swap Operator: Pick two positions i and j at random from the initial solution, and the corresponding values of i and j are exchanged.

$$x_{new} = Swap(x_{i,j}), \tag{11}$$

where x_{new} is the immediate solution of x generated, i and j are two different positional indexes in x , respectively. $Swap(x_{i,j})$ swaps the values of the positions corresponding to i and j (Fig. 6).

Reverse Operator: Pick two positions i and j at random from the initial solution, and then reverse the values between i and j .

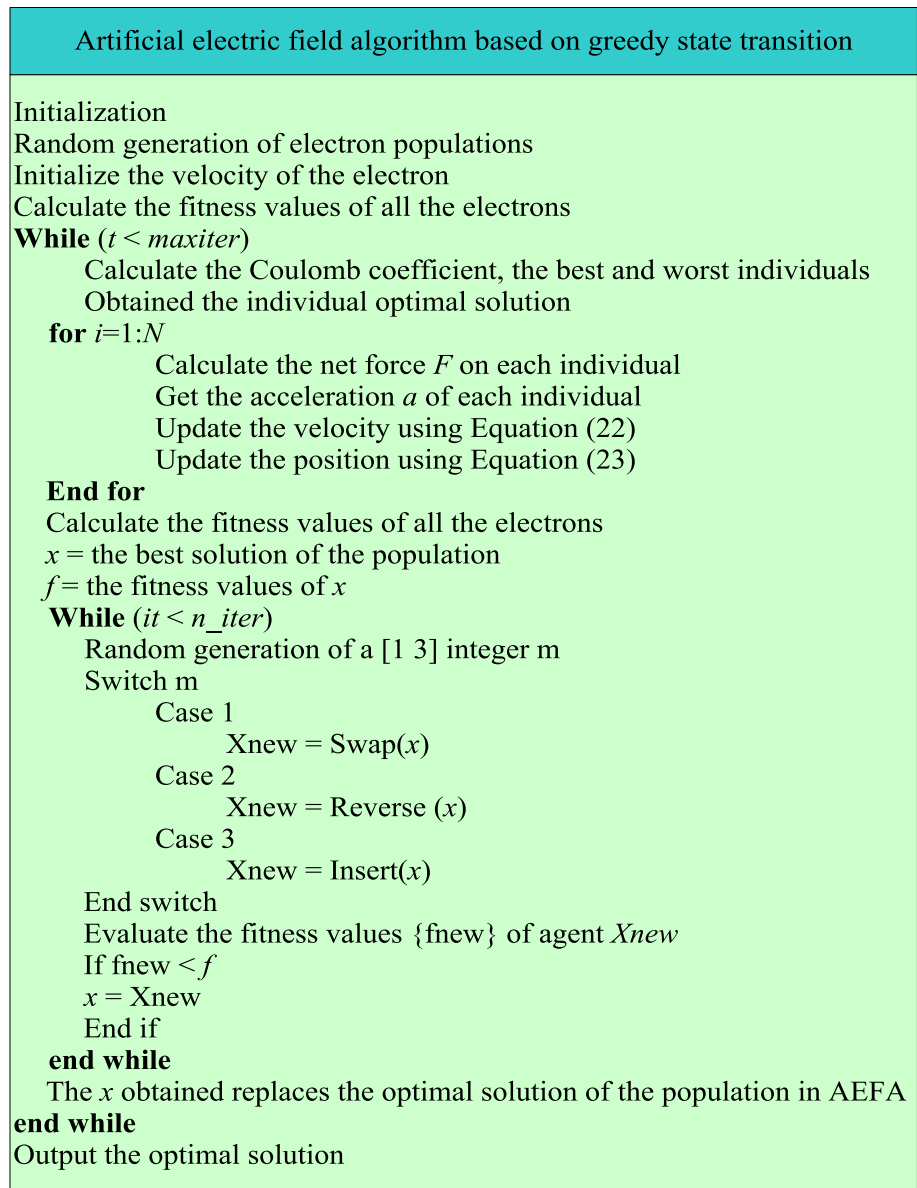
$$x_{new} = Reverse(x_{i,j}), \tag{12}$$

where x_{new} is the immediate solution of x generated, and i and j are two different positional indexes in x , respectively. $Reverse(x_{i,j})$ indicates the reverse order of all values from i to j (Fig. 7).

Insert Operator: Two positions i and j are randomly selected from the initial solution, and the corresponding value for position i is inserted after position j .

$$x_{new} = Insert(x_{i,j}), \tag{13}$$

Fig. 12 GSTAEFA



where x_{new} is the immediate solution of x generated, and i and j are two different positional indexes in x , respectively. Insert ($x_{i,j}$) means that the value corresponding to position i is inserted after position j (Fig. 8).

Figure 9 shows the proposed greedy state transition algorithm.

3.4 The Principle of AEFA

In AEFA systems, the individual is described as a point particle that moves under electrostatic attraction. AEFA is based on Coulomb's law [37]. The initialization of an individual in AEFA is shown in Eq. (14).

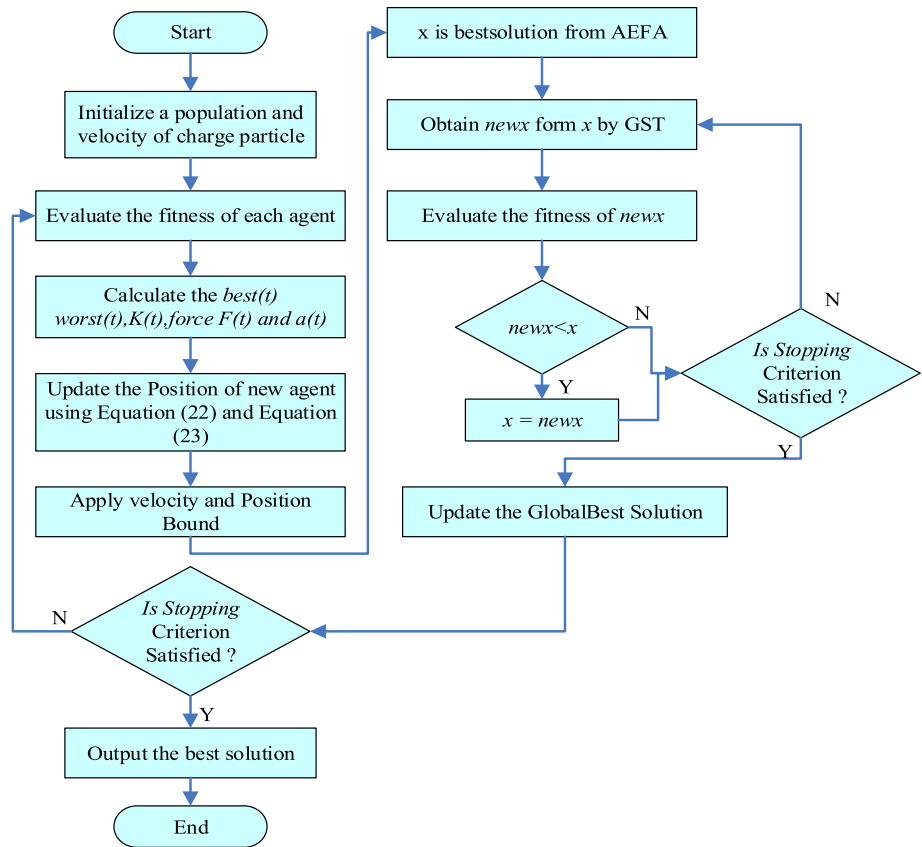
$$X_i(t) = (X_i^1, X_i^2, \dots, X_i^D) \quad i = 1, 2, \dots, N \text{ and } d = 1, 2, \dots, D \quad (14)$$

where N represents the total number of individuals, that is, the total number of charged particles in the system, and D represents the dimension of the problem. In the algorithm, it is necessary to obtain the individual optimal particle in each iteration process as shown in Eq. (15).

$$P_i^d(t+1) = \begin{cases} X_i^d(t+1) & \text{if } f(X_i(t+1)) \leq f(P_i(t)) \\ P_i^d(t) & \text{if } f(X_i(t+1)) > f(P_i(t)) \end{cases} \quad (15)$$

In the process of iteration t , the force of any individual J acting on particle I is expressed by Eq. (16).

Fig. 13 GSTAEFA flowchart



$$F_{ij}^d(t) = K(t) \frac{Q_i(t) \cdot Q_j(t) \cdot (P_j^d(t) - X_i^d(t))}{R_{ij}(t) + \epsilon}, \quad (16)$$

where $Q_i(t)$ and $Q_j(t)$ are the amount of charge in different individuals. K is Coulomb's constant, ϵ is used to prevent the denominator from being 0, and R is the Euclidian distance between two individuals. Formulas 17 and 18 give the calculation method of R .

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2, \quad (17)$$

$$K(t) = K_0 \cdot e^{\left(-\alpha \frac{iter}{\max iter}\right)}, \quad (18)$$

where parameters α and K_0 are two important parameters to balance the exploration and development capabilities of the algorithm.

The resultant force of particle i in the iterative process is shown in Formula (19).

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand \cdot F_{ij}^d(t), \quad (19)$$

where $rand$ is a random number in the range of $[0, 1]$ that conforms to the uniform distribution. Then the resultant force can be calculated through the above equation, and the

electric field strength E of the current particle can be calculated next.

$$E_i^d(t) = \frac{F_i^d(t)}{Q_i(t)}. \quad (20)$$

The current individual acceleration is deduced by calculating the power plant strength and the resultant force.

$$a_i^d(t) = \frac{Q_i(t) \cdot E_i^d(t)}{M_i(t)} \quad (21)$$

Combined with Newton's law of motion, given the current velocity and acceleration of the individual, calculate the velocity and position at the next moment.

$$V_i^d(t + 1) = rand \cdot V_i^d(t) + a_i^d(t), \quad (22)$$

$$X_i^d(t + 1) = X_i^d(t) + V_i^d(t + 1). \quad (23)$$

After mastering the above principles of AEFA, we can describe AEFA in the form of pseudocode, as shown in the figure below (Fig. 10):

Figure 11 is the flowchart of AEFA. The algorithm of AEFA is simple in logic and easy to implement, with less parameter to be optimized.

