Research on the geometric model of Bentauron based on simulated annealing algorithm

Mengyao Zhang [#], Rongpeng Yan [*, #], Shuqi Wang [#]

College of sports Engineering, Beijing Sport University, Beijing, China, 100091

* Corresponding Author Email: 18840697359@163.com

[#]These authors contributed equally.

Abstract. Based on the background of the traditional folk culture activity "Bench Dragon", this paper presents mathematical modeling and optimization analysis for the marching and turning of the dragon dance team on a specific path. Firstly, this paper determines the velocity and position distribution of each bench under the condition of constant faucet speed by helix equation. Aiming at the problem of the faucet entering the turning space smoothly, this paper determines the minimum pitch which satisfies the space limitation by mathematical derivation and iterative algorithm. In order to further optimize the turning path, the optimal turning path is obtained by combining the simulated annealing algorithm and the random seed method, and the tangency between the turning curve and the spiral path is ensured. Finally, through the method of dichotomy and particle swarm optimization, the paper finds the optimal solution that the maximum traveling speed of the leader does not exceed 2 m/s, which ensures the speed coordination and safety of the whole team. The optimization scheme proposed in this paper not only ensures the smooth entry and exit of the dragon dance team in the complex path, but also improves the appreciation and practicability of the performance.

Keywords: Equidistant spiral equation, simulated annealing algorithm, particle swarm optimization algorithm, random seed, and iterative algorithm.

1. Introduction

Bench dragon ", also known as "coiling dragon", is a traditional folk culture activity in Zhejiang and Fujian regions. One of the traditional folk activities originated in Heluo area, it is widely present in the annual grand activities of various provinces and cities in southern China. The dragon dance performance uses teams to control the dragon body, which is connected by several benches, to circle and turn along a specific path, showing complex dance positions and fancy movements. With the expansion of the scale of urban festival activities, the spatial planning and path design of dragon dance team performance have gradually become the key factors affecting the overall performance effect. Proper path planning and turn design can not only improve the fluency of the performance, but also ensure that the team members maintain the right speed and avoid collisions or mistakes between the team members. Therefore, in a limited space, how to optimize the path and speed of the leader has become the core issue to improve the quality and safety of the performance.

In the actual performance, the dragon dance team usually follows a spiral path and completes a S-turn in a limited space. At the same time, the control hand of each bench in the team must maintain the appropriate speed to maintain the overall coordination and safety of the team.

In order to solve these problems, this paper proposes a series of optimization schemes through mathematical modeling and numerical simulation, focusing on the speed and position distribution of the dragon dance team in the process of marching and turning on a specific path. We first modeled the spiral trajectory of the faucet, and discussed the velocity and position distribution law of each bench under the condition of constant faucet speed. Although this part provides basic data support for the subsequent complex problems, the following research will pay more attention to the optimization of path and speed.

First of all, this paper is devoted to determining the minimum pitch for the faucet to enter the turning space smoothly. In this problem, we first calculate the intersection point between the helix and the turning space, and derive the minimum pitch by mathematical formula. Through iterative

calculation and collision detection model, the minimum pitch that can meet the conditions is gradually adjusted and determined, so that the faucet can enter the turning space smoothly and avoid collision with other benches. This optimization not only takes into account the spatial limitations of the path, but also ensures a smooth turn-around process.

Then, in order to further optimize the turning path of the faucet, the trajectory model of the turning curve is established, and the target optimization equation is constructed. Combining the traditional optimization method and simulated annealing algorithm, this paper obtains several possible optimal solutions through iterative calculation of a large number of time steps, and judges the stability of the results by setting random seeds. Finally, the shortest turn-around path is obtained. In this process, the tangential equation is used to ensure the tangency of the turning curve and the spiral path, so as to ensure the smooth transition of the faucet during the turning process, and improve the accuracy and efficiency of path planning.

