Arithmetic optimization algorithm with mathematical operator for spherical minimum spanning tree

Xiaodong Mi
Guangxi University for Nationalities

Qifang Luo (✉ l.qf@163.com)
Guangxi University for Nationalities

Yuanfei Wei
Universiti Kebangsaan Malaysia

Yongquan Zhou
Guangxi University for Nationalities

Huajuan Huang
Guangxi University for Nationalities

Research Article

Keywords: mathematical operator, arithmetic optimization algorithm, mining spanning tree, spherical MST, communication network problem, metaheuristic

Posted Date: May 4th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2876027/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.
Arithmetic optimization algorithm with mathematical operator for spherical minimum spanning tree

Xiaodong Mi, Qifang Luo, Yuanfei Wei, Yongquan Zhou, Huajuan Huang

1College of Artificial Intelligence, Guangxi University for Nationalities, Nanning, 530006 China
2Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia
3Xiangsihu College, Guangxi University for Nationalities, Nanning 530225, China
4Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning, 530006 China

*Corresponding authors: l.qf@163.com(Qifang Luo);yongquanzhou@126.com(Yongquan Zhou)

Abstract In this paper, to effectively reinforce the exploration and exploitation of Arithmetic optimization algorithm (AOA) and reasonably achieve their balance. A novel mathematical operator-based arithmetic optimization algorithm (MAOA) is proposed, firstly, we use mathematical symmetry operator and median operator to improve the exploitation and exploration ability of the population, respectively. Secondly, we use sine and cosine operator to effectively reinforce the exploration and exploitation of AOA algorithms and reasonably achieve their balance. Finally, the MAOA algorithm is used to solve the spherical mining spanning tree (sphere MST) and communication network problems. Experimental results show that the proposed MAOA has achieved excellent results in terms of global performance, accuracy, robustness, and convergence speed.

Keywords mathematical operator; arithmetic optimization algorithm; mining spanning tree; spherical MST; communication network problem; metaheuristic

1 Introduction

At present, with the continuous progress of natural science, the difficulty and depth of optimization problems in modern industrial are increasing. Optimization problems exist in many engineering fields, and have become one of the hotspots in the fields of medicine, engineering design and transportation [1], [2]. When faced with NP-hard problems in the field of engineering, traditional algorithms are usually unable to solve or the running time is unacceptable [3]. Considering these problems, meta-heuristic algorithms are widely used in many fields. Generally, optimization techniques are classified into two categories, namely meta-heuristic and deterministic
algorithms [4]. Because there's nothing random about this strategy, giving the same input to a given problem always yields the same output [5]. Deterministic algorithms may not be effective in solving multi-objective, large-scale, or multimodal problems [6]. However, the optimization problem gradually tends to be complex in the real world. The structure of engineering optimization problems is more complex, and the parameter variables have continuous or discrete forms. Practical problems have larger scale, higher accuracy and larger dimension, and the objective function is non-differentiable or nonlinear [7]. Therefore, meta-heuristics with few parameters and strong adaptability provide a way to solve current optimization problems [8], [9] because of their simple and understandable principle, easy implementation, low computational cost, and strong robustness [10].

Metaheuristic algorithms can be used for single objective or multi-objective optimization problems. They are divided into two important stages: exploration stage and development stage [11]. Exploration behavior is a process in which individuals perform a global search in the solution space and find multiple sub-solution spaces. In order for more high-quality solutions to be found, the exploitation behavior is the process of population individuals searching for high-quality solutions in the adjacent neighborhood of the second-level solution space [12]. Exploration and exploitation behaviors are usually cooperative in order to approach the optimal solution [13]. Four categories are typically used to categorize meta-heuristic algorithms [2].

(1) Metaheuristic algorithms based on evolution: Inspired by the evolution of species in nature, evolutionary algorithms mainly simulate the natural population evolution law to ensure that the proportion of high-quality individuals in the population is increasing. The earliest evolutionary algorithm is genetic algorithm (GA) [14], and other common evolutionary algorithms mainly include evolutionary strategy (ES) [15], differential evolution algorithm (DEA) [16], biogeography based optimization algorithm (BBOA) [17], evolutionary programming (EP) [18], immune algorithm (IA) [19] and so on.

(2) Metaheuristic algorithm based on swarm intelligence: Inspired by the collective behaviors and attributes of animals in nature, swarm intelligence algorithms mainly imitate the behavior of animals in hunting or other behaviors. Inspired by the living habits of different species, a variety of swarm intelligence algorithms emerged, in which each swarm is a group of organisms, and individuals in the swarm cooperate with each other to complete complex tasks in the real world.
The most representative algorithm is the particle swarm optimization (PSO) proposed by Kennedy et al. [20]. Ant colony optimization (ACO) and artificial bee colony (ABC) are also popular intelligent algorithms in such algorithms [21], [22]. The ant colony algorithm, proposed in 2006 by Dorigo et al., is inspired by ants looking for the shortest path located between their nest and the food, and imitates the behavioral process of ants looking for food. ABC, which is largely inspired by the social behavior of bees seeking food, classifies bees into three categories: employed bees, onlookers and scouts. Other popular swarm intelligence algorithms in recent years include ant lion optimization (ALO) [23], gray wolf optimization (GWO) [24], squirrel optimization algorithm (SSA) [25], whale optimization algorithm (WOA) [26], emperor penguin optimization algorithm (EPO) [27], artificial hummingbird algorithm (AHA) [28], chimp optimization algorithm (ChOA) [29], firefly algorithm (FA) [30], chicken swarm optimization algorithm (CSOA) [31] and marine predator algorithm (MPA) [32].

(3) Physics-based metaheuristic algorithms are proposed inspired by physical phenomena or laws in nature. In recent years, widely recognized by the academic community include simulated annealing algorithm (SAA) [33], gravitational search algorithm (GSA) [34], golden ratio optimization algorithm (GROA) [35], black hole algorithm (BHA) [36], sine cosine algorithm (SCA) [37]. In addition, the equilibrium optimizer (EO) [38] was proposed inspired by the controlled volume mass balance. Artificial electric field algorithm (AEFA) [39] is proposed inspired by Coulomb's Law of Electrostatic Force; Henley gas solubility optimizer (HGSO) [40] based on Henley's Law, etc., and they have been used by academics in various engineering problems.

(4) Metaheuristics based on human behavior that are currently widely recognized by the academic community include social learning optimization algorithm (SLOA) [41], volleyball premier league algorithm (VPLA) [42], teaching and learning optimization algorithm (TLBO) [43], election algorithm (EA) [44], brain storm optimization (BSO) [45], cultural evolution algorithm (CEA) [46] and so on. Moghdani et al. [47] proposed a multi-objective volleyball premier league algorithm, which has been applied to classical engineering problems. Kanwal and Iqbal et al. [48] developed an improved election optimization algorithm which has been applied to an Internet of Things cloud data processing. Xu and Peng et al. [49] proposed an improved instructional optimization algorithm which has been applied to remote fitness learning strategies.
Several representative meta-heuristics are shown in each category. AOA is widely popular due to its unique arithmetic properties. Short running time, simple principle and simple structure are the advantages of AOA [50], [51]. Because of the above advantages, Many engineering field problems have been solved using AOA, such as: optimizing deep neuro-fuzzy classifier [52], optimizing deep convolutional spiking neural network model [53], design problems of energy storage system in power grid [54], optimal power flow problem in DC networks [55], multi-objective problems in iot-enabled smart home [56], efficient text document clustering approach [57] and so on. But the no free lunch theorem states that no algorithm is applicable to all fields [58]. Therefore, researchers improve the existing algorithms or propose new meta-heuristic algorithms according to the characteristics of the current problems [59].