Table 1 The results of different algorithms

City	Metric	GSTAEFA	AEFA	SA	GA	FA	ABC	SMA	GBO	FPA
25	Best	12.2659	17.8427	13.0189	22.4515	13.3187	16.5753	13.4363	16.7849	18.2515
	Worst	13.3031	27.6676	17.3576	27.1218	21.1041	22.7531	19.0105	23.8987	26.1379
	Mean	12.6859	22.9612	14.5134	25.1259	17.4966	19.8188	16.0500	20.0828	22.6546
	Std	0.2574	2.6717	0.8282	1.0375	2.0343	1.7699	1.3341	1.9401	2.1432
50	Best	18.9704	39.7550	21.3992	56.7231	26.5726	47.4343	33.8753	39.8742	50.1582
	Worst	21.6607	52.7779	27.3104	61.8521	43.5759	58.4299	47.5290	56.4006	59.9728
	Mean	20.1615	46.7592	24.3748	59.3237	35.7598	55.0358	39.6150	47.4063	56.3602
	Std	0.7480	3.1937	1.4055	1.3211	3.9125	2.2966	2.9162	4.5322	2.5860
100	Best	30.6540	92.5786	37.5775	121.9332	62.6927	116.0651	82.5532	97.6291	111.4641
	Worst	37.0926	115.0049	45.7418	132.5262	86.5049	129.7012	113.0407	116.4376	131.7030
	Mean	33.1572	101.2104	40.8490	128.7085	77.2783	125.4620	94.7354	106.1319	125.4348
	Std	1.7512	4.1576	1.8741	2.4613	5.7260	3.0183	6.7852	5.1330	4.8862
150	Best	41.0929	151.8082	54.2255	194.2023	109.2390	188.8349	137.6719	149.8335	188.2387
	Worst	48.5411	173.0375	64.7362	205.2288	136.2112	200.4998	173.5753	190.3252	205.6513
	Mean	45.3411	162.7115	59.2002	200.9191	122.8757	196.9506	154.4054	174.0331	199.1189
	Std	1.6539	5.8761	2.9220	2.6880	6.9193	2.4236	9.9721	8.0075	4.3373
200	Best	52.1173	200.2330	74.0615	266.5113	157.2991	257.3318	205.7324	219.1170	242.0012
	Worst	67.5497	249.2681	85.8465	276.1055	183.5040	272.6618	253.2867	257.3528	276.4259
	Mean	60.2786	220.6967	79.1095	272.1671	170.4854	267.6942	223.0627	239.0869	269.7285
	Std	3.6175	9.3922	2.7561	2.1960	7.2560	3.7414	11.0059	9.7603	6.2704
300	Best	91.4491	336.8330	122.0842	412.3508	266.4446	402.4688	330.6292	363.6838	410.6104
	Worst	113.3120	369.8211	135.6951	426.2965	318.7650	421.0053	391.1878	401.4178	427.5726
	Mean	100.8531	352.4861	129.1399	419.8391	291.0377	416.9570	366.9345	381.1536	418.8782
	Std	5.4146	8.9164	3.8978	3.4250	14.2221	3.8217	13.3297	10.6900	3.7614
400	Best	133.3439	450.6558	173.6742	558.3877	344.6362	552.9211	487.0252	476.5118	543.2884
	Worst	153.3989	509.7010	194.8835	575.4567	453.8138	568.1294	533.4852	552.0739	570.9623
	Mean	143.8503	487.3507	180.5232	567.2465	417.1712	562.0554	507.2280	525.9420	560.7849
	Std	5.4086	12.5566	4.6852	3.8799	25.5232	3.8317	13.4703	16.0116	6.5265
500	Best	178.2932	594.2182	230.5083	708.6881	491.6034	697.5049	630.5138	639.1904	672.3726
	Worst	205.4165	656.6768	251.8035	727.6473	711.6490	721.7767	696.3405	698.2988	720.6142
	Mean	194.5212	625.4403	239.5992	721.3680	568.3318	713.4527	660.6123	671.7454	708.4456
	Std	7.0451	15.1778	5.4465	4.3418	47.8250	6.0320	15.0767	17.0345	10.4925

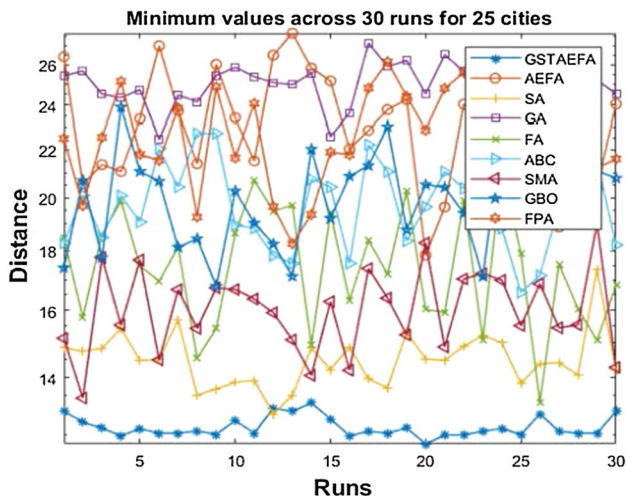


Fig. 14 Results from 30 separate runs

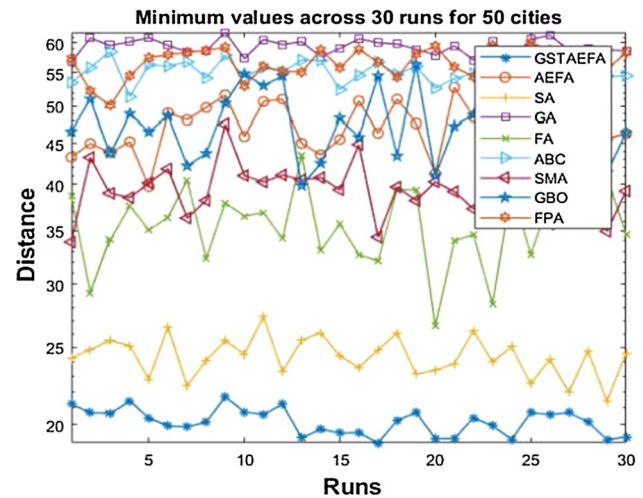


Fig. 15 Results from 30 separate runs



Fig. 16 Results from 30 separate runs

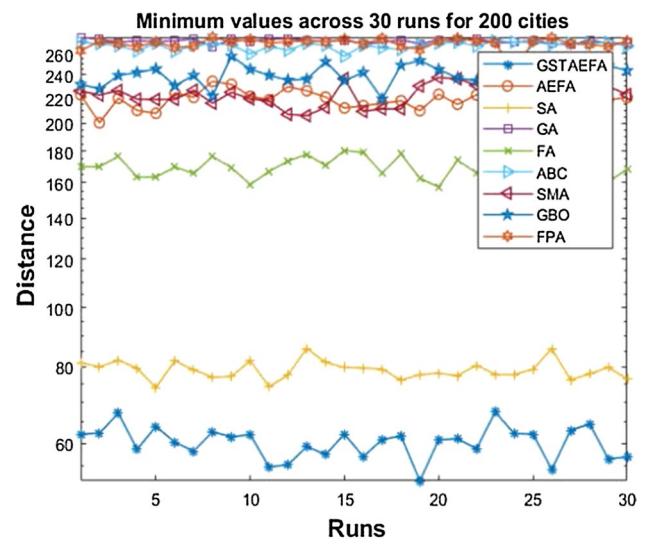


Fig. 18 Results from 30 separate runs



Fig. 17 Results from 30 separate runs



Fig. 19 Results from 30 separate runs

3.5 Artificial Electric Field Algorithm Based on Greedy State Transition

Although AEFA has many advantages, it still needs to be improved. For example, the algorithm can be improved to speed up convergence and reduce the probability of falling into local solutions. Therefore, we add the greedy state transition strategy in this paper to improve its performance after combining with the original algorithm. The local search mechanism improves the ability of AEFA to search deeper in the solution space to obtain better solutions. In the optimization process of AEFA, the optimal individuals in each

generation are selected for local search using GST strategy to obtain better individuals. After each GST operation, a second-best solution is obtained, which is returned to AEFA and the next search begins. Figure 12 is the pseudocode for GSTAEFA, and Fig. 13 is the flowchart of GSTAEFA.

3.6 GSTAEFA Complexity Analysis

The complexity of any optimization algorithm is mainly derived from the algorithm itself and the evaluation of the objective function. In our algorithm, the complexity mainly