Finally, this paper studies the relationship between the different positions of the spiral curve and the speed of the faucet, and finds that the speed of each bench decreases with the increase of the distance from the position of the faucet. Therefore, an innovative method combining dichotomy and particle swarm optimization algorithm is proposed in this paper. The maximum speed of the spigot meeting the conditions is approximated gradually by dichotomy, and then the fitness value of each particle is iteratively calculated by combining the particle swarm optimization algorithm. Finally, the optimal solution satisfying the speed of each segment is found no more than 2 m/s. This optimization scheme not only ensures the reasonable speed distribution of each bench, avoids team mistakes caused by excessive speed, but also effectively improves the overall performance.

The main problems of the existing research on the motion trajectory of Benzosaurus are as follows: lack of systematic technical analysis, the motion trajectory modeling is too simple, and its complexity is not fully considered; the research on collision detection and path optimization is not deep enough to solve the practical problems in the performance. In this paper, a simulated annealing algorithm is used to propose a solution with more technical depth[1], including accurate motion modeling through isometric spiral equations[2], real-time collision detection mechanisms, and velocity optimization strategies[3] combining particle swarm optimization and dichotomy[4]. These methods effectively make up for the shortcomings of existing research, and provide important technical support for the optimization of modern Bentauron movement and performance improvement.

2. Methods

2.1. Simulated annealing algorithm

Simulated annealing algorithm (SA) is a probability-based algorithm derived from the principle of solid annealing. Starting from a certain high initial temperature, the simulated annealing algorithm randomly finds the global optimal solution of the objective function in the solution space with the continuous decline of temperature parameters and combined with the probabilistic jump characteristics, that is, the local optimal solution can jump out probabilistically and eventually tend to the global optimal [5]. The simulated annealing algorithm consists of two parts, the Metropolis algorithm and the annealing process, which correspond to the inner cycle and the outer cycle respectively. The outer cycle is the annealing process, the solid reaches a higher temperature (initial temperature T (0)), and then the temperature decreases according to a certain proportion according to the cooling coefficient alpha. When the termination temperature Tf is reached, the cooling ends, that is, the annealing process ends. The Metropolis algorithm is an inner loop, that is, at each temperature, it iterates L times, finds the minimum value of energy at the temperature (that is, the optimal solution), and accepts the new state with probability, rather than using a fully determined rule, called the Metropolis criterion, which is less computational. As shown in Figure1.

2.2. Dichotomy

Suppose we want to find the root of the function f(x) in the interval [a, b], that is, solve for f(x) =0. The function f(x) is continuous at [a, b] and satisfies f (a) f (b) <0, that is, the sign of the function values at a and b is different, indicating that according to the mean value theorem [6], the function has at least one zero in the interval [a, b]. Dichotomies work by constantly dividing the interval in half and gradually narrowing down the range where the zero point may be[7].