Tree is a common data structure in real life. It is widely used and come from graph theory [60]. In addition, the MST problem is often encountered in the field of power line network layout to obtain the most economical layout [62]. This was first considered in 1926 by Boruvka [63]. The famousness and importance of the MST originates from several aspects underlying: when faced with large graphs, efficient solutions exist, algorithms for solving the MST problem were proposed by Kruskal [64], Prim [65], Dijkstra [66] and Sollin [67]. This problem has been applied in real-life and is widely used in engineering fields such as traffic problems, drainage system design, telecommunication network design, and distribution systems [68]. Traditional methods used to solve MST problems are greedy strategies, which usually take a lot of time and can only obtain a MST. However, in most current problems, it is generally necessary to find a set of best or
second-best values in a relatively short time [61]. So, it is still necessary to find a method to solve the MST problem [69]. With the development of industry, the difficulty and scale of various problems continue to increase. In the real world, the earth that human beings live on is a sphere, so the research on the sphere becomes very practical significance. This paper further proves the effectiveness of the meta-heuristic algorithm and the sphere research in solving practical problems by studying the problem of laying optical cable in the world’s popular cities.

This paper, to effectively reinforce the exploration and exploitation of Arithmetic optimization algorithm (AOA) and reasonably achieve their balance, as well as the main contributions and motives for this study:

• A novel mathematical operator-based arithmetic optimization algorithm (MAOA) is proposed,
  • The mathematical symmetry operator and mathematical median operator have been proposed and to improve the exploitation and exploration ability of the population, respectively.
  • The sine and cosine operator to effectively reinforce the exploration and exploitation of AOA algorithms and reasonably achieve their balance.
  • The MAOA algorithm is used to solve the spherical mining spanning tree (sphere MST) and communication network problems.
  • Experimental results show that, the proposed MAOA has achieved excellent results in terms of global performance, accuracy, robustness, and convergence speed for solving the spherical MST problem.

The main contents of the other sections of this article are as follows: the related research work and spherical MST mathematical model in Section 2. In Section 3, arithmetic optimization algorithms are introduced and an improved arithmetic optimization algorithm for solving the spherical MST model is introduced. Section 4 experimental results are discussed and analyzed. In Section 5, examples of cable laying problems in popular cities around the world with different problem sizes are used to verify the performance of MAOA. Section 6 discusses the experimental results through fitness curves and variances, etc. The last section 7 is the summary and prospect of the future work.

2. Related works

This section introduces two types of work related to spherical minimum spanning trees:
minimum spanning tree problem, spherical geometry problem and spherical MST model. In addition, MST is a classic issue in graph theory, with important applications in the laying of optical cables between cities, the construction of highways, and the laying of water and gas pipelines. And it is widely recognized that meta-heuristic algorithms are employed to solve such problems. In the real world, the earth that human beings live on is a sphere, so the research on the sphere becomes very practical significance. This paper further proves the effectiveness of the meta-heuristic algorithm and the sphere research in solving practical problems by studying the problem of laying optical cable in the world's popular cities.

2.1. MST problem

The MST problem has been intensively studied in the last decades. However, it is more challenging and attractive for most researchers to solve more complex MST, such as the degree-constrained MST problem proposed by Narula and Ho et al. [70], the probabilistic MST problem introduced by Bertsimas et al. [71], the stochastic MST problem presented by Shiode and Ishii et al. [72], the problem of quadratic MST presented by Xu et al. [73], the problem of generalized MST introduced by Myung et al. [74], the problem of MST with degree constraints proposed by Cieslik et al. [75], the bi-objective MST problem introduced by Steiner et al. [76], The MST problem with conflict constraints and its variations considered by Zhang et al. [77], the marked MST problem described by Consoli et al. [78], the lower bounds and exact algorithms for the quadratic MST problem described by Pereira et al. [79], the min-degree constrained MST problem with fixed centrals and terminals introduced by Dias et al. [80], the angular constrained MST problem described by da Cunha et al. [81], the capacitated MST problem with arc time windows presented by Kritikos et al. [82], the spherical MST problem presented by Bi, Zhou, Zhang et al. [61], [69], etc. However traditional algorithms cannot effectively solve these problems at present. Next, this study proposes an improved AOA that uses mathematical methods: median operators, symmetric operators, sine and cosine operators to better find solutions to the MST. And it extends the problem to the sphere by applying fiber optic cables to major cities around the world.

2.2. Spherical Geometry

The principle of spherical geometry: first fix a point, and then the set of all points $r$ away from the point is a sphere. Where $r$ needs to be greater than or equal to 0 and is called the radius of
the sphere. And the center of the sphere is this fixed point, and does not belong to the sphere. $r = 1$ is a special case of the ball, called the unit ball. In analytic geometry, the set of all points $(X, Y, Z)$ with radius $R$ is denoted by equation (1).

$$X^2 + Y^2 + Z^2 = r^2$$

(1)

A plane is used to intercept a sphere, and the resulting intersection is a circle. The area of the obtained tangent circle is maximized when the tangent plane passes through the center of the sphere. But the circle cut on the center of the sphere is called the small circle [83]. Arcs smaller than semicircle are called inferior arcs. The shortest distance between two different cities on Earth is through the center of the Earth and the lower arc of the tangent circle of these two cities [84].

![Fig. 2 Sphere](image)

2.2.1. Representation of points on the sphere

Euclidean curves are one-dimensional objects whose positions along the path of three-dimensional curves can be described by t-parameters. Generally speaking, a vector function can represent any two points on a curve in Cartesian coordinates [85].

$$P(t) = (x(t), y(t), z(t))$$

(2)

In normal circumstances, the coordinate equation can be shown by the parameter $t$, where $t$ is between $0$ and $1$. For instance, the parametric form given by Eq. (3) allows the definition of a circle with $(0, 0)$ as its center in the $x$, $y$ plane [85].

$$\begin{align*}
x(t) &= r \cos(2\pi t) \\
y(t) &= r \sin(2\pi t) & 0 \leq t \leq 1 \\
z(t) &= 0
\end{align*}$$

(3)

Any point on the sphere can be defined by Eq. (4) parametric forms. An arbitrary curve on a three-dimensional space surface is a 2D variant represented by the parameters $m$ and $n$. The implicit representation of a surface is given, that is, the algebraic equation contains three variables $x$, $y$ and $z$. Find three rational parametric equations, each of which contains $m$ and $n$ parameters.
The three variables in Cartesian coordinates are represented by the parameters \( m \) and \( n \), and these two parameters \( m \) and \( n \) are between 0 and 1, respectively. Set the origin to the center of the sphere and set the radius of the sphere to \( r \), the spherical coordinate equation is given by Eq. (5).

\[
\begin{align*}
    x(m,n) &= r \cos(2\pi m) \sin(\pi n) \\
    y(m,n) &= r \sin(2\pi m) \sin(\pi n) \quad 0 \leq m \leq 1, \quad 0 \leq n \leq 1 \\
    z(m,n) &= r \cos(\pi n)
\end{align*}
\]  

(5)

Where the parameter \( m \) denoted the lines of longitude, and the parameter \( n \) denoted the lines of latitude on the spherical, as shown in Fig. 3. The parameters \( m \) and \( n \) can represent all points on the spherical surface, and the parameters \( m \) and \( n \) correspond uniquely to the position coordinates of the city. In this paper, the unit sphere is studied. Suppose that real-world engineering problems are not unit spheres, the value of coordinate \( x \), \( y \) and \( z \) are also expanded by the same factor.

2.2.2. The geodesic of a sphere

The arc of a section through the center of a sphere intersecting the surface of a sphere is a great circle [69]. Geodesic usually refers to the shortest distance between different cities on Earth [86]. The semi-circular arc connecting the north and south poles on the earth's surface is called the longitude. Each line of longitude has its own numerical value, called longitude. The orbit of a point on the Earth's surface as the Earth rotates is called a latitude line, and any latitude line is circular and parallel in pairs. The equator is the largest, and the closer the circle gets to the poles, the smaller the circle [84].