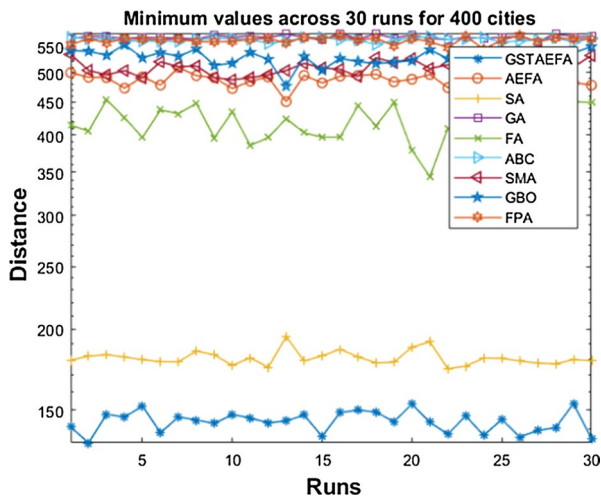


Fig. 20 Results from 30 separate runs

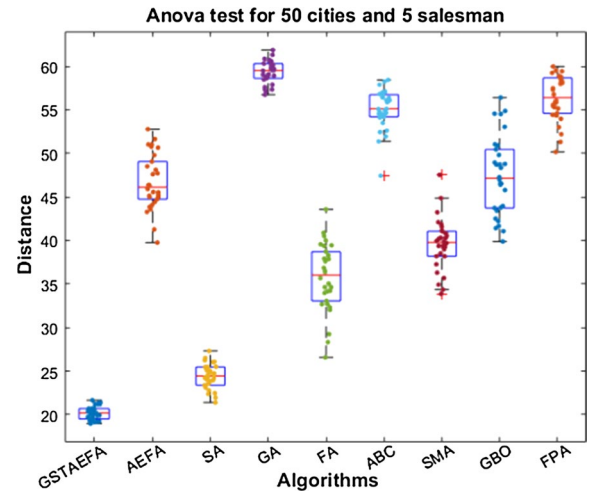


Fig. 23 Variance value and variance divergence



Fig. 21 Results from 30 separate runs

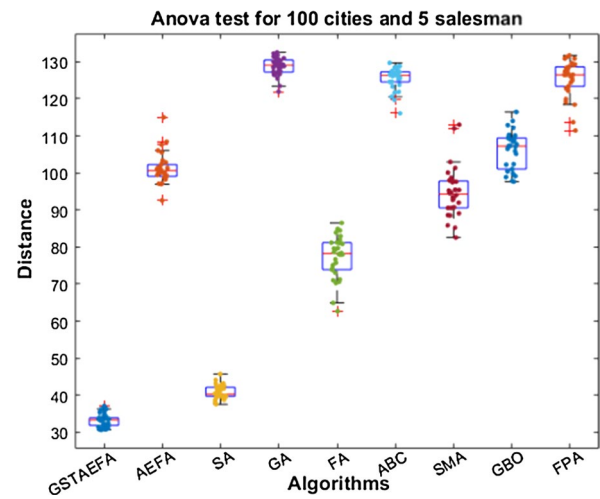


Fig. 24 Variance value and variance divergence

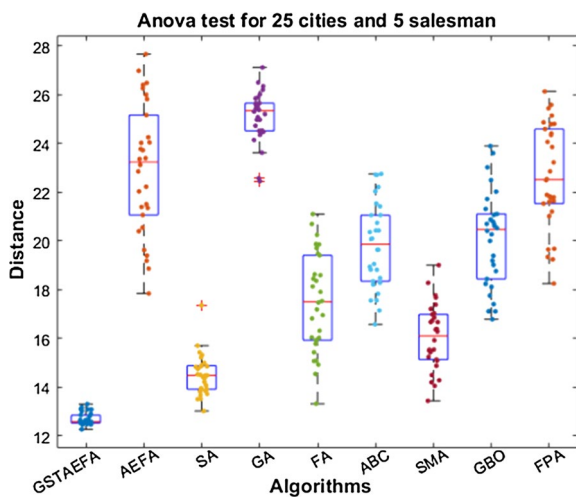


Fig. 22 Variance value and variance divergence

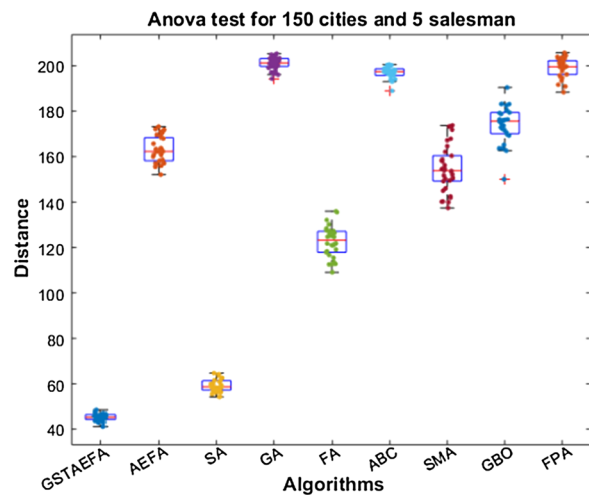


Fig. 25 Variance value and variance divergence

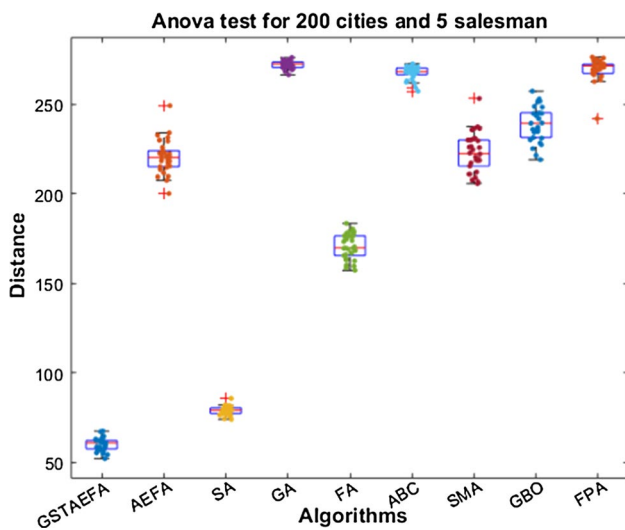


Fig. 26 Variance value and variance divergence

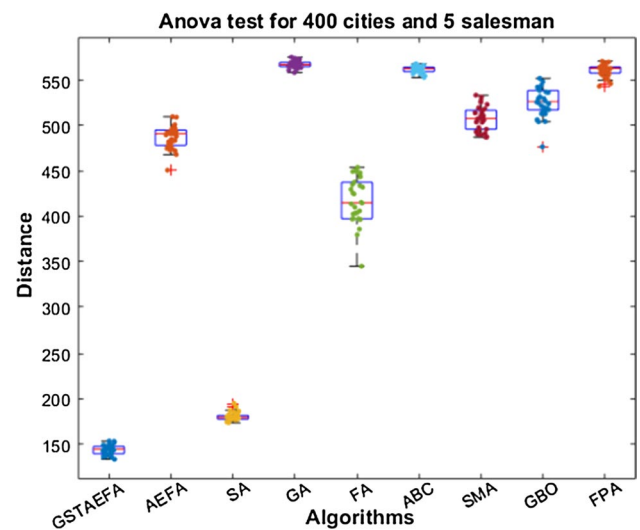


Fig. 28 Variance value and variance divergence

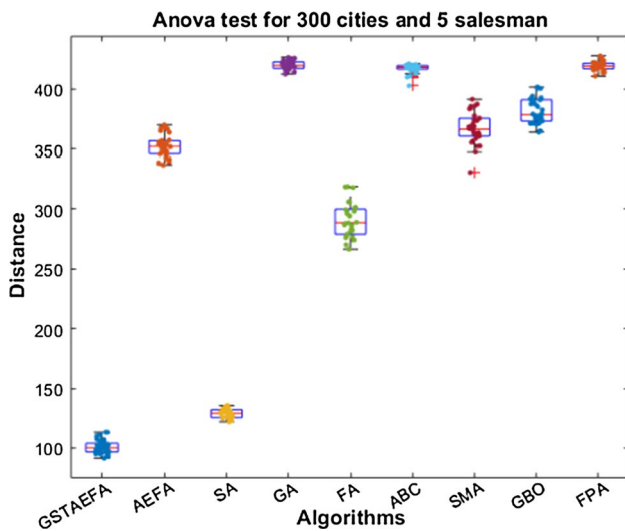


Fig. 27 Variance value and variance divergence

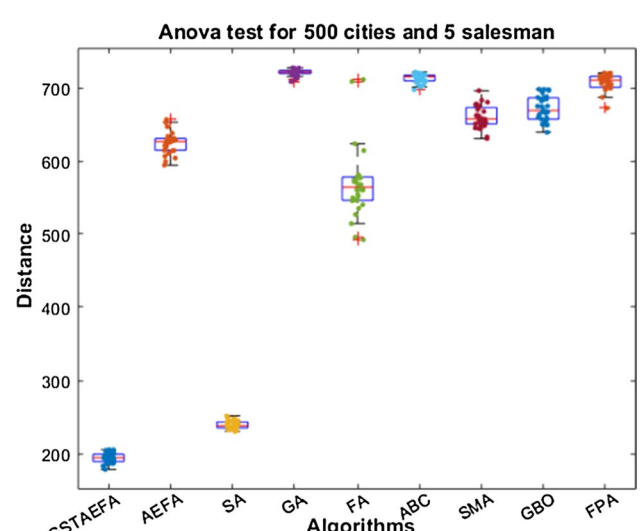


Fig. 29 Variance value and variance divergence

comes from the following parts: the maximum number of iterations T , the number of electrons N , the dimension D of the problem to be solved, and local search strategy GST run times t . Therefore, the time complexity of AEFA, GST, and GSTAEFA can be estimated. AEFA: $O(t \times N \times D)$, GST: $O(t \times it \times D)$, and GSTAEFA: $O(t \times (N \times d + it \times D))$.

4 Experimental Setup and Discussion of Results

We chose a more suitable hardware environment for the simulation of the algorithm, and maintained the principle of fairness in evaluating the algorithm performance. All

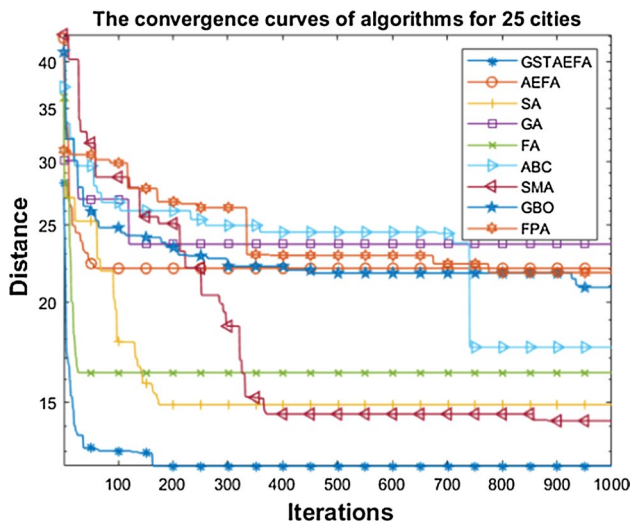


Fig. 30 Convergence diagram of all test algorithms

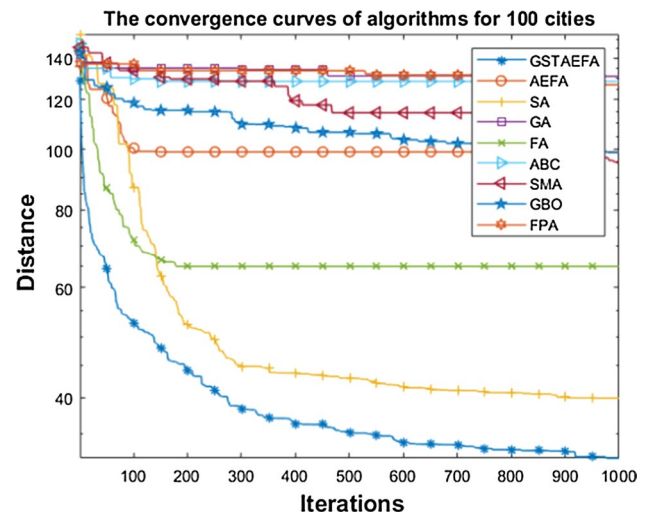


Fig. 32 Convergence diagram of all test algorithms

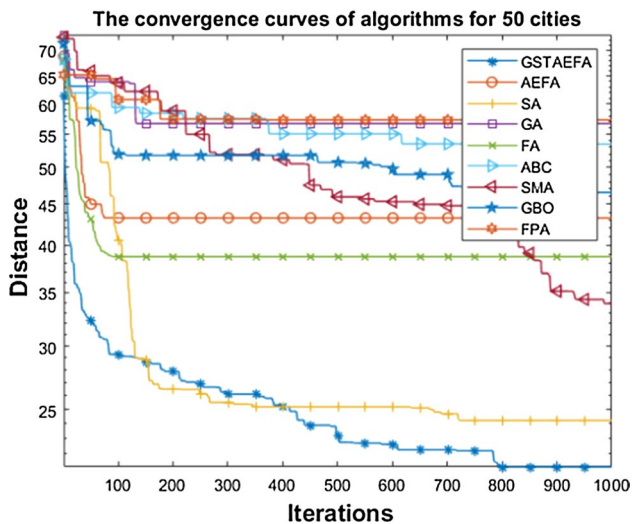


Fig. 31 Convergence diagram of all test algorithms

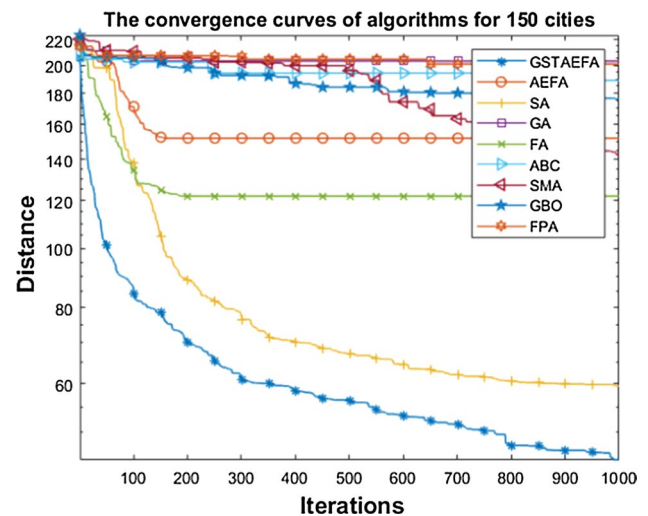


Fig. 33 Convergence diagram of all test algorithms

the comparison algorithms were run on the same machine. The software environment was MATLAB (R 2018a), and the hardware environment was Intel Core i3-6100 processor and 8 GB memory. The experimental results were compared with algorithms that have been applied to solve related problems, including ABC, FPA, GA, FA, SA and other latest metaheuristic algorithms, including AEFA, GBO, and SMA.