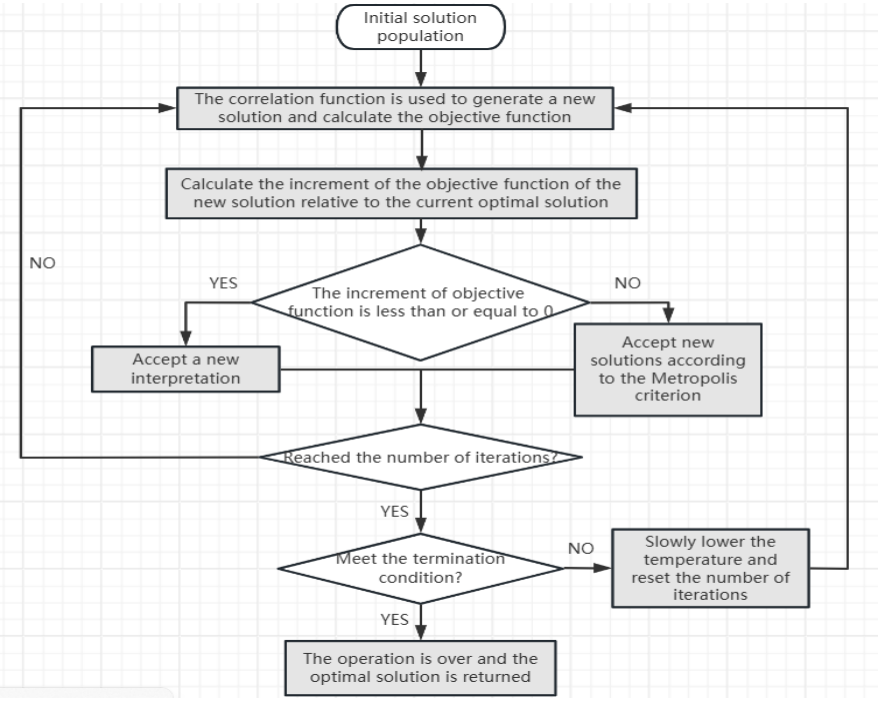


Figure 1. Flow chart of simulated annealing algorithm.

2.3. Particle swarm algorithm and two particle swarm algorithm

2.3.1. Particle swarm optimization (PSO) principle

First, PSO is an optimization algorithm based on swarm intelligence, which simulates the collective behavior of birds or fish to solve complex optimization problems. The basic idea is that each solution is treated as a particle that moves through the search space, and the position of each particle represents a possible solution. The particle is constantly updated by adjusting its own speed and position, with the goal of finding the global optimal solution in the search space [8].

A. Speed update formula:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot \left(p_{best,i} - x_i(t)\right) + c_2 \cdot r_2 \cdot \left(g_{best} - x_i(t)\right) \quad (1)$$

B. Location update formula:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Where, $v_i(t)$is the velocity of particle i at time t,$x_i(t)$is the position of particle i at time t, $p_{best,i}$ is the best place in the history of particle i, $g_{best}$is the best place in the world, w, c_1, c_2 are inertia weight, individual learning factor and group learning factor respectively, $r_1, r_2$is a random number between [0,1].

2.3.2. Two particle swarm optimization

The core idea of two-particle swarm algorithm is to introduce two different particle swarms (population 1 and population 2) on the basis of PSO, and each particle swarm has different parameter Settings or search strategies to cooperate with each other to improve optimization performance[9]. The two-particle swarm algorithm divides the search process into two groups:

Group 1 (Exploration Group): It is mainly responsible for global search, exploring the unknown search space, and preventing the algorithm from falling into the local optimal. The inertia weight of this group is larger to maintain a wider range of particle search capabilities and enhance exploration.

Group 2 (Exploitation Group): mainly responsible for local search, using known good solutions to fine-tune, improve the convergence of the algorithm. The inertia weight of this group is smaller, the particle motion amplitude is smaller, and the emphasis is on precise optimization.

There is information exchange between the two groups, and groups 1 and 2 influence and learn from each other through their own global optimal solutions. Group 1 May obtain a more accurate solution through the local search of group 2, whereas group 2 May benefit from the global exploration of group 1 and jump out of the local optimal. The inertia weight and learning factor can be dynamically adjusted during the running of the algorithm. For example, as the number of iterations increases, the inertia weight may gradually decrease to enhance convergence and avoid the endless search of particles. Dynamic tuning can help particles explore more extensively in the early stages of the algorithm, and in the late stages focus on the precise search for locally optimal solutions. The velocity and position update formulas of the two-particle swarm algorithm are still similar to the traditional PSO, but it applies different parameters to population 1 and population 2 respectively. For example: the speed update tends to global exploration, and the inertia weight is larger; for group 2, the speed update is biased towards local development, and the inertia weight is small.