Suppose there are two points \((P_i, P_j)\) on the sphere. The geodetic line is shown in Fig. 4. First of all, the vectors formed by the center of the sphere and the two points are \( \vec{V}_i = x_i, y_i, z_i \)
and \( \vec{V}_j = x_j, y_j, z_j \), and the angle between them is represented by \( \theta(J) \). Secondly, the scalar product between them is expressed by Eq. 6.

\[
\vec{V}_i \cdot \vec{V}_j = |\vec{V}_i| |\vec{V}_j| \cos(\theta)
\]  

(6)

\[
\vec{V}_i \cdot \vec{V}_j = x_i x_j + y_i y_j + z_i z_j
\]  

(7)

Finally, \( \theta \) and \( \hat{d}_{p_1,p_2} \) are calculated using the following formula.

\[
\theta = \arccos \left( \frac{x_i x_j + y_i y_j + z_i z_j}{r^2} \right)
\]  

(8)

\[
\hat{d}_{p_1,p_2} = r \times \theta = r \times \arccos \left( \frac{x_i x_j + y_i y_j + z_i z_j}{r^2} \right)
\]  

(9)

In any two points on the sphere, the length from city \( P_i \) to \( P_j \) is equal to the geodesic from \( P_i \) to \( P_j \). Suppose the problem size is \( n \), the \( n \)-dimensional matrix is obtained by calculating the geodesic lines between different cities of the sphere. This matrix is represented as follows:

\[
D = \begin{bmatrix}
\hat{d}_{11} & \hat{d}_{12} & \cdots & \hat{d}_{1n} \\
\hat{d}_{21} & \hat{d}_{22} & \cdots & \hat{d}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{d}_{n1} & \hat{d}_{n2} & \cdots & \hat{d}_{nn}
\end{bmatrix}
\]  

(10)

where \( \hat{d}_{i,j} \) denotes geodesic distance between city \( P_i \) and \( P_j \). Where \( \hat{d}_{i,i} = \infty \) means that point \( P_i \) cannot choose a route to reach itself.

2.3. Spherical MST Mathematical Model

An undirected graph can be represented by \( G(V,E) \), where \( V \) is a nonempty set for the 2D MST problem, called the vertex set. \( E \) is the set of unordered binary groups formed by the elements of \( V \), called edge sets. The set of nodes, \( V \), is a one-dimensional array, and the set of
relationships between cities, $E$, is a two-dimensional data set, also known as the adjacency matrix. An edge between any two nodes has a corresponding weight $w$. $x$ is a two-dimensional matrix, where $x_{ij} = 0$ or 1. If $x_{ij} = 1$ representing the edge between city $i$ and $j$ is selected, otherwise, it is not selected. The spherical MST problem can be modeled as follows:

$$
\min f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_{ij},
$$

subject to:

$$
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij} = n-1,
$$

$$
\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S|-1, \forall S \subset V, |S| \geq 2,
$$

$$
x_{ij} \in \{0,1\}, \quad i, j = 1,2,\ldots,n,
$$

where the Eq. (12) restricts the final result to a tree structure. Moreover, in order to ensure that the solution set during the operation of the algorithm does not generate a closed loop, Eq. (13) is added as a constraint.

For the 3D spherical MST problem, the node $P = \{P_1, P_2, \ldots, P_n\}$ is a set of points on a sphere, where $n$ is the problem size. Each node is denoted by $p_i = (x_i, y_i, z_i)$. The set of geodesic data for all nodes on the sphere is denoted by $A = \{d_{pj} | p_i, p_j \in V\}$. Calculate the lengths $\hat{d}_{ij}$ of all geodesics by Eq. (9). Eq. (10) can be used to construct the $n \times n$ two-dimensional symmetric distance matrix $D$, where $x_{ij} = 0$ or 1, represents selected and unselected. Finally, the mathematical model with Eq. (15) as spherical MST is obtained

$$
\min f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \hat{d}_{ij} x_{ij}
$$

Similarly, constraints 12-14 are applied to Eq. (15).

3. Our proposed method

3.1. Arithmetic optimization algorithm

Arithmetic optimization algorithm (AOA) is a metaheuristic optimization algorithm based on population recently proposed by Abualigah et al. [88]. It is mainly based on the background of arithmetic operators in mathematics. Due to the properties of meta-heuristics, this algorithm can solve the optimization problem without gradient information, and the operation process of AOA
first goes through initialization parameters and then optimization through exploration and exploitation stages [88].

3.1.1. Inspiration

The entire area of mathematics relies heavily on arithmetic. And usually number theory also provides some basic support for solving problems in real life. AOA is designed to take into account and is inspired by the rules of arithmetic operators in the field of mathematics.

3.1.2. Initialization phase

The matrix \( X \) represents the initialization population of the algorithm, which are randomly generated. As the AOA continues to operate, the information of the population matrix continues to improve. The algorithm also records the best solution during the run.

\[
X = \begin{bmatrix}
  x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,n} \\
  x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\
  x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,n}
\end{bmatrix}
\]  

(16)

The exploitation behavior or the exploration behavior should be chosen during each iteration of the AOA. Therefore the math optimizer accelerated (MOA) is an important coefficient in AOA, and its iterative formula is shown in Eq.17.

\[
MOA(CurrentIte) = Min + CurrentIte \times \left( \frac{Max - Min}{MaxIter} \right)
\]

(17)

where \( CurrentIte \) represents the iteration number and is between 1 and \( MaxIter \). Min and Max denote the upper and lower bounds during the iteration.

Furthermore, the iterative formulation of the math optimizer probability (\( MOP \)) in AOA.

\[
MOP(CurrentIte) = 1 - \frac{CurrentIte^{\alpha}}{MaxIter^{\alpha}}
\]

(18)

Where \( CurrentIte \) indicate the number of iterations and \( MaxIter \) represents the maximum number of iterations of the algorithm. \( \alpha \) a is a constant parameter that determines the precision of exploitation. Abualigah et al. mention setting it to 0.5 in the article. [88].

3.1.3. Exploration phase
The exploratory behavior of the AOA is introduced in this section. Depending on the
operators in number theory, calculations using the multiplication (M) or division (D) operator can
obtain lots of highly concentrated sub-optimal solutions or some optimal solutions. However,
compared to the subtraction (S) and addition (A) operators, the M and D operators are more
difficult to find the optimal solution objective due to their high dispersion. AOA conducts
continuous random search in the solution space through D and M main search strategies.

The two major operators for AOA to find better solutions in the exploration stage are D and
M, which are modeled in Eq. (19). And the secondary phase is controlled by the accelerated math
optimizer (MOA). If the value of MOA is less than the random number \( r_1 \) (\( r_1 \) is a random number),
AOA enters this stage. Next, the comparison with 0.5 is made by generating a random number \( r_2 \)
between 0 and 1. Decide whether to choose the D or M operator.

\[
x_{i,j}(CurrentIter + 1) = \begin{cases} 
  \text{best}(x_j) - MOP \times (UB_j - LB_j) \times \mu + LB_j, & r_2 < 0.5 \\
  \text{best}(x_j) + MOP \times (UB_j - LB_j) \times \mu + LB_j, & \text{otherwise}
\end{cases}
\]

(19)

where \( x_{i,j}(CurrentIter + 1) \) indicate the \( j \)th position of the \( i \)th solution in the next
iteration, and \( \text{best}(x_j) \) indicate that the \( j \)th position has so far been the best-obtained
solution. \( \ell \) indicate a quite small value. The algorithm's upper and lower bounds are indicated by
the letters UB and LB, respectively. In the studies of Abualigah et al., \( \mu \) is set to 0.5 [88].

3.1.4. Exploitation phase

This section introduces AOA’s exploitation strategy. Calculations utilizing the subtraction (S)
or addition (A) operators result in highly packed data. However, as opposed to other operators (D
and M), because of their minimal dispersion, these operators (S and A) may readily approach the
target. In AOA, based on these two primary search techniques, S and A operators randomly use
the search space to discover the nearly ideal solution. Furthermore, the exploitation operators
worked to help the exploitation stage by improving communication between them during the
exploitation phase. The value of MOA determines whether the development phase is selected or
not in AOA.

\[
x_{i,j}(CurrentIter + 1) = \begin{cases} 
  \text{best}(x_j) - MOP \times (UB_j - LB_j) \times \mu + LB_j, & r_3 < 0.5 \\
  \text{best}(x_j) + MOP \times (UB_j - LB_j) \times \mu + LB_j, & \text{otherwise}
\end{cases}
\]

(20)

Based on this equation, the first operator (S) is conditioned by \( r_3 < 0.5 \) (\( r_3 \) is a random
number). When \( r_3 \) is greater than 0.5, the A is neglected. Otherwise, the A operator will be used
for position update and the $S$ operator will become invalid.