4.1 Test Problems

Since the problem in this study is new, some test questions are needed to evaluate the proposed metaheuristic algorithm. For this purpose, seven size test questions were considered from the data collected in the relevant article [34]. The

number of cities in each question is 25, 50, 100, 150, 200, 300, and 400. Finally, in order to improve the persuasiveness of the algorithm, we increased the number to 500 cities to test the performance of the algorithm.

4.2 Experiment Setting

In the experimental process, to ensure fairness, the number of individuals of all the comparison algorithms was set to 30, and the algorithm controls the end of the algorithm according to the value of Iterations, and its value is 1000. To carry out the final experiment on the test problem, some problems are considered to obtain better solutions, and a

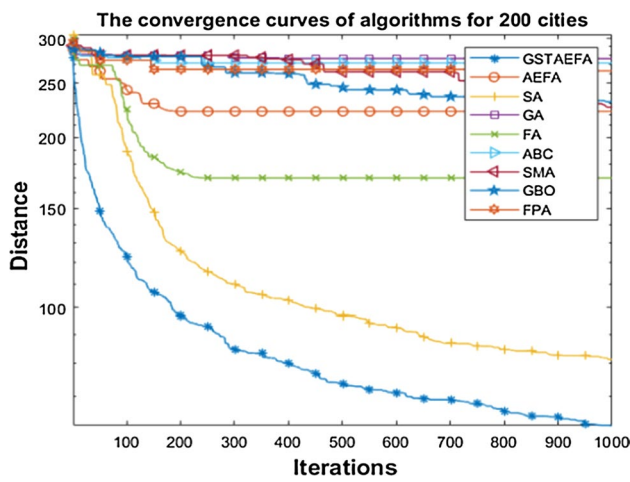


Fig. 34 Convergence diagram of all test algorithms

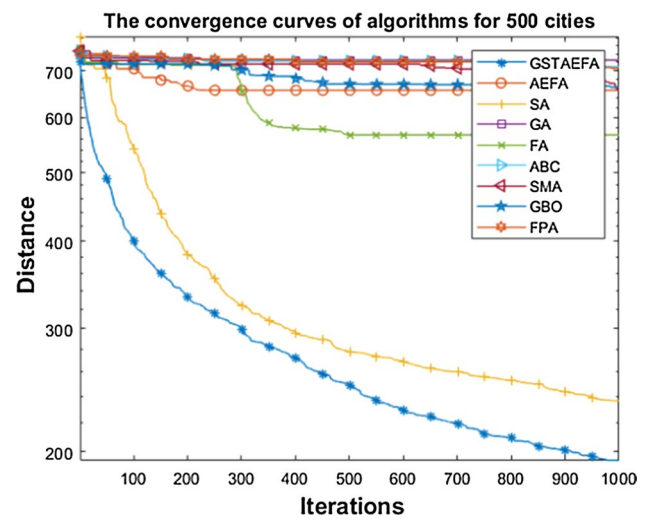


Fig. 37 Convergence diagram of all test algorithms

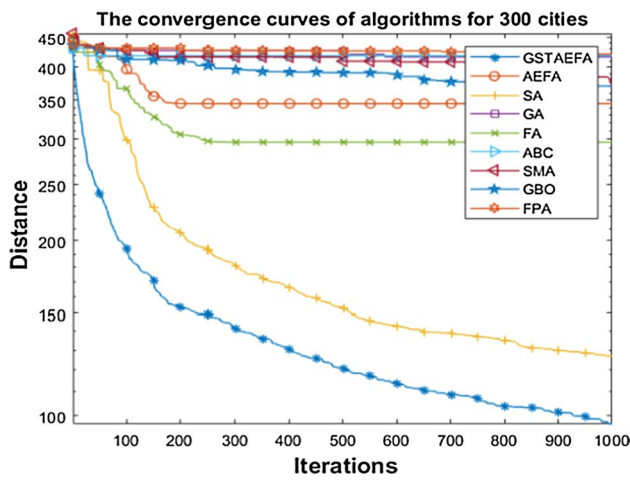


Fig. 35 Convergence diagram of all test algorithms

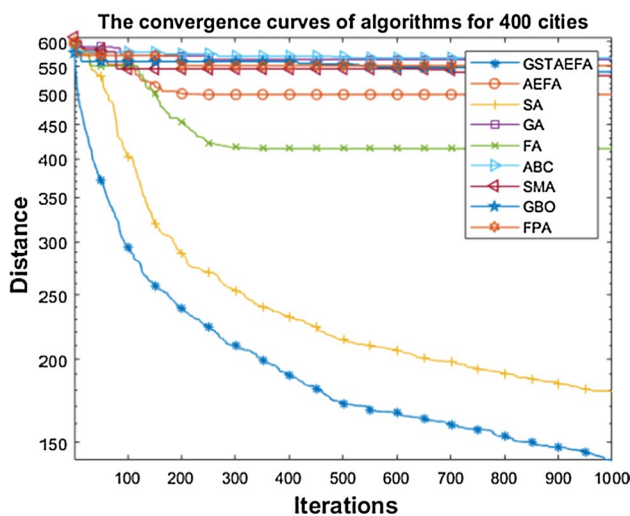


Fig. 36 Convergence diagram of all test algorithms

The best route for 25 cities and 5 salesman

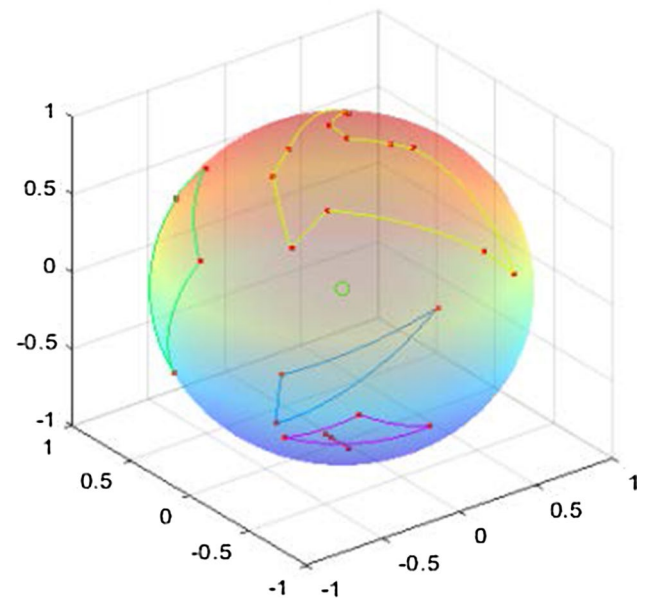


Fig. 38 The optimal planning of 5 salesmen in 25 cities

fair comparative study is also provided for the algorithm. Each algorithm uses standard parameters. To reduce the randomness of the algorithm, each algorithm was run 30 times for each test problem. The minimum value, maximum value, average value, and standard deviation of the obtained objective function were compared, and Friedman rank sum test and *P* value test were performed on the results of all algorithms.

GSTAEFA: NP = 30, Coulomb's constant K_0 is 500, the number of local searches is 100.

AEFA: NP = 30, Coulomb's constant K_0 is 500 [37].

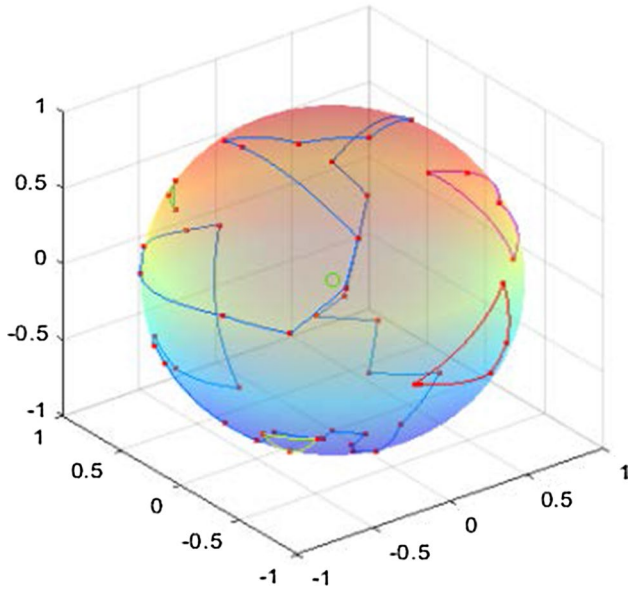
The best route for 50 cities and 5 salesman

Fig. 39 The optimal planning of 5 salesmen in 50 cities

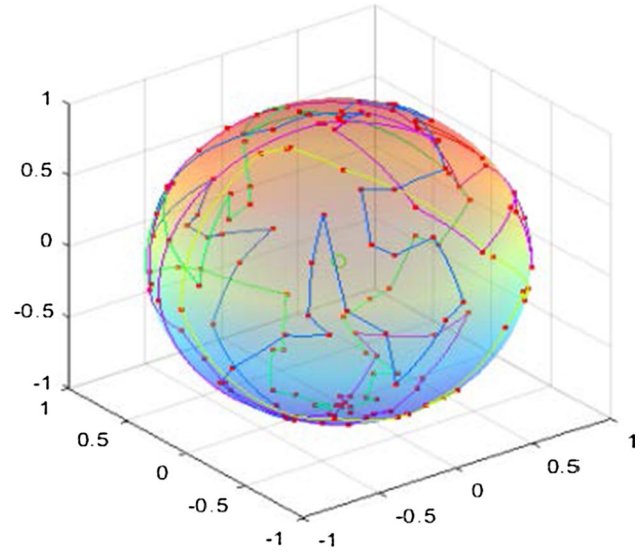
The best route for 150 cities and 5 salesman