2.4. The first three questions about mathematical derivation process

2.4.1. Spiral equation

Firstly, the spiral equation is established. The parametric equation of the spiral is used to calculate the position of the node on the spiral. The front handle of the faucet enters clockwise along the isometric spiral, and the formula is established as follows:

$$\begin{cases} \mathbf{x} = r\cos(\theta) \\ y = r\sin(\theta) \\ r = a + b\theta \\ b = \frac{0.55}{2\pi} \end{cases} \tag{3}$$

Here, r is the polar diameter of the spiral (the distance from the origin to the center of the spiral), increases linearly with Angle θ. a is the radius of the spiral where the faucet is located at the initial time (t=0). In this case, the initial faucet is located at the point A of the 16th circle, so the starting half-diameter a=16*0.55, that is, the starting radius is 8.8 meters. b is the converted increment of the pitch, indicating the increase of the radius of the spiral each turn. 0.55 is the pitch 55cm given in the title [10].

2.4.2. Angular change

Then calculate the Angle change, the faucet moves along the spiral at a constant speed v, the unit is m/s, each time step Δt, the unit is s, the corresponding Angle change Δ0. The following formula is obtained:

$$\Delta\theta = (v \cdot \Delta t)/r \tag{4}$$

$\Delta\theta$ Represents the polar Angle, which varies with time and satisfies the above formula.

2.4.3. Location update

Then the position of the dragon body and tail is updated. For each node of the dragon body and tail, when they enter the spiral clockwise from point A, make sure that the front handle position of each node is always on the spiral. We use parametric equations to calculate the new position of each node. Since the distance between each bench is fixed, the position of subsequent benches is gradually calculated by recursive calculation method [11]. The position of the bench in each section can be

derived according to the position and direction of the bench in the previous section, and the formula is as follows:

$$y_{new"} = y_{old"} - v \cdot \Delta t \tag{5}$$

If the node is already in a spiral, update its position according to the spiral equation:

$$\begin{cases} r_i = a + b\left(\theta - \frac{L_i}{r}\right) \\ \theta_i = \theta - \frac{L_i}{r} \\ x_i = r_i \cos(\theta_i) \\ y_i = r_i \sin(\theta_i) \end{cases} \tag{6}$$

L_i is the cumulative length of the current node from the head, including the head length and the length of all the dragons in front of it.

2.4.4. Velocity calculation

Finally, the speed of each bench is calculated, and after each time step, the speed of each node is updated as the distance from the next node divided by the time step:

$$v_i = \frac{(x_{i+1} - x_i, y_{i+1} - y_i)}{\Delta t} \tag{7}$$

Where: x_i, y_i are the current coordinates of node i. x_{i+1}, y_{i+1} are the current coordinates of the next node. Finally at the resultant rate:

$$v_{\text{total}} = \sqrt{v_x^2 + v_y^2} \tag{8}$$

2.4.5. Collision detection condition

The length of each bench is fixed (341cm for the head and 220cm for the body and tail), so we need to detect the distance between the adjacent benches and iterate to prevent them from overlapping or colliding with each other. Distance can be judged by the following formula:

$$d = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \tag{9}$$

Detection criteria: For adjacent benches in section m and section n, the distance between their front handles is detected to see if it is less than the length of the bench. If the distance d is less than the length of the bench, collision will occur.

Detection implementation: Simulate the movement of the dragon dance team, calculate the position information of all benches per second, and detect the distance between all benches in real time. If at some point the distance between the two benches is found to be less than the minimum distance allowed, it is considered that the dragon dance team cannot continue the disc, and the current moment and its position and speed are recorded as the termination moment of the disc.

3. Results

3.1. The establishment of simulation model

The running track prediction model of bench dragon is implemented in Python software.

3.2. Analysis of experimental results

3.2.1. Determination of turning space

$$x^2 + y^2 = R^2 \tag{10}$$

The turning space is a circle with a diameter of 9m centered at the origin and R=4.5m.

3.2.2. Shortest path problem

Calculate path length. According to the meaning, the turn-around path is an S-shaped curve composed of tangent arcs at both ends, and is located in the turn-around space (a circle with a diameter of 9 meters). We use two arcs to represent the turn-around process of the dragon dance team respectively. Since the radius of the former arc is twice that of the latter, the total length ds of the S-shaped path can be calculated according to the following formula:

$$d_s = R\mathbf{1} \times |\theta\mathbf{1}| + R\mathbf{2} \times |\theta\mathbf{2}| \tag{11}$$

$R\mathbf{1}$ And $R\mathbf{2}$ are the radii of the first large arc and the second small arc, and $\theta\mathbf{1}$ and $\theta\mathbf{2}$ are the angles of the two arcs, respectively.