### 3.1.5. Pseudo-code and intuitive process for AOA

**Algorithm 1** Pseudo-code of the AOA

- Initialize the AOA parameters $\alpha$ and $\mu$.
- Initialize the population matrix $N_i (i = 1, ..., n)$.

while ($CurrentIter < MaxIter$) do

  The fitness value of each individual in the population is calculated.
  
  Record the best solution yet.

  The values of $MOA$ and $MOP$ are calculated by Eqs. (17) and (18).

  for ($i = 1 : N$) do
      for ($j = 1 : Dim$) do
          Generate random numbers $r_1$, $r_2$, and $r_3$ between 0 and 1.
          
          if $r_1 > MOA$ do
              if $r_2 > 0.5$ do
                  (1) The $D$ operator is used for operations ("÷")
                  
                  Calculate the $i$th position of the individual using the $D$ operator through Eq. (19).
              
              else
                  (2) The $M$ operator is used for operations ("×")
                  
                  Calculate the $i$th position of the individual using the $M$ operator through Eq. (19).
          
          else
              else
                  if $r_3 > 0.5$ then
                      (3) The $S$ operator is used for operations ("−")
                      
                      Calculate the $i$th position of the individual using the $S$ operator through Eq. (20).
                  
                  else
                      (4) The $A$ operator is used for operations ("+")
                      
                      Calculate the $i$th position of the individual using the $A$ operator through Eq. (20).
                  
          end if
      
  end for

  $CurrentIter = CurrentIter + 1$

end while

Return the solution ($X_{best}$).
3.2. The MAOA Algorithm

In this section, an optimization algorithm called arithmetic optimization algorithm based on mathematical strategy is introduced. It is used to solve the spherical MST problem with different problem sizes. For the traditional AOA position update based on the historical optimal individual, the population diversity will not be sensitive to the performance of the algorithm, and the development stage is easy to fall into local optimal. The position update of AOA is affected by upper and lower bounds, and the stride length is too large in some applications. MAOA avoids these shortcomings by introducing sine and cosine operators and mathematical median operators. First of all, adding sine and cosine operators to form vector triangles between population individual position and historical optimal individual position can improve AOA’s ability to jump out of local optimal. Secondly, the convergence accuracy of AOA can be effectively improved by introducing the median operator in the later stage. Finally, mathematical symmetry operators are introduced to help balance AOA exploration and development behavior. MAOA consists of traditional arithmetic optimization algorithm (AOA), sine and cosine operators, mathematical median operators and mathematical symmetry operators.
3.2.1. Sine and cosine operators

The update of AOA particle position was added, subtracted, multiplied and divided by the historical optimal individual. However, the population diversity could still play a role, and the exploration ability needed to be provided. In order to solve this problem, the sine and cosine operator is introduced, that is, AOA can be iteratively optimized through a variety of mathematical formulas. The following is the position update formula:

\[ x_i\left(CurrentIter + 1\right) = X_i + \sin(r \cdot 2\pi)(r + \cos(r \cdot 2\pi)(best(X) - X_i)) \]  \hspace{1cm} (21)

where \( x_i(\text{CurrentIter} + 1) \) indicate the position of the solution in the following iteration, and \( \text{best}(x) \) denotes the best solution obtained at the present time. \( r \) denotes a random number between 0 and 1, \( X_i \) is the position of the \( ith \) individual.

3.2.2. Mathematical median operator

In the operation of MAOA algorithm, a new operator is introduced on the basis of the original one by means of the median operator. Moreover, this operator can improve the convergence accuracy of the original algorithm and retain the excellent characteristics of AOA. This operator is inspired by the idea of finding the median between two points in mathematics and is used in arithmetic optimization algorithms. The updated formula is as follows:

\[ x_i(\text{CurrentIter} + 1) = \frac{\text{best}(X) + X_i}{2} \]  \hspace{1cm} (22)

where \( x_i(\text{CurrentIter} + 1) \) indicate the position of the solution in the following iteration, and \( \text{best}(x) \) denotes the best solution obtained at the present time. \( X_i \) is the position of the \( ith \) individual.

3.2.3. Mathematical symmetric operator

In geometric mathematics, there is an idea of symmetry which plays an important role. In mathematics, point A and point B are symmetric about the origin, that is, line segment AB passes through the origin, and the distance between point A and the origin is numerically equal to the distance between point B and the origin. In AOA, A mathematical symmetry operator can improve the algorithm's capacity for exploration and is useful for striking a balance between development and exploration. The updated formula is as follows:

\[ x_i(\text{CurrentIter} + 1) = \text{best}(X) + r \times (\text{best}(X) - X_i) \]  \hspace{1cm} (23)

where \( x_i(\text{CurrentIter} + 1) \) indicate the position of the solution in the following iteration, and
\( \text{best}(x) \) denotes the best solution obtained at the present time. \( x_i \) is the position of the \( i \)th individual. \( r \) represents a random number between 0 and 1.

### 3.2.4. Pseudo-code of the MAOA

**Algorithm 2** Pseudo-code of the MAOA algorithm

The parameters \( \alpha \) and \( \mu \) of MAOA are initialized.

Initialize the population matrix \( N_i \) (\( i = 1, ..., n \)).

**while** (\( \text{CurrentIter} < \text{MaxIter} \)) **do**

The fitness value of each individual are calculated.

Record the best solution yet.

The \( \text{MOA} \) and \( \text{MOP} \) are calculated by Eqs. (17) and (18).

**for** (\( i = 1 : N \)) **do**

Calculate the \( i \)th position of the individual using the rule in Eq. (21).

**for** (\( j = 1 : \text{Dim} \)) **do**

Generate random numbers \( r_1, r_2, \) and \( r_3 \) between 0 and 1.

**if** \( r_1 > \text{MOA} \) **do**

**if** \( r_2 > 0.5 \) **do**

(1) The \( D \) operator is used for operations ("\( / \)"")

Calculate the \( i \)th position of the individual using the \( D \) operator through Eq. (19).

else

(2) The \( M \) operator is used for operations ("\( \times \)"")

Calculate the \( i \)th position of the individual using the \( M \) operator through Eq. (19).

**end if**

else

**if** \( r_3 > 0.5 \) **then**

(3) The \( S \) operator is used for operations ("\( - \)"")

Calculate the \( i \)th position of the individual using the \( S \) operator through Eq. (20).

else

(4) The \( A \) operator is used for operations ("\( + \)"")

Calculate the \( i \)th position of the individual using the \( A \) operator through Eq. (20).

**end if**

**end if**

**end for**

Calculate the \( i \)th position of the individual using the rule in Eq. (22).

Calculate the \( i \)th position of the individual using the rule in Eq. (23).
end for

\[ \text{CurrentIter} = \text{CurrentIter} + 1 \]

end while

Return the solution \((X_{\text{best}})\).

### 3.3.1. MST Based on Prüfer Coding

Prüfer coding is the way to mark rootless trees. Its idea comes from Cayley's theorem, which indicate that in a complete graph with \(n\) nodes, there exist \(n^{n-2}\) different trees. In combinational mathematics, Prüfer coding is a sequence transformed from a tree with labeled vertices, and a tree with \(n\) points yields a Prüfer sequence of length \(n - 2\). Of course, it can also be understood as a bijection between the spanning tree of the complete graph and the sequence. A Prüfer sequence corresponds to a unique unrooted tree, and an unrooted tree corresponds to a unique Prüfer sequence. Prüfer is not generally used to maintain tree structures and is often used for combinatorial counting problems.