Fig. 41 The optimal planning of 5 salesmen in 150 cities

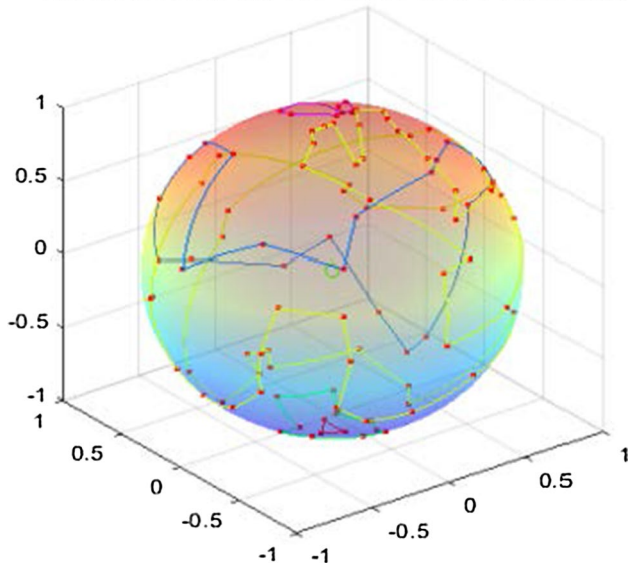
The best route for 100 cities and 5 salesman

Fig. 40 The optimal planning of 5 salesmen in 100 cities

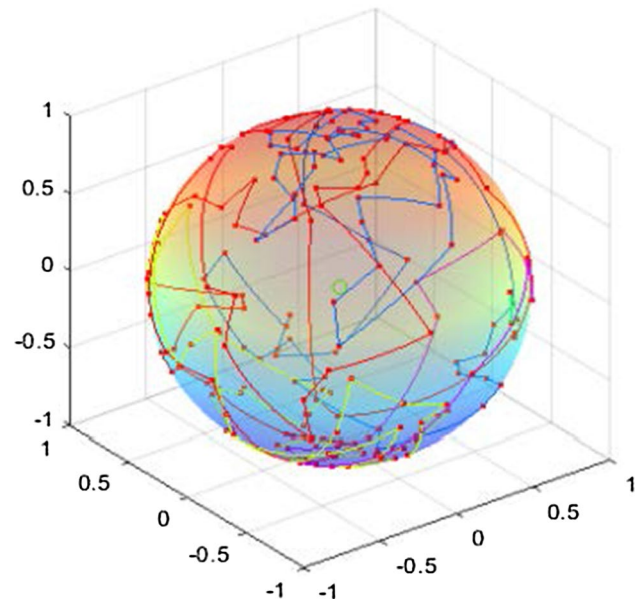
The best route for 200 cities and 5 salesman

Fig. 42 The optimal planning of 5 salesmen in 200 cities

SA: the number of local searches is 100 [18].

GA: NP=30, p_c is 0.8, p_m is 0.8 [31].

FA: NP=30, alpha is 0.8, beta is 0.8, mutation is 0.8, mutation damping ratio is 0.8 [53].

ABC: NP=30, (Employed bees + Onlooker bees), Food number = NP/2, acceleration coefficient reduces exponentially from 2 to 0 [11].

SMA: NP=30, parameter z is 0.003 [55].

GBO: NP=30, β is 0.2 to 1.2, Probability Parameter is 0.5 [56].

FPA: NP=30, the transfer probability of global and local pollination of flowers was 0.8 [54].

The best route for 300 cities and 5 salesman

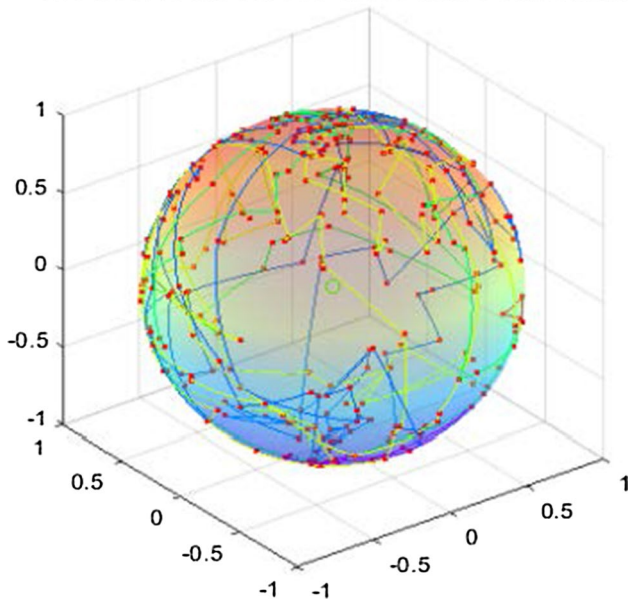


Fig. 43 The optimal planning of 5 salesmen in 300 cities

The best route for 500 cities and 5 salesman

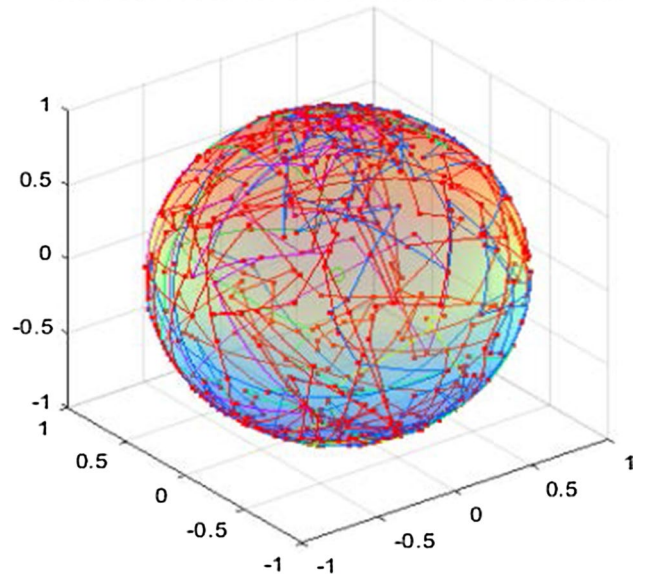


Fig. 45 The optimal planning of 5 salesmen in 500 cities

The best route for 400 cities and 5 salesman

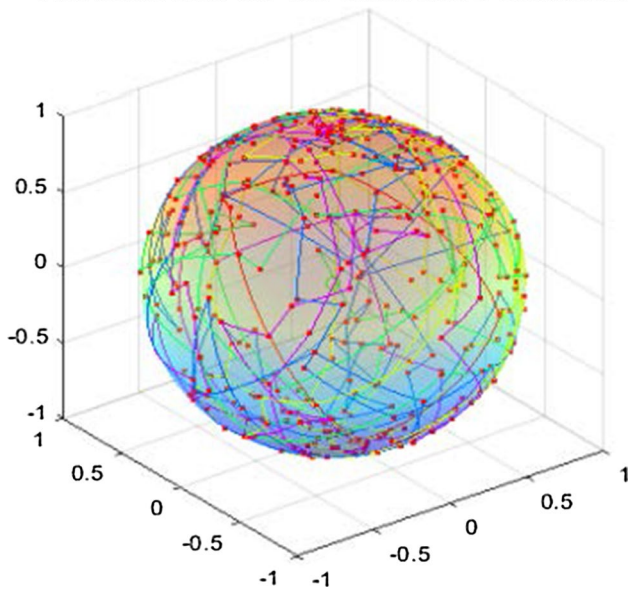


Fig. 44 The optimal planning of 5 salesmen in 400 cities

Different lengths of salesmen in 25 cities

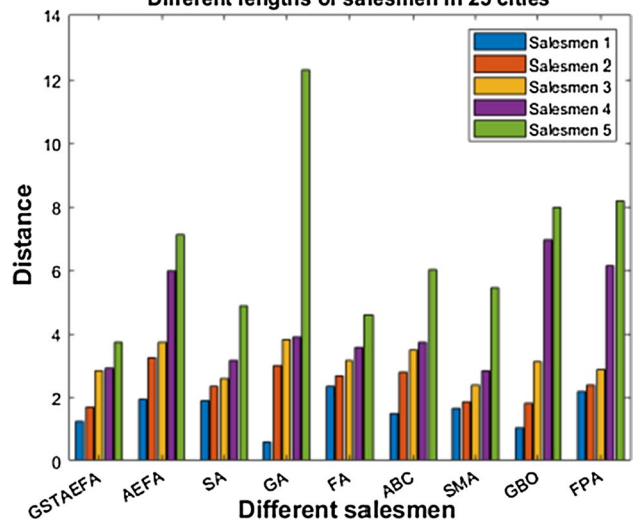


Fig. 46 Path lengths for different salesmen

4.3 Analysis and Comparison of Experimental Results

Next, we explore how the number of cities affects the solution and compare the test results of nine algorithms.

The complexity of the problem is mainly determined by the number of cities, the more cities, the more difficult to solve. Each algorithm runs 30 times independently for different numbers of cities. Table 1 shows the test values of all algorithms at different points on the sphere, where “City” represents the number of cities.

As shown in Table 1, GSTAEFA was significantly superior to other algorithms. In all cases, GSTAEFA obtained the highest accuracy of the results, and at the same time,