Calculation of S-shaped curve the path coordinates of the two arcs can be calculated by determining the S-shaped curve for the subsequent tangency check [12]. Calculated according to the following formula: The coordinates of the first great arc:

$$\begin{cases} x_{\text{arcl}} = R1 \times \cos(\pi - \theta 1) \\ y_{\text{arcl}} = R1 \times \sin(\pi - \theta 1) \end{cases} \tag{12}$$

Second circle coordinates:

$$\begin{cases} x_{\text{arc2}} = R2 \times \cos(\theta 2) + x_{\text{tangent}} - R2 \times \cos(0) \\ y_{\text{arc2}} = R2 \times \sin(\theta 2) + y_{\text{tangent}} - R2 \times \sin(0) \end{cases} \tag{13}$$

Where, x_{tangent} and y_{tangent} is the coordinate of the end point of the first arc, as the starting point of the second arc.

In order to ensure that the S-path is tangential and smoothly connected to the disc in and disc out spiral, it is necessary to check the tangency between them. The specific steps are as follows:

• Calculate the tangent direction of the starting and ending points of the arc:

$$\begin{cases} arc_tangent_start = arctan(\frac{y_{\text{arc}}[1]-y_{\text{arc}}[0]}{x_{\text{arc}}[1]-x_{\text{arc}}[0]}) \\ arc_tangent_end = arctan(\frac{y_{\text{arc}}[-1]-y_{\text{arc}}[-2]}{x_{\text{arc}}[-1]-x_{\text{arc}}[-2]}) \end{cases} \tag{14}$$

• Calculate the tangent direction of the starting and ending points of the spiral:

$$\begin{cases} spiral_tangent_start = arctan\left(\frac{y_{\text{spiral}}[1]-y_{\text{spiral}}[0]}{x_{\text{spiral}}[1]-x_{\text{spiral}}[0]}\right) \\ spiral_tangent_start = arctan\left(\frac{y_{\text{spiral}}[1]-y_{\text{spiral}}[0]}{x_{\text{spiral}}[1]-x_{\text{spiral}}[0]}\right) \end{cases} \tag{15}$$

To ensure that the connecting points of the two arcs are tangential and opposite in direction, calculate the tangential directions of the end point of the big arc and the start point of the small arc. Use the following formula to determine:

$$\begin{cases} tangent-1-end = arctan\left(\frac{y_{\text{spiral}}[-1]-y_{\text{spiral}}[-2]}{x_{\text{spiral}}[-1]-x_{\text{spiral}}[-2]}\right) \\ tangent-2-start = arctan\left(\frac{y_{\text{arc}}[1]-y_{\text{arc}}[0]}{x_{\text{arc}}[1]-x_{\text{arc}}[0]}\right) \end{cases} \tag{16}$$

3.2.3. S-curve results show

As can be seen from the Figure 2, the S-curve satisfies all tangential conditions and no collision occurs. According to the figure, the results before and after optimization are the same. However, since the results have been stabilized by random seeds, the correctness and accuracy of the results can be fully determined.

Through the path optimization of the S-shaped curve, the research successfully shortened the path length to 5.890 meters, and ensured that the maximum traveling speed of the faucet did not exceed 2 m/s, which fully demonstrated the optimization performance of the algorithm and the effectiveness

and safety under complex conditions. The optimization strategy of simulated annealing, dichotomy and particle swarm optimization not only speeds up the convergence of the optimal solution, but also ensures the accuracy of path planning under multi-variable constraints. This collaborative optimization mechanism greatly improves the global search ability of the problem, avoids the dilemma of local optimal, and further improves the accuracy and operability of the understanding through fine adjustment of local search. In addition, the algorithm can effectively dynamically adjust the search scope and parameters, making the optimization process more flexible to adapt to the needs of different stages, especially in the early stage of extensive exploration, the later focus on local fine optimization. Overall, the optimization mechanism of the algorithm significantly improves the efficiency of problem solving, ensures the maneuverability of the path, reasonable control of the speed and the stability of the whole system. As shown in Table1 and Figure 2.