### 3.3.2. Generation of Prüfer sequence

Generating the Prüfer coding is an iterative process, assuming that a tree is constructed with seven nodes. The topology of this tree and the process of generating Prüfer sequence are shown in Fig.7. The following four steps make up the main portion of the process of transforming a tree into a Prüfer sequence:

1. Find the leaf node \(i\) on tree \(T\) that has the smallest value
2. Suppose that the only node connected to node \(i\) is \(j\), and then \(j\) is the encoding of the first number, and the sequence of encoding from left to right.
3. The edges \((i, j)\) and node \(i\) are removed, and a tree of \(n\) nodes is obtained.
4. This operation is repeated, and the algorithm is finished when there are two points left in the tree.

Through the above steps, the unique Prüfer sequence can be obtained, which is the permutation of \(n-2\) numbers between 1 and \(n\).

![Prüfer Code: 2 2 3 3 2](image)

![Prüfer Code: 2 2 3 3 2](image)

![Prüfer Code: 2 2 3 3 2](image)

**Fig. 7** Prüfer sequence
Finally, the Prüfer sequence is: 2, 2, 3, 3, 2.

### 3.3.3. The generation of spanning trees is based on Prüfer sequence

According to the construction principle of Prüfer sequence and the primary transformation from Prüfer sequence to tree is mainly divided into the following four steps:

1. Finds the minimum node $i$ that is not in the Prüfer sequence of the set, and store the node $n - 2$ at the end of the Prüfer sequence.
2. Connect this node with the first number of Prüfer sequence.
3. The node $i$ in the set and the first number in Prüfer sequence are deleted.
4. This operation is repeated, and the algorithm is complete when there is no node in the Prüfer sequence.

Through the above steps, the unique spanning tree $T$ of Prüfer sequence can be obtained.

### 3.3.4. Code design

In this study, assuming that there are $n$ nodes on the sphere, all nodes are numbered starting from 1. Thus, a tree of $n$ nodes is uniquely represented by a Prüfer sequence of $n - 2$ integers. Then the dimension of each individual in the AOA is $n - 2$. And the variables in the individual are rounded to obtain a sequence of integers.

Suppose there is an individual is represented by

$$X_i : (3.11, 2.02, 3.21, 5.99, 2.22)$$

The Prüfer sequence obtained by rounding the $X_i$ sequence is as follows:

$$X_i \rightarrow X_j : (3, 2, 3, 6, 2)$$

According to Prüfer sequence $X_j$, the spanning tree is as follows:

![Fig. 8 spanning tree](image)

### 3.3.5. The pseudo-code for calculating the unique spanning tree through the Prüfer sequence
Pseudo-code that computes a unique spanning tree through a Prüfer sequence

Initialize the tree $T$ and Prüfer sequence $P$.

Store the node $n - 2$ at the end of the set $P$.

Finds the minimum nodes that are not in set $P$ and put them in set $Q$.

**While** (there are elements in the set $P$) **do**

  Sort the set $Q$ according to the principle of small element first.

  Let $i$ be the first element on the left in the set $Q$.

  Let $j$ be the first element on the left in the set $P$.

  Connect node $i$ to node $j$ and them to tree $T$.

  Remove node $i$ from set $Q$.

  Remove node $j$ from set $P$.

  if (node $j$ is not in set $P$) **do**

    Put node $j$ into set $Q$.

  **end if**

**end while**

Output the tree $T$

---

3.3.6. The pseudo-code for calculating the unique Prüfer sequence through the spanning tree

Pseudo-code that computes a unique Prüfer sequence through a spanning tree.

Initialize a tree $T$.

Initialize a Prüfer sequence $P$ with $n$ nodes.

While (there are more than two elements in the tree $T$)

  Find the leaf node $i$ on tree $T$ that has the smallest value

  Let the unique node $j$ connected to $i$ be the first digit of the encoding

  the sequence of encoding from left to right.

  Node $i$ and the edge from $i$ to $j$ are deleted

End while

Output the Prüfer sequence $P$

---

4. Experimental results and analysis
In this section, to evaluate the validity and superiority of the MAOA for solving the spherical MST model. In order to verify the capability of MAOA for solving spherical MST problems, a variety of scale problems were used for benchmarking. The main contents of this section are as follows:

1. The problem scale and algorithm parameters are shown in the experimental setup section.

2. The experimental results of 25~400 dimensions are presented and compared in the small-scale section.

3. The experimental results of 500~1000 dimensions are presented and compared in the medium-scale section.

4. The experimental results of 1000~2000 dimensions are presented and compared in the large-scale section.

4.1. Experimental setup

In this experiment, the unit sphere with \( n = 25, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500 \) and 2000 points (cities) is simulated respectively. All node data will not change during the run, and the same points will be used for all experiments. The experimental results in cities with a size less than or equal to 400 are from the literature Bi and Zhou et al. [61], and between 500 and 1000 nodes of data and results from the literature Zhang et al. [69]. In order to avoid the randomness of these algorithms, the intelligent algorithm involved in the experiment was run independently for 30 times.

MAOA is compared with the AOA, bio-inspired bare bones mayfly algorithm (BBMA) [69], artificial electric field algorithm with inertia and repulsion (IRAFA) [61], PSO [20], GA [14], GWO [24], SMA [87], DE [16] and ACO [21] in best, worst, mean, standard deviation and Friedman test. This paper effectively proves the performance of MAOA by compared the convergence curves, the ANOVA test, the average fitness values of 30 independent runs and the Wilcoxon rank-sum non-parametric statistical test.

In order to ensure the authenticity and reliability of the experimental results, MAOA will use the same system configuration as other algorithms. MATLAB 2018a was used to run the experiment, the processor on the device was Intel Core(TM) i7-9700U and the frequency was 3.00 GHz, the operating system was Windows10 and the RAM was 16GB.
Table 1 Parameter variables involved in all algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAOA</td>
<td>$\alpha$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$\mu$</td>
<td>0.5</td>
</tr>
<tr>
<td>AOA</td>
<td>$\alpha$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$\mu$</td>
<td>0.5</td>
</tr>
<tr>
<td>PSO</td>
<td>Cognitive and social constant</td>
<td>$C_1=2$, $C_2=2$</td>
</tr>
<tr>
<td></td>
<td>Inertia weight</td>
<td>Gradient down from 0.9 to 0.1</td>
</tr>
<tr>
<td>GA</td>
<td>Mutation probability</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Crossover probability</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>Elite fraction</td>
<td>0.05</td>
</tr>
<tr>
<td>DE</td>
<td>Scaling factor</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Crossover probability</td>
<td>0.5</td>
</tr>
<tr>
<td>ACO</td>
<td>The importance of heuristic factors</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The importance of pheromones</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>Pheromone evaporation rate</td>
<td>1</td>
</tr>
<tr>
<td>GWO</td>
<td>Convergence parameter (a)</td>
<td>Linear reduction from 2 to 0</td>
</tr>
<tr>
<td>SMA</td>
<td>Foraging success probability</td>
<td>0.3</td>
</tr>
<tr>
<td>BBMA</td>
<td>Positive attraction constants</td>
<td>$\alpha_1=1$, $\alpha_2=1.5$</td>
</tr>
<tr>
<td></td>
<td>Visibility constants</td>
<td>$\beta=1$</td>
</tr>
<tr>
<td></td>
<td>Nuptial dance constants</td>
<td>$d=1$</td>
</tr>
<tr>
<td></td>
<td>Random walk constants</td>
<td>$f_l=0.1$</td>
</tr>
<tr>
<td>IRAEFA</td>
<td>Coulomb’s constant</td>
<td>500</td>
</tr>
</tbody>
</table>