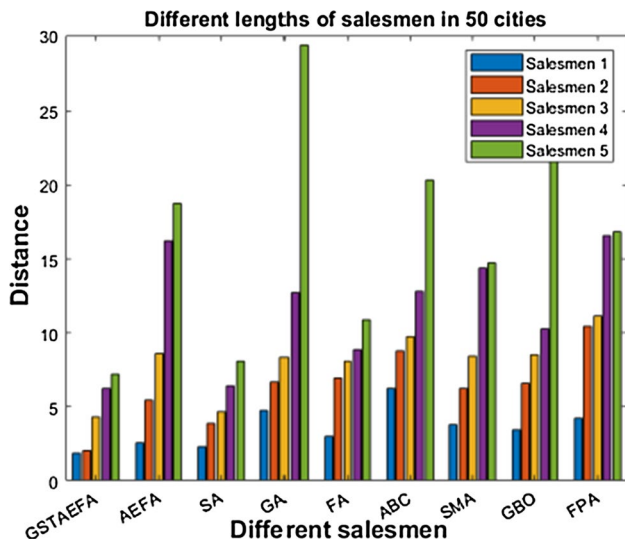


Fig. 47 Path lengths for different salesmen

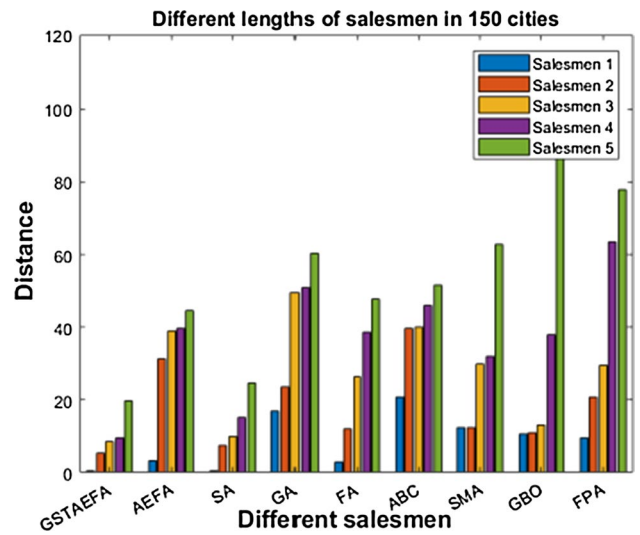


Fig. 49 Path lengths for different salesmen

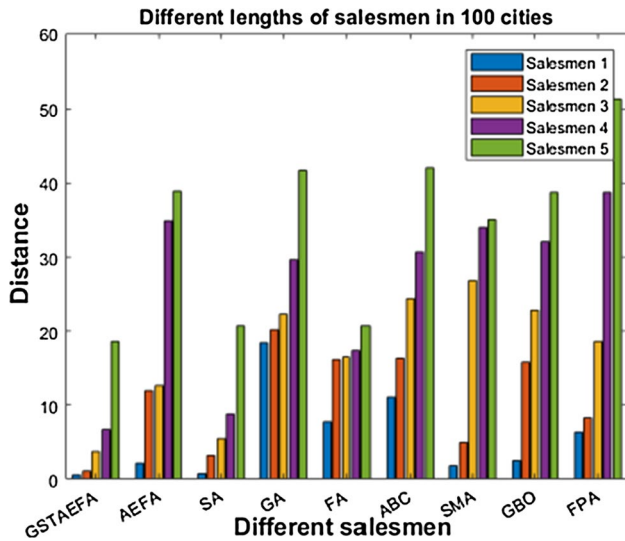


Fig. 48 Path lengths for different salesmen

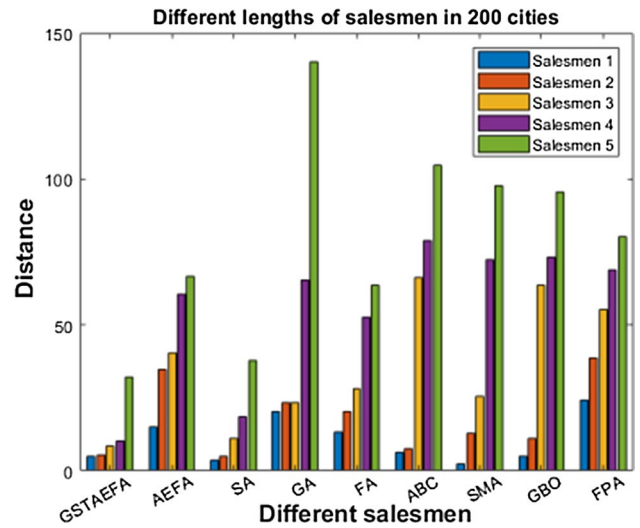


Fig. 50 Path lengths for different salesmen

the average value in all cases was also the best. In the case of fewer cities, the results obtained by different algorithms are very close, but GSTAEFA has the best stability. It can be analyzed from the results that when the number of points on the sphere increases, the searching ability of other algorithms will be greatly reduced, and when the number of cities is 200, 300, 400, and 500, the stability of SA, GA, and ABC is better than that of GSTAEFA, while the results of GSTAEFA algorithm are 52.1173, 91.4491,

133.3439, and 178.2932, respectively. Development performance is optimal.

Figures 14, 15, 16, 17, 18, 19, 20, and 21 show a line plot of the 30 times result optimal values. When the dimension of the problem is low, the results of the algorithm proposed by us are not different from other algorithms, but the accuracy of the results is always higher than other algorithms. With the continuous increase of the number of cities, the accuracy of GSTAEFA algorithm gradually opened a gap with other

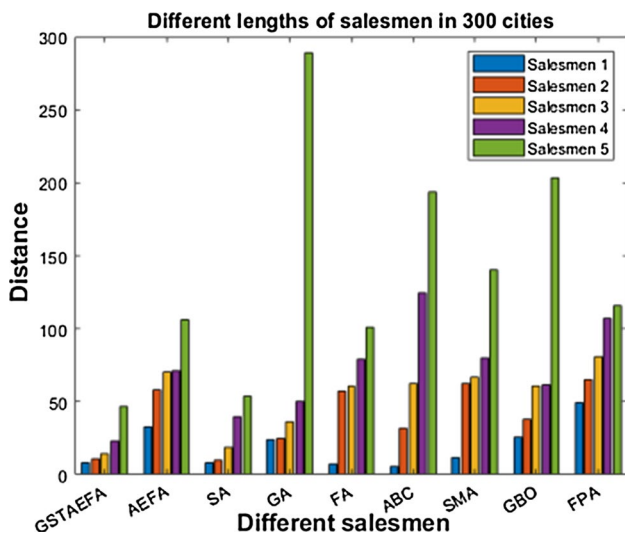


Fig. 51 Path lengths for different salesmen

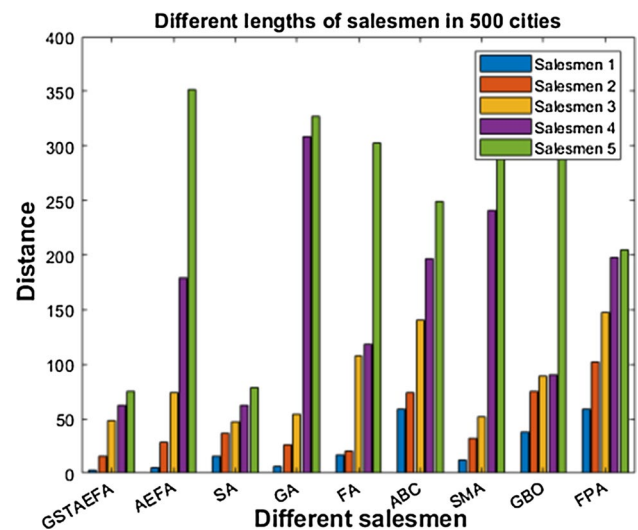


Fig. 53 Path lengths for different salesmen

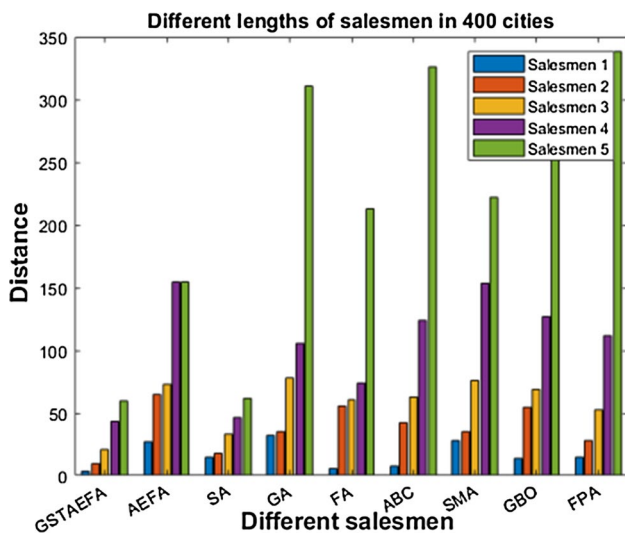


Fig. 52 Path lengths for different salesmen

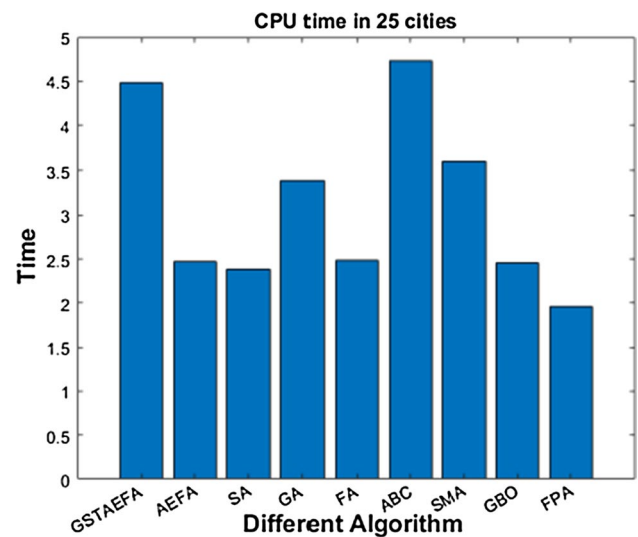


Fig. 54 CPU time of different algorithms

algorithms. In the variance diagram and scatter diagram shown in Figs. 22, 23, 24, 25, 26, 27, 28, and 29, it can be seen that GSTAEFA has good stability, and the results of each time are around the range of the variance diagram. It can be seen from the scatter diagram that the points of AEFA and FA are relatively discrete and have poor stability.

To visually demonstrate the power of all algorithms when solving SMTSP, we use the convergence curve shown in Figs. 30, 31, 32, 33, 34, 35, 36, and 37. Before the execution of the algorithm is finished, the results of other algorithms have been stabilized in a local solution range. The convergence effect of SA was similar to that of GSTAEFA, but the convergence speed and accuracy of SA were lower than

that of GSTAEFA. Although the optimal solution procedure may not be given, we can conclude that on most data sets, GSTAEFA can obtain the best results and converge faster than the compared algorithm.

Figures 38, 39, 40, 41, 42, 43, 44, and 45 intuitively show the optimal path searched by GSTAEFA. All cities and routes can be seen simultaneously in transparent mode. It should be noted that the routes shown in the figure are all approximately globally optimal, especially in large-scale problems.

We present a bar chart of five path lengths with different number of cities, as shown in Figs. 46, 47, 48, 49, 50, 51, 52, and 53. It can be seen that the path length of each

