Research on Dynamic Path Planning and Real-Time Collision Detection for Bench Dragon Based on Multi-Algorithm Collaborative Optimization

Zexi Ding

School of Mathematics & Computing Science, Guilin University of Electronic Technology, Guilin, China, 541004

13921523816@163.com

Abstract. The dynamic and complex nature of traditional dragon dance performances, especially the multi-segmented "Bench Dragon" presents considerable challenges in real-time collision detection and path optimization. This study introduces an integrated optimization framework that combines an enhanced golden section search algorithm, dynamic collision detection based on the Separation Axis Theorem (SAT), and a genetic algorithm to tackle these challenges. The enhanced golden section algorithm effectively reduces computational complexity by efficiently narrowing the search space, while SAT ensures collision-free motion through real-time projection analysis. Additionally, genetic algorithms optimize the turnaround path by minimizing the total length under geometric and continuity constraints. Experimental results demonstrate the framework's effectiveness: an optimal pitch of 0.4529 m facilitates smooth coiling, with the total turnaround path length reduced to 13.6206 m without any collisions. The algorithm maintains high accuracy while ensuring computational efficiency, as validated through simulations of the dragon's movement across critical time intervals (0–300 s). This research offers a robust solution for coordinating large-scale rigid-chain movements, with potential applications in swarm robotics and automated logistics. However, limitations include idealized assumptions of rigidity and 2D motion, underscoring the need for future research into real-time dynamic modeling and multi-physics constraints.

Keywords: Bench Dragon, Collision Detection, Simulated Annealing Algorithm, Genetic Algorithm.

1. Introduction

The intricacy and energy of dragon dance performances, a traditional cultural activity, have drawn more attention recently. To prevent collisions and guarantee a seamless performance, the dragon body's multi-segmented movement necessitates exact coordination. However, because of the system's dynamic nature and the requirement for real-time collision detection, planning and optimizing such complicated trajectories continues to be difficult. Few studies have examined the combined optimization of collision detection and path planning simultaneously, but prior research has concentrated on collision detection algorithms [1] and path optimization strategies [2]. By putting forward an enhanced collision detection and path optimization technique, this work seeks to solve this issue.

Current collision detection methodologies, exemplified by the Separated Axis Theorem (SAT) [3], have achieved widespread implementation in computer graphics and robotic systems. While effective in static environments, these techniques suffer from significant computational complexity in dynamic scenarios, particularly when real-time detection constraints are imposed. Parallel developments in path optimization, such as genetic algorithm-based approaches [4], demonstrate potential for solving complex optimization problems. However, such evolutionary algorithms exhibit prohibitive computational costs when scaled to large-dimensional problems, with conventional optimization frameworks frequently neglecting critical dynamic system constraints, thereby yielding suboptimal solutions.

The computational limitations are evident in recent implementations. Rodrigues Maia et al. [5] demonstrated SAT-based collision detection achieving $\mathbf{O(n^2)}$ time complexity in dynamic environments, severely restricting real-time applicability. Similarly, Liu X's genetic algorithm implementation [6] for path optimization, while theoretically sound, encounters practical limitations due to $\mathbf{O(n \log n)}$ computational requirements that degrade performance in time-sensitive applications. These case studies collectively highlight the persistent challenge of balancing algorithmic precision with computational efficiency in dynamic systems.

The principal contributions of this study are systematically summarized as follows:

1) Enhanced golden section search algorithm in path optimization: This work introduces an improved golden section search algorithm that enhances the efficiency of path optimization by dynamically adjusting search space reduction mechanisms. Compared to conventional techniques, the proposed method achieves more effective narrowing of the solution space through optimized contraction ratios and iterative strategies. This advancement addresses computational challenges in high-dimensional trajectory planning for complex dragon dance movements.

2) SAT: A SAT-based collision detection framework is developed to ensure movement safety and precision for multi-segment dragon structures. By modeling geometric characteristics of articulated dragon components, this method enables accurate collision prediction for non-rigid body motions while fulfilling real-time processing requirements inherent in dynamic performances.

3) Integrated optimization framework for dynamic performances: A unified computational architecture is established to synergistically combine path optimization and collision detection modules. The framework implements bidirectional constraint updating during path iteration processes, effectively accommodating real-time environmental variations in dragon dance choreography.

Through comprehensive simulations and experimental validation, the integrated algorithm demonstrates significant computational efficiency improvements while maintaining motion accuracy. This research provides a scalable technical foundation for digitizing traditional cultural performances.

2. Methodology

2.1. Basic definition of parameters

The “Bench Dragon” is a traditional folk cultural activity practiced in Zhejiang Province and other regions of China. This vibrant dance involves linking multiple benches, head to tail, to create a sinuous dragon shape. The movement and speed of the dragon's head, body, and tail play a crucial role in determining the overall visual impact as the dance team weaves along a spiral path. It is essential to calculate the movement of the dragon dance team along this specified helix, establish the optimal speed and trajectory for coiling, and address the challenges of turning under specific conditions. This requires precise calculations of position and speed to prevent collisions between benches during the performance and to optimize the turning path.