4.2. The experimental results of the small-scale

In this section, the spherical MST is solved with 25, 50, 100, 200, 300 and 400 points (cities) of problem scale. The accuracy and performance of all algorithms are compared with other algorithms. The following results are obtained by running all algorithms for 30 times. Table 2 records the data results of these algorithms for 25–400 dimensions. Fig.9 displays the convergence curves in these four scenarios, Fig.10 shows the findings of the analysis of variance, Fig.11 shows the fitness values of the 30 runs, and Fig.12 shows the spherical MST for this experiment.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Std</td>
<td>0.8548</td>
<td>1.0401</td>
<td>1.6687</td>
<td>1.2759</td>
<td>2.3995</td>
<td>1.2658</td>
<td>1.0752</td>
<td>1.0202</td>
<td>1.3235</td>
<td>0.9210</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>2.12E-06</td>
<td>0.2059</td>
<td>2.37E-05</td>
<td>1.73E-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>56.7224</td>
<td>51.7636</td>
<td>51.5066</td>
<td>61.6177</td>
<td>57.8262</td>
<td>47.6913</td>
<td>30.8955</td>
<td>34.817</td>
<td>52.5764</td>
<td>28.0005</td>
</tr>
<tr>
<td>Std</td>
<td>1.0267</td>
<td>1.6024</td>
<td>3.0896</td>
<td>2.3682</td>
<td>1.5102</td>
<td>3.0873</td>
<td>1.8417</td>
<td>1.7546</td>
<td>2.1445</td>
<td>2.0232</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.79E-05</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>125.2646</td>
<td>115.9142</td>
<td>105.388</td>
<td>122.6827</td>
<td>115.5757</td>
<td>102.8094</td>
<td>66.8479</td>
<td>69.2455</td>
<td>127.2740</td>
<td>52.3542</td>
</tr>
<tr>
<td>Std</td>
<td>131.6558</td>
<td>125.0204</td>
<td>122.9785</td>
<td>141.3354</td>
<td>132.9414</td>
<td>127.4766</td>
<td>79.0415</td>
<td>82.8067</td>
<td>136.2751</td>
<td>64.4948</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>269.3156</td>
<td>251.2678</td>
<td>241.6883</td>
<td>268.7868</td>
<td>266.3731</td>
<td>241.7519</td>
<td>166.8751</td>
<td>158.8478</td>
<td>271.0750</td>
<td>123.6745</td>
</tr>
<tr>
<td>Std</td>
<td>280.3753</td>
<td>265.8108</td>
<td>269.7025</td>
<td>296.2953</td>
<td>277.069</td>
<td>275.1255</td>
<td>197.8590</td>
<td>186.5355</td>
<td>284.4685</td>
<td>146.5015</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>404.8393</td>
<td>381.7567</td>
<td>390.3274</td>
<td>422.2974</td>
<td>416.0361</td>
<td>391.2315</td>
<td>289.9457</td>
<td>275.7485</td>
<td>412.3600</td>
<td>227.8733</td>
</tr>
<tr>
<td>Std</td>
<td>421.2182</td>
<td>410.6107</td>
<td>425.3143</td>
<td>448.4215</td>
<td>425.4912</td>
<td>417.1096</td>
<td>328.7072</td>
<td>320.7405</td>
<td>428.0944</td>
<td>264.6494</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>559.3794</td>
<td>534.0305</td>
<td>535.9726</td>
<td>562.2246</td>
<td>558.2885</td>
<td>545.4407</td>
<td>422.8765</td>
<td>387.1640</td>
<td>552.2144</td>
<td>345.8756</td>
</tr>
<tr>
<td>Std</td>
<td>579.4299</td>
<td>561.5491</td>
<td>563.4369</td>
<td>601.2813</td>
<td>576.4934</td>
<td>577.0593</td>
<td>472.5275</td>
<td>438.9316</td>
<td>577.3100</td>
<td>428.3214</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Experimental results for the ten algorithms for 25~400 cities.
Fig. 9 The convergence curves of algorithms for 25-400 points.
Fig. 10 The ANOVA test for 25–400 points.
MAOA is compared with other algorithms in 25–400 cities respectively. The running results are shown in Table 2. Compared to IRAEFA, the convergence accuracy of the MAOA suggested in this research for 25 cities is subpar, but the standard deviation is superior than the algorithm. At the same time, the findings of the Friedman test, the best, worst, and mean values are superior to those of alternative methods. Among the 50 cities, the standard deviation of MAOA is 0.9210, which is lower than ACO, but the best, worst and average values are superior to other algorithms.
When the number of cities increases to 100, 200, 300 and 400, MAOA is better than other algorithms in terms of best, mean values and Friedman, and its standard deviation is also better than most algorithms. The disadvantage is that its standard deviation is lower than ACO in some cases. As the scale of the problem continues to complexity, MAOA's Friedman ranking continues to be the best. Table 2 shows the results of Wilcoxon rank sum test at low dimensions. The $p$ value is very small, so it can be seen that MAOA is significantly superior to other algorithms.

Fig. 9 shows the convergence curve of all algorithms. It can be seen from Fig.9 that when running for about 100 generations, it is better than most algorithms and still searching for the optimal solution. The optimization accuracy of DE, AOA and ACO algorithms is relatively low. It can be seen that when the number of cities is less than 400, the convergence accuracy and speed of MAOA achieve good results with the scale of the problem continues to complexity.

Fig.10 shows the variance graph of all algorithms. It can be seen that MAOA, ACO and DE have high stability in most cases, while GWO and PSO have poor stability. Fig. 11 shows the optimal value of 30 runs. It can be clearly seen that MAOA is superior to other algorithms in most cases. When other algorithms become very slow to update, MAOA algorithm can still continue to search for optimization. With the scale of the problem continues to complexity, MAOA outperforms other algorithms. Fig.12 displays the spherical MST paths for 25-400 cities.

### 4.3. The experimental results of the medium-scale

In this section, the proposed algorithm MAOA is simulated with other nine algorithms at 500~1000 cities respectively. Table 3 records the data results of these algorithms in 500~1000 dimensions. The use of bold denotes that it is the optimal solution for the nine algorithms. Fig. 13 shows the convergence curves of the six problem sizes, Fig.14 shows the results of the analysis of variance of the six problem sizes, and Fig.15 shows the average fitness values of the 30 runs. And the spherical MST is listed in Fig.16.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Best</td>
<td>712.2368</td>
<td>693.7499</td>
<td>700.4049</td>
<td>722.1846</td>
<td>710.3885</td>
<td>690.8498</td>
<td>526.7288</td>
<td>710.6879</td>
<td>470.3627</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>721.7657</td>
<td>703.2983</td>
<td>714.2092</td>
<td>743.3494</td>
<td>721.7850</td>
<td>714.6237</td>
<td>551.4301</td>
<td>721.9112</td>
<td>505.7130</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>/</td>
</tr>
</tbody>
</table>