Table 1. The value of each parameter after optimization.

Comparison	Numerical value
The optimized R1	3.000 m
Optimized R2	1.500 m
Shortest path length	5.890 m

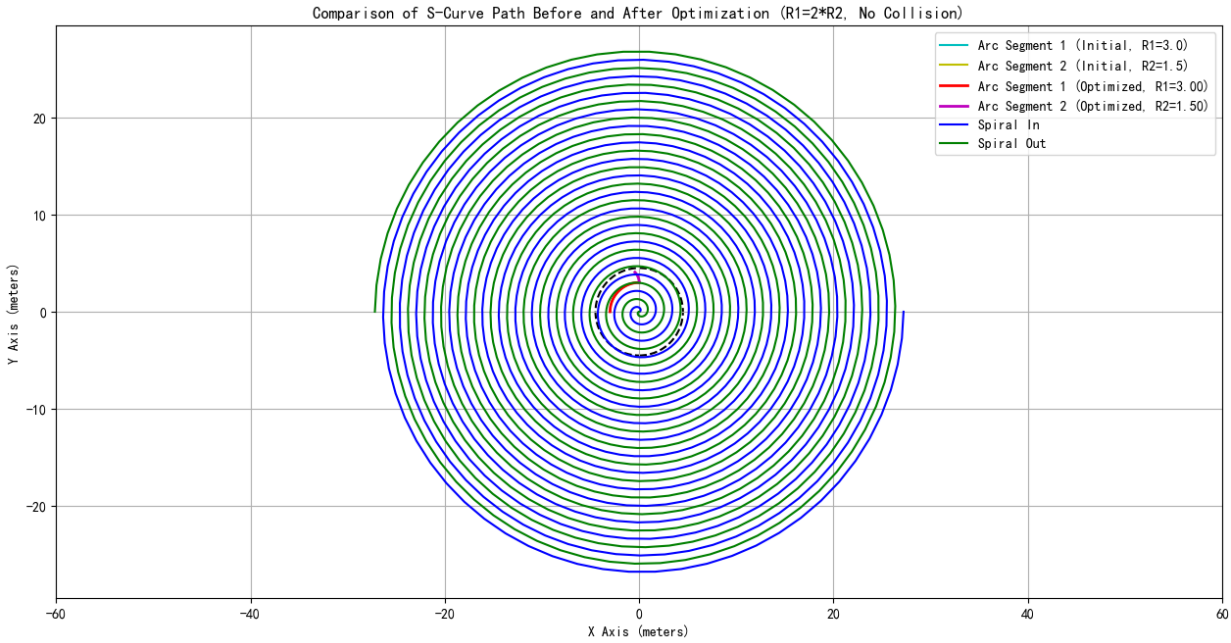


Figure 2. Reverse path comparison diagram.

3.2.4. Maximum velocity problem

It can be obtained from the first four questions: the bench speed after the head is less than the previous bench speed, that is, the bench speed decreases with the position. Therefore, it can be preliminarily inferred that if the speed of each handle of the dragon dance team does not exceed 2m/s, when the maximum traveling speed of the head does not exceed 2m/s, the conditions are met.

Dichotomy to solve the maximum speed of the faucet.

Initialization interval: The initial range of the faucet speed is set to [vmax, vmin], which can be set to [0, 2]m/s according to the above assumptions.

Calculation iteration: Calculate the intermediate value vmid and bring the speed vi of each handle into the calculation.

Check whether the vi values do not exceed 2 Mbit /s. If satisfied, it means that the faucet speed can continue to increase. On the contrary, the faucet speed should be continuously reduced.