A bench dragon is comprised of 223 sections of benches. The first section represents the dragon's head, followed by 221 sections that form the dragon's body, and the final section serves as the dragon's tail. The length of the dragon's head is 341 cm, while both the body and tail sections measure 220 cm each. All benches have a width of 30 cm. Each section features two holes, each with a diameter of 5.5 cm, positioned such that the center of the holes is 27.5 cm away from the head of the nearest bench.

This paper outlines fixed parameters aimed at facilitating the calculation of formulas and the graphical representation of the findings. The dragon dance team executes a clockwise movement along an isometric screw line, with the dragon's head maintaining a speed of 1 m/s as it coils inward. The center of each handle is positioned along the screw line, which is defined as an Archimedean screw line by default. The turnaround area is a circular space with a diameter of 9 meters, centered around the thread itself, which is symmetrical concerning the thread. To achieve a successful turnaround, the dragon dance team establishes this designated space, consisting of two arcs that are tangent to one another, forming an S-shaped curve. The radius of the first arc is twice that of the second arc, and both arcs are tangent to the coil-in and coil-out solenoids. These arcs can be adjusted to maintain tangency, thereby reducing the overall length of the turnaround curve.

2.2. Establishment of basic mathematical model

First, for the given path of the "disk dragon" in the question, we establish the Archimedean solenoidal equation:

$$r = a + b \cdot \theta \tag{1}$$

Where a is the initial radius; b is the proportionality constant.

$$b = \frac{p}{2\pi} = \frac{0.55}{2\pi} \approx 0.864 \tag{2}$$

θ is the polar angle. A coordinate transformation of the solenoidal equation yields:

$$\begin{cases} x = 0.864\theta cos\theta \\ y = 0.864\theta sin\theta \end{cases} \tag{3}$$

Next, the length of the Archimedean spiral is determined through integration, with the angle θ corresponding to each tap and its associated coordinates calculated based on the gradual reduction in the length of the spiral. To compute the length $L(\theta)$ of the helix from the starting point to a specific angle θ, we utilize a calculus formula. The expression for the infinitesimal length element dL of the spiral is [7]:

$$dL = \sqrt{\left(\frac{dr}{d\theta}\right)^2 + r(\theta)^2} d\theta \tag{4}$$

Due to

$$r(\theta) = a + b \cdot \theta \tag{5}$$

Then:

$$\frac{dr}{d\theta} = b \tag{6}$$

Thus, the differential expression can be written as:

$$dL = \sqrt{b^2 + r(\theta)^2} d\theta \tag{7}$$

The total length $L(\theta)$ is integrated over the angle θ:

$$L(\theta) = \int_0^\theta \sqrt{b^2 + r(\theta')^2} d\theta' \tag{8}$$

Because the bench has rigidity, the distance between its two apertures is constant, so we use iteration to make a circle to find the body and tail of the dragon, the equation of the circle is coupled with the equation of the spiral line, and the point of the spiral line in the plane can be transformed from polar coordinates to Cartesian coordinates, in the iterative process, we keep looking for the intersection point of the spiral line with multiple circles. Assuming that the current center of the circle $i(x_c, y_c)$ and the radius of the circle is $r_{rcircler}$, the distance from any point on the helix to the center of the circle can be expressed as:

$$d(\theta) = \sqrt{(x(\theta) - x_c)^2 + (y(\theta) - y_c)^2} \tag{9}$$

We want to find points on the spiral that satisfy the conditions:

$$d(\theta) - r_{\text{circle}} = 0 \tag{10}$$

The center of the circle is updated to the solved, and the position coordinates are solved by iterating sequentially. Where the radius update only exists for the case of the dragon head and the first section of the dragon body.

We saved the position coordinates of each part at different times and used the center difference method to calculate the velocities (solving for the x- and y-directions, respectively):

$$\begin{cases} v_x(t_i) \approx \frac{x(t_{i+1})-x(t_{i-1})}{2\Delta t} \\ v_y(t_i) \approx \frac{y(t_{i+1})-y(t_{i-1})}{2\Delta t} \end{cases} \tag{11}$$

$v_x(t_i)$ and $v_y(t_i)$ are the velocities at the i^{th} time point in the x and y directions, respectively. $x(t_{i+1})$ and $x(t_{i-1})$ and $y(t_{i+1})$ and $y(t_{i-1})$, are the displacements in the x and y directions at the $i+1^{th}$ and $i-1^{th}$ moments, respectively. Δt is the time interval.

2.3. Dynamic collision modeling

1) Bench position derivation: for each bench dragon's motion, we must calculate the position of his two handles, the former handle position coordinates are set as (x_{i-1}, y_{i-1}), then the position expression of the latter handle is:

$$\begin{cases} x_i = x_{i-1} + Lcos\theta \\ y_i = y_{i-1} + Lsin\theta \end{cases} \tag{12}$$

The previous handle position coordinates satisfy the solenoidal equation, and L is the bench length.

2) Collision Detection: We need to consider two cases of benches colliding with each other: one is the collision of neighboring benches, one is the collision with the outer benches.