Table 3: Experimental result for the nine algorithms for 500 - 1000 cities.
<table>
<thead>
<tr>
<th>h-Value</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>/</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>Best</td>
<td>845.7269</td>
<td>836.9045</td>
<td>830.4673</td>
<td>883.8227</td>
<td>859.0083</td>
<td>834.3700</td>
<td>661.5117</td>
<td>850.6166</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>874.7280</td>
<td>861.8101</td>
<td>875.6180</td>
<td>910.6815</td>
<td>881.7119</td>
<td>866.6941</td>
<td>720.9847</td>
<td>880.3411</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>865.3516</td>
<td>851.2513</td>
<td>856.0039</td>
<td>899.1887</td>
<td>871.8175</td>
<td>854.6719</td>
<td>687.1937</td>
<td>872.9360</td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.92E-06</td>
<td>1.73E-06</td>
<td>/</td>
</tr>
<tr>
<td>h-Value</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>700</td>
<td>Best</td>
<td>997.0200</td>
<td>996.3382</td>
<td>991.1713</td>
<td>1019.2579</td>
<td>1014.0621</td>
<td>991.4004</td>
<td>795.8841</td>
<td>1015.7392</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1028.7226</td>
<td>1011.0239</td>
<td>1028.9777</td>
<td>1070.9165</td>
<td>1033.3789</td>
<td>1022.5838</td>
<td>887.1201</td>
<td>1042.4202</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1017.4256</td>
<td>1003.0214</td>
<td>1009.3892</td>
<td>1047.1824</td>
<td>1024.7324</td>
<td>1010.1426</td>
<td>826.7429</td>
<td>1030.1895</td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>h-Value</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>800</td>
<td>Best</td>
<td>1159.6173</td>
<td>1128.1320</td>
<td>1129.6123</td>
<td>1188.4263</td>
<td>1153.7342</td>
<td>1137.4899</td>
<td>920.5137</td>
<td>1131.0289</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1185.3013</td>
<td>1163.4454</td>
<td>1175.3558</td>
<td>1224.0369</td>
<td>1185.4567</td>
<td>1185.2648</td>
<td>1003.667</td>
<td>1167.2801</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1176.5963</td>
<td>1153.9216</td>
<td>1157.8010</td>
<td>1204.1616</td>
<td>1171.0356</td>
<td>1166.0980</td>
<td>961.2036</td>
<td>1159.4564</td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>2.61E-04</td>
<td>1.73E-06</td>
<td>/</td>
</tr>
<tr>
<td>h-Value</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>900</td>
<td>Best</td>
<td>1364.4897</td>
<td>1356.2774</td>
<td>1346.2868</td>
<td>1372.1236</td>
<td>1365.4940</td>
<td>1357.1972</td>
<td>1072.67</td>
<td>1369.3126</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1380.5913</td>
<td>1366.5276</td>
<td>1373.0494</td>
<td>1393.6464</td>
<td>1376.8187</td>
<td>1376.3410</td>
<td>1167.294</td>
<td>1380.7887</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1374.7512</td>
<td>1360.9010</td>
<td>1364.1205</td>
<td>1384.5569</td>
<td>1372.1504</td>
<td>1367.2020</td>
<td>1107.199</td>
<td>1376.5435</td>
</tr>
<tr>
<td>Std</td>
<td>3.7688</td>
<td>2.8044</td>
<td>7.3119</td>
<td>6.2967</td>
<td>2.4914</td>
<td>5.1026</td>
<td>28.7825</td>
<td>3.2142</td>
<td>7.5412</td>
</tr>
<tr>
<td>Rank</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>8.18E-05</td>
<td>1.73E-06</td>
<td>/</td>
</tr>
<tr>
<td>h-Value</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>Best</td>
<td>1476.3932</td>
<td>1452.4452</td>
<td>1443.5458</td>
<td>1500.2107</td>
<td>1464.5448</td>
<td>1454.3861</td>
<td>1167.5380</td>
<td>1481.3639</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1499.3696</td>
<td>1475.0927</td>
<td>1490.7540</td>
<td>1542.2565</td>
<td>1494.2965</td>
<td>1496.3472</td>
<td>1344.461</td>
<td>1511.6308</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1485.2603</td>
<td>1461.4399</td>
<td>1470.4062</td>
<td>1520.8309</td>
<td>1481.8861</td>
<td>1477.4963</td>
<td>1248.647</td>
<td>1498.4057</td>
</tr>
<tr>
<td>Rank</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>p-Value</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>1.73E-06</td>
<td>3.11E-05</td>
<td>1.73E-06</td>
<td>/</td>
</tr>
<tr>
<td>h-Value</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 13 The convergence curves of algorithms for 500–1000 points.

The ANOVA test for 500 points

The ANOVA test for 600 points
Fig. 14 The ANOVA test for 500–1000 points.
Table 3 shows that in the 500 and 600 cities, different algorithms have significant differences in search ability. The table shows that even though MAOA’s standard deviation is not ideal, the best, worst, and average values all come out on top. In addition, Friedman has the worst GA ranking in the test and MAOA’s Friedman ranked best. Fig.13 shows the convergence curves of these algorithms in the 500 and 600 cities. Fig.14 shows the ANOVA results of each algorithm under 500 and 600 cities. The figure shows that the stable results of MAOA are also within the allowed range. Fig.15 and Fig.16 respectively show the fitness values and MST of these
algorithms run for 30 times in 500 and 600 cities.

Table 3 displays the simulation results of these algorithms in 700 and 800 cities. The table shows that MAOA performs well in best, worst, and average values, but the standard deviation is not satisfactory. The experiment shows that GA's comprehensive performance is the worst. DE and ACO get the optimal value of standard deviation in the case of 700 and 800 cities respectively, and MAOA gets the optimal value in other cases. Taking all the metrics together, MAOA ranked the best in Friedman's test. Fig.13 shows the convergence curves of these algorithms in 700 and 800 cities, and it is clear that MAOA has the fastest and most accurate convergence. Fig. 14 shows the analysis of variance results of these algorithms in the case of 700 and 800 cities, and demonstrates that the MAOA stability results are also within the permitted range. Fig. 15 and Fig. 16 show the fitness values and spherical MST of these algorithms run for 30 times in 700 and 800 cities, respectively.

The cases of 900 and 1000 cities are also shown in Table 3. At 900 cities, MAOA performed poorly in terms of best, worst and mean values, but the standard deviation of 7.5412 was better than BBMA's 28.7825. When increasing to 1000 cities, MAOA outperforms other algorithms in terms of best, worst, mean values and standard deviation, and is also optimal in Friedman test. Fig. 13 and Fig. 14 respectively show the analysis of convergence curves and variance results of these algorithms in the case of 900 and 1000 cities. It is evident that MAOA's convergence speed and accuracy are at their best, and its stability findings are within acceptable bounds. Fig. 15 and Fig. 16 show the fitness values and spherical MST of these algorithms run for 30 times in 900 and 1000 cities respectively.

4.3. The experimental results of the large-scale

After the above comparison, MAOA showed good performance. However, in the face of the increasing complexity of problems in the real world, in this section, MAOA will be tested at a higher problem size, where the city size is 1500 and 2000. Table 4 records the best, worst, mean and standard deviation, Wilcoxon rank and nonparametric test results and mean ranking at 1500 and 2000 dimensions. The usage of bold signifies that it is the best of these algorithms. Fig.17 displays the convergence curves of the two problem sizes, Fig. 18 shows the results of the analysis of variance of the two problem sizes, and Fig.19 shows the average fitness values of the 30 runs. And the spherical MST is listed in Fig.20.
### Table 4 Experimental result for the eight algorithms for 1500 and 2000 points.

<table>
<thead>
<tr>
<th>Points</th>
<th>Index</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACO</td>
</tr>
<tr>
<td>1500</td>
<td>Best</td>
<td>2221.7795</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>2260.3448</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2244.2728</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>p-Value</td>
<td>1.734E-06</td>
</tr>
<tr>
<td></td>
<td>h-Value</td>
<td>1</td>
</tr>
<tr>
<td>2000</td>
<td>Best</td>
<td>2999.8031</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>3033.1283</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>3015.5001</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>p-Value</td>
<td>1.734E-06</td>
</tr>
<tr>
<td></td>
<td>h-Value</td>
<td>1</td>
</tr>
</tbody>
</table>

![Fig. 17](image1.png) The convergence curves of algorithms for 1500 and 2000 points.  
![Fig. 18](image2.png) The ANOVA test for 1500 and 2000 points.
Table 4 shows the running results of MAOA and other algorithms in high dimensions of 1500 and 2000, respectively. The best, worst, average and standard values are compared respectively, and Wilcoxon rank and nonparametric test results are given. The table shows that MAOA outperforms other algorithms in terms of average, best, and worst values. For 1500 and 2000 dimensions, respectively, its standard deviation yields the best result. Table 4 displays the Wilcoxon rank sum test outcomes in high dimensions. The $p$ value is very small, so it is clear that MAOA is vastly better than other algorithms. Obviously, MAOA performs superior in terms of convergence accuracy and speed, and its exploration capabilities are enhanced and contribute to the balance between exploration and exploitation. Fig. 18 shows the results of variance analysis. AOA is more stable than MAOA. Fig. 19 displays the fitness values for 30 independent runs. In addition, Fig. 20 shows the MST of MAOA in different dimensions.

5. Communication network problem

The MST problem is an important problem in the design of communication networks. In
order to further verify the efficiency and performance of the proposed MAOA algorithm, different algorithms are used to compare and analyze the engineering design optimization problems. It is proved that MAOA has strong ability in finding acceptability and fast convergence speed. In this test, the population individual of all meta-heuristics is set to 30, the iteration number is 300, and they are run independently 10 times.