Repeat steps 1-3 until the difference of v_{max}-$v_{\min}$ is less than the given tolerance to output the result, at which time vmid is the maximum traveling speed of the faucet.

3.2.5. Optimization of two particle swarm optimization algorithms

The Figure 3 describes the working principle of the two-particle swarm optimization algorithm, that is, the process of determining the range of the search feasible solution by dichotomy first, and then obtaining the optimal solution by particle swarm optimization algorithm in detail.

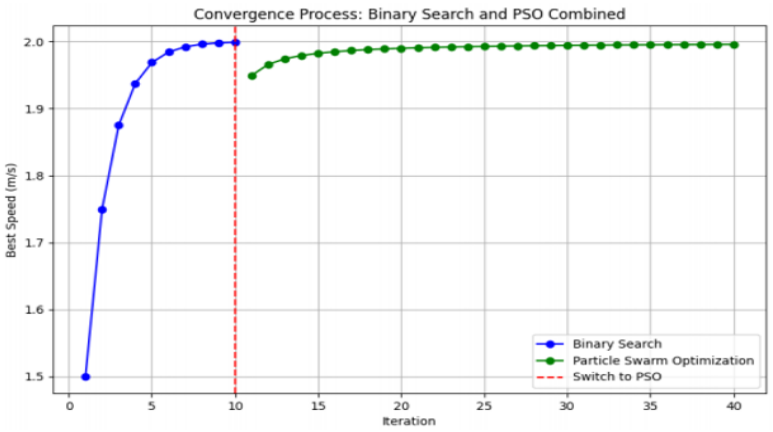


Figure 3. Optimization algorithm graph.

From the visual analysis of the above Figure 3, it can be concluded that: combining the dichotomy and particle swarm optimization algorithm, the two-particle swarm optimization algorithm shows better search performance. Group 1 (global search group) uses large inertia weights to explore in a wide solution space to avoid falling into local optimality. Group 2 (local development group) uses a small inertia weight and fine-tuning mechanism to carry out fine optimization after finding a good solution to improve the accuracy of the final solution. The two groups operate cooperatively and learn from each other. Group 1 provides group 2 with the opportunity to jump out of the local optimal through global search, and group 2 optimizes the result, which in turn affects the global search of group 1. This cooperative mechanism and flexible weight adjustment make PSO more suitable for complex, multidimensional and nonlinear problems. Compared with the binary method, the two-particle swarm optimization algorithm does not rely on the initial setting, can find the optimal solution that meets the requirements of the problem faster and more accurately, and can find the maximum traveling speed of the faucet that meets the conditions of the problem more quickly under the same conditions than the simple use of the binary method. As shown in Figure 4.

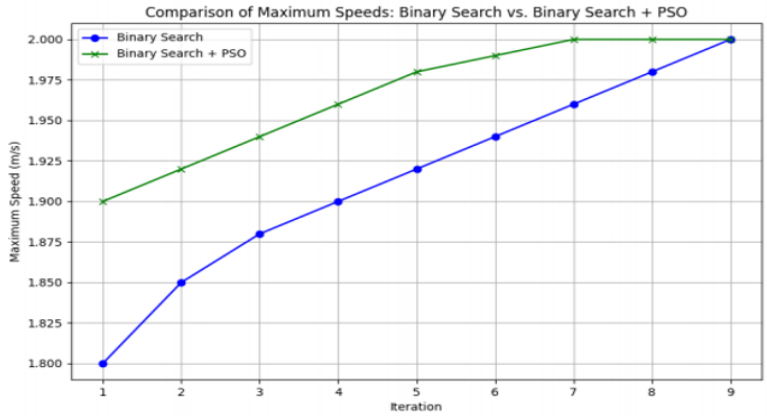


Figure 4. Comparison diagram of particle swarm and dichotomy

4. Conclusions

The advantage of the model is that it directly deduces the mathematical theory of the problem model and directly solves the problem results, which makes the problem-solving ideas clearer and clearer. The iterative algorithm is used to approach the optimal solution continuously, and the minimum pitch is obtained directly without complicated mathematical derivation, but the reasonable solution is found