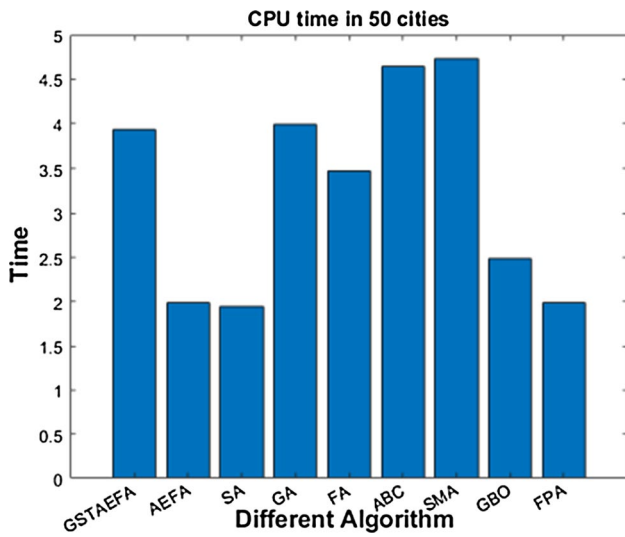


Fig. 55 CPU time of different algorithms

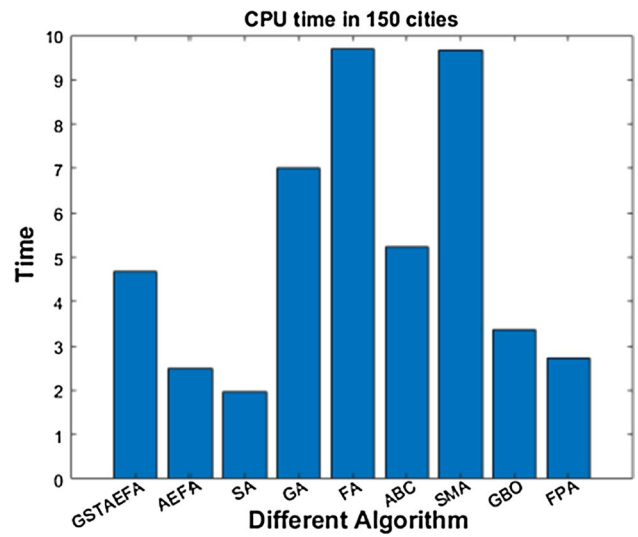


Fig. 57 CPU time of different algorithms

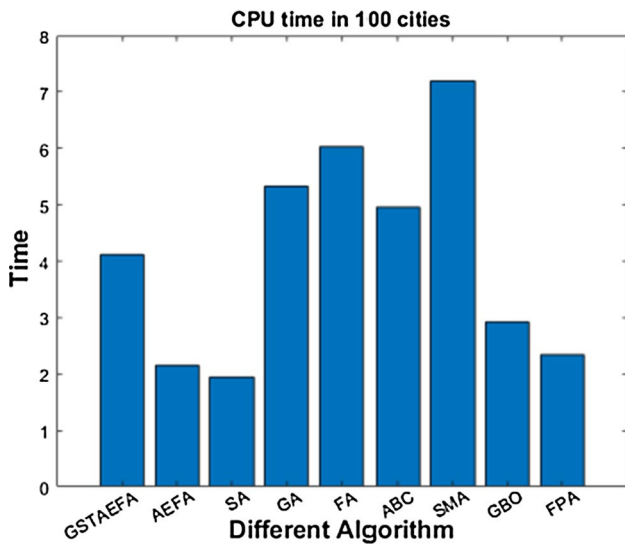


Fig. 56 CPU time of different algorithms

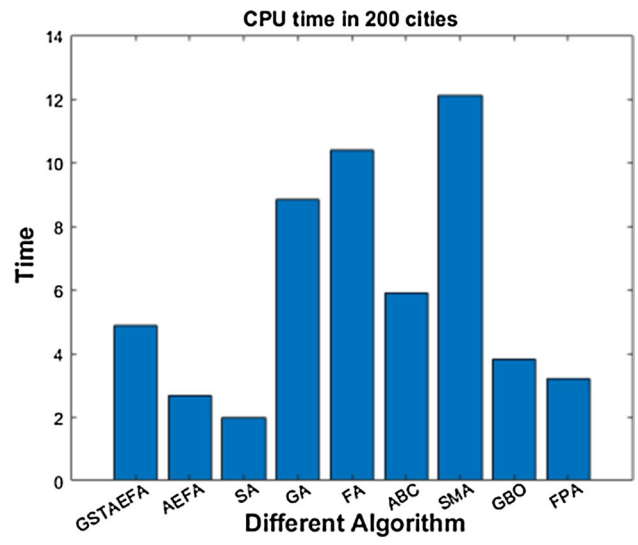


Fig. 58 CPU time of different algorithms

salesperson is smaller than other algorithms, so the sum of all paths must be better than other algorithms. So far, our proposed algorithm has a great advantage in finding the optimal solution, regardless of the size of the city.

As shown in Figs. 54, 55, 56, 57, 58, 59, 60, and 61, we show a bar chart of CPU elapsed time for different algorithms. In the case of lower dimension, the running time of GSTAEFA is relatively large, but the convergence accuracy is the highest. And as the problem dimension increases, GSTAEFA's time performance becomes more and more outstanding.

4.4 Statistical Analysis

In order to conduct statistical analysis on the experimental results, two nonparametric tests, Friedman test [57] and Wilcoxon signed-rank test [58], were used to evaluate the performance of the proposed algorithm. Friedman's test is a statistical test for the consistency of multiple correlated samples. The test calculates the ranking of the shortest distance achieved by each algorithm on each instance, with a best value of 1 and a worst value of 9. Table 2 shows the average ranking obtained by the nine algorithms using the

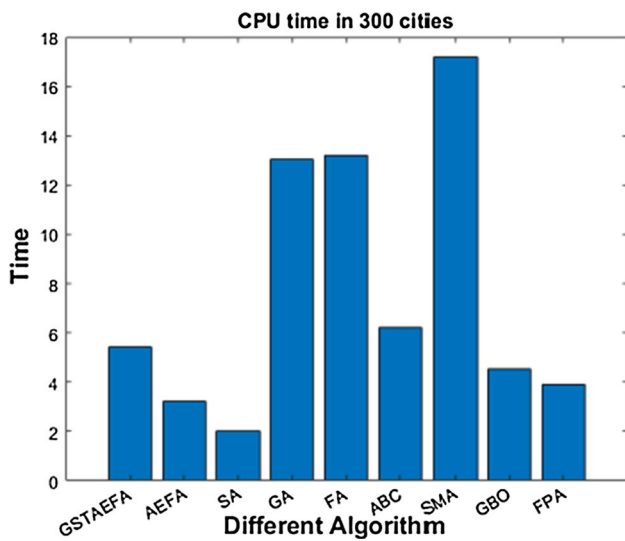


Fig. 59 CPU time of different algorithms

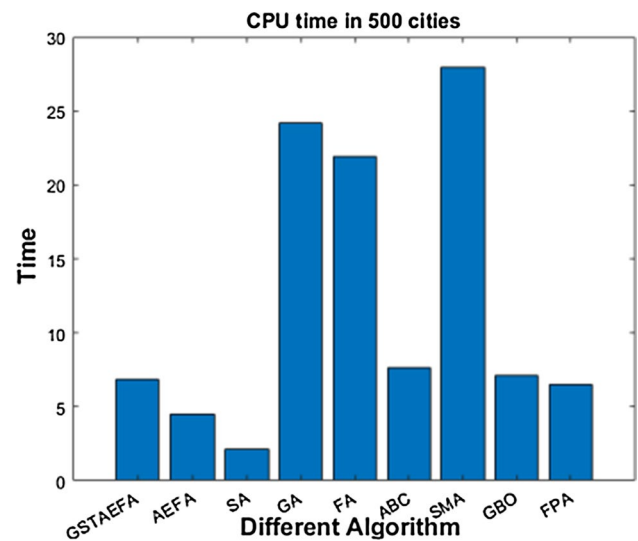


Fig. 61 CPU time of different algorithms

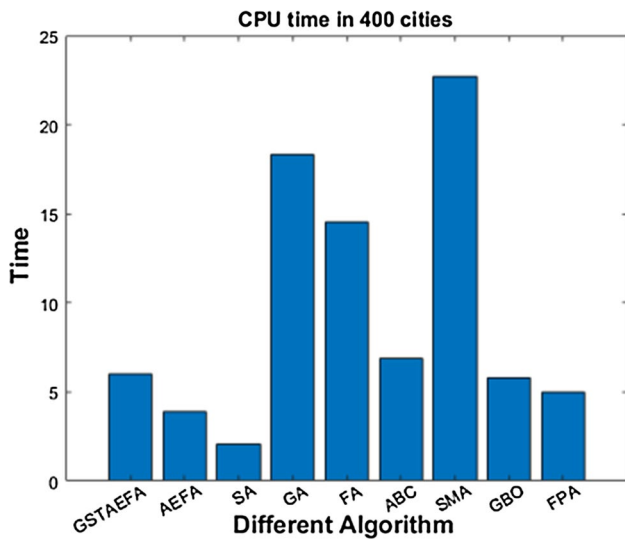


Fig. 60 CPU time of different algorithms

Friedman test at the 95% confidence level. As you can see, our proposed algorithm ranks first. Experimental results show that the proposed GSTAEFA search strategy is effective in solving the SMTSP problem, and its capabilities are the best among the compared algorithms.

Wilcoxon signed-rank test is a nonparametric test. Statistical test of rank is based on sample observation. The test results were used to compare the differences between the algorithms. Less than 5% indicates that the proposed algorithm is good enough. Table 3 shows the detection results. *P* values less than 0.05 indicate that GSTAEFA has a good performance in statistics. In the table, the value is greater than or equal to 0.05. All detection results are less

Table 2 Friedman test results for different algorithms

Algorithm	Friedman rank	Rank
GSTAEFA	1.00	1
AEFA	4.92	5
SA	2.03	2
GA	8.61	9
FA	3.23	3
ABC	7.28	7
SMA	4.46	4
GBO	5.69	6
FPA	7.76	8

than 0.05, indicating that the performance of GSTAEFA is irreplaceable.

From the above two non-parametric test, we can clearly see that GSTAEFA is superior to other advanced methods in solving SMTSP, and the algorithm is obviously different from other algorithms. It shows that GSTAEFA is irreplaceable when solving SMTSP.

5 Conclusions and Further Works

In this paper, a hybrid algorithm GSTAEFA based on traditional AEFA hybrid greedy state transition strategy is proposed. AEFA is an effective naturally inspired algorithm which relies on Coulomb's electrostatic law and Newton's laws of motion. It has been successfully applied to various types of optimization problems and obtained efficient solutions. Although AEFA has a satisfactory exploration experience, its development capability is its

Table 3 Wilcoxon p values are compared with other algorithms

City	AEFA	SA	GA	FA	ABC	SMA	GBO	FPA
25	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
50	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
100	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
150	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
200	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
300	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
400	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
500	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05

main disadvantage. In order to overcome these problems of AEFA and improve its development ability, greedy state transfer strategy is considered in the proposed algorithm to effectively improve its development ability, which will make it more efficient, more effective, and easier to solve various combinatorial optimization problems. We apply GSTAEFA to solve a new spherical multi-traveler problem, which is considered as the shortest path problem for multiple travel agencies around the world, and test the performance of the algorithm with different dimensional examples. The results show that the artificial electric field algorithm based on greedy state transition strategy is superior to the classical algorithm.

In addition, GSTAEFA is weak in time complexity, so it still has room for further optimization. In the future, we will consider reducing the time cost of the algorithm while ensuring the solution accuracy. We can try to reduce the local search times according to probability to reduce the time complexity. Finally, we focus on single-target SMTSP, so it is interesting that our algorithm can be used to solve multi-target projects. As a future research, the formulas proposed in this study can be used to obtain Pareto optimal or nondominated solutions through multi-objective precise and multi-objective metaheuristic solutions.

Acknowledgements The authors express great thanks to the financial support from the National Natural Science Foundation of China.

Author Contributions JB carried out the GSTAEFA algorithm studies, participated in the drafted the manuscript. GZ carried out the original draft and review & editing; YZ carried out the review & editing; QL carried out the design algorithm model; WD carried out the review & editing. All authors read and approved the final manuscript.

Funding This research is funded by the National Natural Science Foundation of China, Grant number 62066005; and Supported by the Project of Guangxi Natural Science Foundation, Grant number 2018GXNSFAA138146.

Availability of Data and Material Not applicable.