Optical cable laying is a common problem in real life, which can be solved by finding the lowest cost objective function. Usually the region involved is mapped to a two-dimensional plane to solve, but it gets bigger and bigger according to the complexity and scale of the problem. This experiment extends the problem to three-dimensional spheres and selects some popular cities on Earth, and the number of cities is divided into 100, 200 and 300 for experimental simulation. The latitude and longitude coordinate data of the cities in the experiment are all from MathWorks MATLAB Mapping toolbox. The experimental results are shown in Fig. 21, Fig. 22 and Fig. 23. It can be observed that the optimal solution and average value sought by MAOA are better than other algorithms, and MAOA provides a high-quality reference plan for decision makers.

Table 5 Experimental result for the eight algorithms for 100, 200 and 300 cities.

<table>
<thead>
<tr>
<th>Points</th>
<th>Index</th>
<th>Algorithms</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACO</td>
<td>DE</td>
<td>PSO</td>
<td>GA</td>
<td>GWO</td>
<td>SMA</td>
<td>AOA</td>
</tr>
<tr>
<td>100</td>
<td>Best</td>
<td>6.2799E+05</td>
<td>5.7411E+05</td>
<td>5.945E+05</td>
<td>6.7401E+05</td>
<td>6.0739E+05</td>
<td>4.7792E+05</td>
<td>6.2721E+05</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>6.6607E+05</td>
<td>6.2169E+05</td>
<td>6.5078E+05</td>
<td>7.1674E+05</td>
<td>6.5083E+05</td>
<td>5.4724E+05</td>
<td>6.7848E+05</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>6.4812E+05</td>
<td>6.0454E+05</td>
<td>6.2226E+05</td>
<td>6.9145E+05</td>
<td>6.3230E+05</td>
<td>5.1848E+05</td>
<td>6.4745E+05</td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>1.4426E+04</td>
<td>1.2046E+04</td>
<td>1.6089E+04</td>
<td>1.4042E+04</td>
<td>1.1161E+04</td>
<td>2.1144E+04</td>
<td>1.6173E+04</td>
</tr>
<tr>
<td>200</td>
<td>Best</td>
<td>1.2880E+06</td>
<td>1.2540E+06</td>
<td>1.2771E+06</td>
<td>1.4657E+06</td>
<td>1.3405E+06</td>
<td>1.1356E+06</td>
<td>1.1690E+06</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1.3407E+06</td>
<td>1.3132E+06</td>
<td>1.3663E+06</td>
<td>1.5149E+06</td>
<td>1.3738E+06</td>
<td>1.1844E+06</td>
<td>1.2016E+06</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.3204E+06</td>
<td>1.2812E+06</td>
<td>1.3289E+06</td>
<td>1.4871E+06</td>
<td>1.3608E+06</td>
<td>1.1613E+06</td>
<td>1.1881E+06</td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>1.6239E+04</td>
<td>1.6221E+04</td>
<td>2.7477E+04</td>
<td>1.6661E+04</td>
<td>1.0394E+04</td>
<td>1.6038E+04</td>
<td>1.0333E+04</td>
</tr>
<tr>
<td>300</td>
<td>Best</td>
<td>2.1074E+06</td>
<td>2.0609E+06</td>
<td>2.0364E+06</td>
<td>2.2402E+06</td>
<td>2.1379E+06</td>
<td>1.7954E+06</td>
<td>2.0511E+06</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>2.1738E+06</td>
<td>2.1260E+06</td>
<td>2.1729E+06</td>
<td>2.3243E+06</td>
<td>2.1982E+06</td>
<td>1.8492E+06</td>
<td>2.1090E+06</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.1438E+06</td>
<td>2.0933E+06</td>
<td>2.1065E+06</td>
<td>2.2820E+06</td>
<td>2.1691E+06</td>
<td>1.8206E+06</td>
<td>2.0758E+06</td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>2.1232E+04</td>
<td>2.2441E+04</td>
<td>4.3821E+04</td>
<td>2.7227E+04</td>
<td>2.1849E+04</td>
<td>1.7123E+04</td>
<td>2.0959E+04</td>
</tr>
</tbody>
</table>


Fig. 21 The best route around the globe for 100 cities.

Fig. 22 The best route around the globe for 200 cities.
6. Results analysis

It is clear from the simulation results data in Table 5 that MAOA exhibits strong search performance. By adding mathematical strategies, the best outcome in terms of the ideal solution and average value can be obtained with MAOA, which can swiftly converge to the global optimal solution. MAOA can achieve excellent performance from the following aspects.

First of all, MAOA is consistent with the two-stage optimization strategy of AOA in design principle, on the one hand, finding sub-domains through global exploration, and on the other hand, deep optimization through exploitation in each sub-domain. With the operation of the algorithm, it gradually changes from exploration behavior to exploitation behavior, which improves the balance between exploration and exploitation. Secondly, sine and cosine operators can add a new optimization method for AOA, which can avoid the single mode of population in the optimization process. The mathematical symmetric operator keeps jumping out of the local behavior to avoid population deterioration and make the population develop better. The median operator can effectively help MAOA dynamically adjust the stride length in the development stage and can be
used to improve the accuracy. This improves the probability of finding the global optimum and the convergence rate of MAOA.

Although the aforementioned experimental results demonstrate that MAOA performs better than most algorithms, it still has certain drawbacks and restrictions. When the problem scale is small, the speed and accuracy of MAOA are not much different from most algorithms, but worse than some improved algorithms. However, with the increase of problem scale, the accuracy of MAOA consistently achieves excellent results compared to other algorithms. For the global optical cable laying problem, it can be clearly seen that MAOA has a higher level of ability to find the best solution compared with the other seven algorithms. The main reason is that AOA is designed to solve these kinds of problems. Adding binary encoding enables MAOA to solve a wider variety of practical optimization problems.

7. Conclusions and future directions

AOA algorithm is a meta-heuristic optimization algorithm based on mathematical model. AOA has its simple and direct implementation, which is highly adaptable to the problems in real life. It does not require resizing the overall size and stopping many parameters beyond the criteria. Its built-in coefficients MOA and MOP also improve the balance of the AOA search process. It can be done without requiring a derivative. It can effectively solve many problems in continuous space. According to NFL theorem, AOA itself also has some shortcomings. The sine and cosine operators proposed in this paper can add new optimization operators to AOA, and increasing the diversity of optimization methods can avoid the fixed mode of population in the optimization process. Mathematical symmetry operator is a case of applying mathematical geometry method to arithmetic optimization algorithm. It makes the individual of the population constantly jump out of the local behavior to avoid the population into the local optimal, so that the population can develop better. Median operator adopts the idea of median between any two points in mathematics, which can effectively help MAOA dynamically adjust stride length in the development stage and can be used to improve accuracy. In this paper, the spherical MST problem is solved on 14 different 3D spheres with BBMA, IRAEFA, GA, PSO, GWO, SMA, DE, ACO and AMO. The performance of the proposed MAOA is verified by comprehensive experiments. Therefore, MAOA algorithm is a more suitable algorithm for solving spherical MST problem. According to
NFL theorem, it is impossible for an algorithm to show superior performance for all optimization problems. Like all algorithms, MAOA has some drawbacks. MAOA currently has two major drawbacks: long running time and improved stability. Of course, MAOA has other drawbacks. These problems are also within the scope of practical application to some extent. Next, MAOA will be further studied to solve the spherical MST problem and solve the above two shortcomings. Finally, the improved algorithm will be applied to solve the spherical MST problem, such as eliminating the noise of the femur surface and directional position estimator. In addition, its practical difficulties in other engineering fields will also be considered for application, such as last-kilometer distribution, logistics center sit, three-dimensional path planning, thermoelectric dispatching and other problems.

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

Author Contributions X. M: carried out the MAOA algorithm studies; Q. L: Supervision, writing-review and editing; Y.W: participated in the drafted manuscript; Y.Z carried out the review and editing. H. H: arried out the review; All authors read and approved the final manuscript.

Funding This research is funded by the National Natural Science Foundation of China, Grant number U21A20464, 62066005.

Availability of data and materials 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.

References


61. Jian Bi, Yongquan Zhou, Zhonghua Tang, et al. Artificial electric field algorithm with inertia


76. Sarah Steiner, Tomasz Radzik. Computing all efficient solutions of the biobjective minimum