through many attempts. Because there are many variables in the path optimization problem, simulated annealing algorithm is used to avoid the sub-optimal problem caused by unreasonable initial parameter setting. The combination of dichotomy and particle swarm optimization can rapidly narrow the search scope. Through dichotomy, the search scope can be quickly narrowed in a large search interval and the initial search space can be reduced. This allows particle swarm optimization algorithms to concentrate on fine optimizations in a small range, reducing computation time. The model is flexible. By defining different objective functions, particle swarm optimization can flexibly deal with multi-variable constraints and find the optimal solution satisfying the conditions.

At the same time, the shortcomings of the model are: lack of corresponding dragon dance performance experience, there is one-sidedness in the setting of conditions and constraints. The actual path of the dragon dance team may deviate from the set helix. In the collision detection model, the collision condition is idealized by the hypothesis, which is different from the reality. The parameter setting of the simulated annealing algorithm can only rely on common sense, which leads to certain errors in the final result.

It can also be applied to other performance scenes: the model is not only suitable for path planning and speed control problems in dragon dance team performance, but also can be extended to other similar performance scenes, such as float parade. In these scenarios, coordination between performers, path design and speed control are equally critical, and the methods in the model can provide theoretical support for path optimization.

Application to automated logistics systems: Unmanned or automated equipment in logistics distribution systems in complex warehouses or high-volume transportation environments, it is also necessary to optimize paths and control speed to improve efficiency and avoid collisions and congestion. The collision detection model and speed control strategy in the model can provide effective solutions for this kind of automated driving system.

References

[1] Morshedin H M, Sadjadi J S, Amiri B, et al. Competitive pricing and seed node selection in a two-echelon supply chain [J]. Operations Research Perspectives, 2024, 13: 100314.

[2] A. W A, S. A J. Applicability of GLI 2012 spirometry equation among preschool aged Jordanian [J]. Respiratory Medicine, 2021, 182: 106397.

[3] Putra A L, Köstler M, Grundwürmer M, et al. State estimation of a biogas plant based on spectral analysis using a combination of machine learning and metaheuristic algorithms [J]. Applied Energy, 2024, 377: 124447.

[4] Copple R. The Fallacy of Unambiguous Warning [J]. Parameters, 2024, 54(3): 50.

[5] Mallaiah M, Rama Rao K, Venkateswarlu C. A simulated annealing optimization algorithm based nonlinear model predictive control strategy with application [J]. Evolving Systems, 2021, 12(1): 225-231.

[6] Chernov A, Lê T. Analytic and Gevrey class regularity for parametric elliptic eigenvalue problems and applications [J]. SIAM Journal on Numerical Analysis, 2024, 62(4): 1874-1900.

[7] Lin H, Yang X. Dichotomy algorithm for solving weighted min-max programming problem with addition-min fuzzy relation inequalities constraint [J]. Computers & Industrial Engineering, 2020, 146: 106537.

[8] Han F, Chen W T, Ling Q H, et al. Multi-objective particle swarm optimization with adaptive strategies for feature selection [J]. Swarm and Evolutionary Computation, 2021, 62: 100847.

[9] Gad a G. Particle swarm optimization algorithm and its applications: a systematic review [J]. Archives of computational methods in engineering, 2022, 29(5): 2531-2561.

[10] Al-Azzawi A, Stadler P, Kong H, et al. Length Modelling of Spiral Superficial Soft Strain Sensors Using Geodesics and Covering Spaces[J]. Robotics, 2023, 12(6): 164.

[11] Zhao Liqun. A preliminary discussion on recursive calculation method and recursive algorithm from the perspective of computational thinking [J]. Digital Technology and Application, 2002, 40(07): 140-143.

[12] Zhao J, Liu X, Wang S, et al. Review of Design and Control Optimization of Axial Flux PMSM in Renewable-energy Applications[J]. Chinese Journal of Mechanical Engineering, 2023, 36(1): 45.