Declarations

Conflict of Interest The authors declare no conflicts of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for Publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jünger, M., Reinelt, G., Rinaldi, G.: The traveling salesman problem. *Handbooks Oper. Res. Manag. Sci.* **7**, 225–330 (1995)
- Zhao, Y., Zheng, Z., Liu, Y.: Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **158**, 54–64 (2018)
- Joo, C.M., Kim, B.S.: Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability. *Comput. Ind. Eng.* **85**, 102–109 (2015)
- Xi, F., Wang, W., He, L., Huang, Z., Yang, L., Zhang, L.: Research on improved NSGA-II Algorithm and its application in emergency management. *Math. Prob. Eng.* **2018**, 1–13 (2018)
- Gouveia, L., Leitner, M., Ruthmair, M.: Extended Formulations and branch-and-cut algorithms for the black-and-white traveling salesman problem. *Eur. J. Oper. Res.* **262**(3), 908–928 (2017)
- Kinable, J., Smeulders, B., Delcour, E., Spieksma, F.C.R.: Exact algorithms for the Equitable Traveling Salesman Problem. *Eur. J. Oper. Res.* **261**(2), 475–485 (2017)
- Asadpour, A., Goemans, M.X., Mdry, A., Gharan, S.O., Saberi, A.: An $O(\log n / \log \log n)$ -Approximation algorithm for the asymmetric traveling salesman problem. *Oper. Res.* **65**(4), 1043–1061 (2017)

8. Nagata, Y., Soler, D.: A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Syst. Appl.* **39**(10), 8947–8953 (2012)
9. Buriol, L., Franca, P.M., Moscato, P.: A new memetic algorithm for the asymmetric traveling salesman problem. *J Heuristics* **10**(5), 483–506 (2004)
10. Psychas, I.D., Delimpasi, E., Marinakis, Y.: Hybrid evolutionary algorithms for the multiobjective traveling salesman problem. *Expert Syst. Appl.* **42**(22), 8956–8970 (2015)
11. Choong, S.S., Wong, L.P., Lim, C.P.: An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. *Swarm Evol. Comput.* **44**, 622–635 (2019)
12. Escario, J.B., Jimenez, J.F., Giron-Sierra, J.M.: Ant colony extended: experiments on the travelling salesman problem. *Expert Syst. Appl.* **42**(1), 390–410 (2015)
13. Osaba, E., Del Ser, J., Sadollah, A., Bilbao, M.N., Camacho, D.: A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. *Appl. Soft Comput.* **71**, 277–290 (2018)
14. Osaba, E., Yang, X.S., Diaz, F., Lopez-Garcia, P., Carballedo, R.: An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Eng. Appl. Artif. Intell.* **48**, 59–71 (2016)
15. Ouaraab, A., Ahiod, B., Yang, X.S.: Random-key cuckoo search for the travelling salesman problem. *Soft. Comput.* **19**(4), 1099–1106 (2015)
16. Hatamlou, A.: Solving travelling salesman problem using black hole algorithm. *Soft. Comput.* **22**(24), 8167–8175 (2018)
17. Zhong, Y., Lin, J., Wang, L., Zhang, H.: Discrete comprehensive learning particle swarm optimization algorithm with metropolis acceptance criterion for traveling salesman problem. *Swarm Evol. Comput.* **42**, 77–88 (2018)
18. Ezugwu, A.E.-S., Adewumi, A.O., Frîncu, M.E.: Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. *Expert Syst. Appl.* **77**, 189–210 (2017)
19. Jiang, Z., Yang, Q.: A discrete fruit fly optimization algorithm for the traveling salesman problem. *PLoS ONE* **11**(11), e0165804 (2016)
20. Wang, Z., Fang, X., Li, H., Jin, H.: An improved Partheno-Genetic algorithm with reproduction mechanism for the multiple traveling salesperson problem. *IEEE Access.* **99**, 1–1 (2020)
21. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* **34**(3), 209–219 (2006)
22. Ghafurian, S., Javadian, N.: An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. *Appl. Soft Comput.* **11**(1), 1256–1262 (2011)
23. Yousefikhoshbakht, M., Didehvar, F., Rahmati, F.: Modification of the ant colony optimization for solving the multiple traveling salesman problem. *Romanian J. Info. Sci. Technol.* **16**(1), 65–80 (2013)
24. Changdar, C., Pal, R.K., Mahapatra, G.S.: A genetic ant colony optimization based algorithm for solid multiple travelling salesman problem in fuzzy rough environment. *Soft Comput.* **21**(16), 4661–4675 (2016)
25. Malmberg, C.J.: A genetic algorithm for service level based vehicle scheduling. *Eur. J. Oper. Res.* **93**(1), 121–134 (1996)
26. Carter, A.E., Ragsdale, C.T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *Ship Electron. Eng.* **175**(1), 246–257 (2006)
27. Venkatesh, P., Singh, A.: Two metaheuristic approaches for the multiple traveling salesperson problem. *Appl. Soft Comput.* **26**, 74–89 (2015)
28. Chen, Y., Jia, Z., Ai, X., Yang, D., Yu, J.: A modified two-part wolf pack search algorithm for the multiple traveling salesmen problem. *Appl. Soft Comput.* **61**, 714–725 (2017)
29. Zhou, H., Song, M., Pedrycz, W.: A comparative study of improved GA and PSO in solving multiple traveling salesmen problem. *Appl. Soft Comput.* **64**, 564–580 (2018)
30. Takahashi, S., K. Fujimura, H. Tokutaka.: The SOM-TSP method for the three-dimension city location problem. *International Conference on Neural Information Processing IEEE* (2002)
31. Ugur, A., Korukoglu, S., Caliskan, A., et al.: Genetic algorithm based solution for TSP on a sphere. *Math Comput Appl* **14**(3), 219–228 (2009)
32. Lomnitz, C.: On the distribution of distances between random points on a sphere. *Bull Seismol Soc Am* **85**, 951–953 (1995)
33. Eldem, H., Ülker, E.: The application of ant colony optimization in the solution of 3D traveling salesman problem on a sphere. *Eng. Sci. Technol. Int. J.* **20**(4), 1242–1248 (2017)
34. Chen, X., Zhou, Y., et al.: A hybrid algorithm combining glow-worm swarm optimization and complete 2-opt algorithm for spherical travelling salesman problems. *Appl. Soft Comput.* (2017). <https://doi.org/10.1016/j.asoc.2017.04.057>
35. Zhou, Y., Wang, R., Zhao, C., et al.: Discrete greedy flower pollination algorithm for spherical traveling salesman problem. *Neural Comput & Applic* **31**, 2155–2170 (2019)
36. Ouyang, X., Zhou, Y., Luo, Q., Chen, H.: A novel discrete cuckoo search algorithm for spherical traveling salesman problem. *Appl. Math. Info. Sci.* **7**(2), 777–784 (2013)
37. Anita, S., Yadav, A.: AEFA: artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **48**, 93–108 (2019)
38. Anita, Y.A., Kumar, N.: Artificial electric field algorithm for solving real parameter CEC 2017 benchmark problems. In: Nagar, A., Deep, K., Bansal, J., Das, K. (eds.) *Soft Computing for Problem Solving 2019. Advances in Intelligent Systems and Computing*, vol. 1138. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3290-0_13
39. Anita, S., Yadav, A., Kumar, N.: Artificial electric field algorithm for engineering optimization problems. *Expert Syst. Appl.* **149**, 113308 (2020)
40. Anita, S., Yadav, A.: Discrete artificial electric field algorithm for high-order graph matching. *Appl. Soft Comput.* **92**, 106260 (2020)
41. Petwal, H., Rani, R.: An improved artificial electric field algorithm for multi-objective optimization. *Processes* **8**(5), 584 (2020)
42. Selem, S.I., El-Fergany, A.A., Hasanien, H.M.: Artificial electric field algorithm to extract nine parameters of triple-diode photovoltaic model. *Int. J. Energy Res.* **45**(1), 590–604 (2021)
43. Sinthia, P., Malathi, M.: Cancer detection using convolutional neural network optimized by multistrategy artificial electric field algorithm. *Int. J. Imaging Syst. Technol.* **31**(3), 1386–1403 (2020)
44. Sajwan, A., Yadav, A.: An intelligent model for the detection of white blood cells using artificial intelligence. *Comput. Methods Programs Biomed.* **199**(10), 105893 (2021)
45. Sharma, A., Jain, S.K.: Day-ahead optimal reactive power ancillary service procurement under dynamic multi-objective framework in wind integrated deregulated power system. *Energy* **223**(2), 120028 (2021)
46. Das, H., Naik, B., Behera, H.S.: Optimal selection of features using artificial electric field algorithm for classification. *Arab J Sci Eng* **46**(9), 8355–8369 (2021)
47. Niroomand, S.: Hybrid artificial electric field algorithm for assembly line balancing problem with equipment model selection possibility. *Knowledge-Based Syst.* **8**, 106905 (2021)
48. Naderipour, A., Abdul-Malek, Z., Mustafa, M., Guerrero, J.M.: A multi-objective artificial electric field optimization algorithm

- for allocation of wind turbines in distribution systems. *Appl Soft Comput.* **105**(1), 107278 (2021)
49. Jiang, C., Wan, Z., Peng, Z.: A new efficient hybrid algorithm for large scale multiple traveling salesman problems. *Expert Syst. Appl.* **139**, 112867 (2019)
50. Zhou, X., Yang, C., Gui, W.: Initial version of state transition algorithm. In: 2011 Second International Conference on Digital Manufacturing & Automation (IEEE), pp. 644–647 (2011)
51. Zhou, X., Yang, C., Gui, W.: A new transformation into state transition algorithm for finding the global minimum. In: 2011 2nd International Conference on Intelligent Control and Information Processing (IEEE), vol 2, pp. 674–67 (2011)
52. Zhou, X., Yang, C., Gui, W.: State transition algorithm. *J. Ind. Manag. Optim.* **8**(4), 1039–1056 (2012)
53. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. *Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, June 12–15, 2010, Proceedings, Part I* (2010)
54. Yang, X.S.: Flower pollination algorithm for global optimization. In: *Unconventional computation and natural computation, Lecture Notes in Computer Science*, **445**, 240–249 (2012)
55. Li, S., Chen, H., Wang, M., Heidari, A.A., Mirjalili, S.: Slime mould algorithm: a new method for stochastic optimization. *Futur. Gener. Comput. Syst.* **111**, 300–323 (2020)
56. Ahmadianfar, I., Bozorg-Haddad, O., Chu, X.: Gradient-based optimizer: a new metaheuristic optimization algorithm. *Info. Sci.* **540**, 131–159 (2020)
57. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *J. Heuristics* **15**(6), 617–644 (2009)
58. Han, X., Xu, Q., Yue, L., Dong, Y., Xu, X.: an improved crow search algorithm based on spiral search mechanism for solving numerical and engineering optimization problems. *IEEE Access* **99**, 1–1 (2020)
59. Garg, H.: A hybrid GSA-GA algorithm for constrained optimization problems. *Info. Sci.* **478**, 499–523 (2019)
60. Harish, G.: A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data (2015)
61. Garg, H.: A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **274**, 292–305 (2016). <https://doi.org/10.1016/j.amc.2015.11.001>
62. Kundu, T., Garg, H.: A hybrid ITLHHO algorithm for numerical and engineering optimization problems. *Int. J. Intell. Syst.* (2021). <https://doi.org/10.1002/int.22707>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.