We employ the Separation Axis Theorem for collision detection between the two bench dragons, which can be represented as rectangles. To begin, we calculate the coordinates of the four corner points for each rectangle. The edge vectors for both rectangles are determined by computing the vectors between their adjacent corner points; only two edges are necessary, as the opposite sides of the rectangles are parallel. We utilize the edges of each rectangle as potential projection axes, as these may serve as separation axes.

Next, we project the vertices of the rectangles onto the current axis to obtain the projected values for each vertex along that axis. For a valid comparison, we identify the minimum projection value (P_{min}) and the maximum projection value (P_{max}) for each rectangle on that axis. If the projections of the two rectangles do not overlap—meaning the maximum projection of one rectangle is less than the minimum projection of the other—it indicates that the rectangles do not intersect. If, after checking all possible separation axes, we find overlap in the projections along each axis, it confirms that the two rectangles do indeed overlap.

2.4. Establishment of optimal pitch model

According to the arc length equation:

$$ds = \sqrt{(dx)^2 + (dy)^2} \tag{13}$$

And the velocity of the leading handle is constant as $v = 1m/s$, we can substitute the expression for x, y in the Cartesian coordinate system to derive:

$$s = vt = \int_0^\theta \sqrt{(b\cos\theta - b\theta\sin\theta)^2 + (b\sin\theta + b\theta\cos\theta)^2} d\theta \tag{14}$$

Simplifying the expression, we can finally establish the differential equation capable of describing the variation of the angle θ with time t as the faucet's front handle moves along the screw thread trajectory:

$$\frac{d\theta}{dt} = -\frac{1}{a\sqrt{1+\theta^2}} \tag{15}$$

According to the solenoidal trajectory equations of motion we can derive the coordinates of the handles of the benches in the dragon's body, because the length of the bench is fixed and rigid, and each bench we need to calculate the position of its two handles, combined with the solenoidal equations can be obtained:

$$(k\theta \cos\theta - x_1)^2 + (k\theta \sin\theta - y_1)^2 = D^2 \tag{16}$$

Where (x_1, y_1) is the position of the front handle, D is the distance between the handles, and θ is the angle corresponding to the rear handle that we need to solve. After having the equations of motion of the bench dragon, we can analyze the motion state of the bench dragon during the process of discing along the spiral into the turnaround interval, and we propose an improved iterative algorithm to simulate the simulation.

2.5. Steps of Improved Iterative Algorithm

We propose an improved golden section search algorithm based on the traditional golden section method to make it more efficient and stable in the iteration process [8].

Improved golden section algorithm method steps: (1) Calculate the initial split point c and d: $c = a + \text{resphi} \cdot (b - a)$ and $d = a + b - c$. Calculate the value of the objective function at c and d : $f(c) =$The value of the objective function at c. $f(d) =$ The value of the objective function at d.

(2) Define the SAT collision detection function: Utilize the Separation Axis Theorem (SAT) to compute the boundaries of the object and identify the separation axes based on these boundaries. Assess whether the projections of the object on all the separating axes overlap in order to determine if a collision has occurred. Additionally, calculate the model of the bench dragon at each pitch in the objective function and apply SAT to check for potential collisions.

(3) Update interval: compare the objective function values and update the search interval: if $f(c) < f(d)$, update the right endpoint b to d:

$$\begin{cases} b = d = c \\ c = a + \text{resphi} \cdot (b - a) \\ f(d) = f(c) \end{cases} \tag{17}$$

If $f(c) \geq f(d)$, then update the left endpoint a to be c:

$$\begin{cases} a = c = d \\ d = a + b - c \\ f(c) = f(d) = New\ objective\ function\ value \end{cases} \tag{18}$$

(4) Iterative update: Calculate the interval length h: $h = b - a$. If h is less than or equal to the tolerance tol, terminate the iteration and return the midpoint of the optimal solution:

$$\text{optimal} = \frac{a+b}{2} \tag{19}$$

(5) Visualize the optimization process: use Matplotlib to plot the points and convergence curves of the optimization process, including the initial points, the points in the iteration process, and the optimal solution.

(6) Output: Print the value of the optimal pitch P and save the image of the optimisation process and the convergence curve.

2.6. Integrated dynamic path planning and optimization model based on genetic algorithm

Our integrated dynamic path planning and optimization model utilizes a genetic algorithm alongside SAT collision detection to address the head-turning path planning issue for a dragon dance team. The movement process consists of three phases: disc in, turnaround, and disc out, with each phase defined by specific geometric models and constraints [9].

The turnaround path is modeled with a header curve comprising two circular arcs, adjustable by manipulating the arcs' radius and center position. The optimization aims to minimize the total path length while ensuring no collisions occur between adjacent benches, using the SAT algorithm for collision detection.

The model transforms the path planning challenge into a constrained nonlinear optimization problem, focusing on key factors such as:

1) Geometric properties of screw motion, represented by parametric equations.

2) Parametric representation of the header curve, primarily adjusting the arcs' parameters for a shorter path.

3) Continuity constraints to maintain smooth transitions, ensuring the position, velocity, and acceleration are consistent throughout.

4) Geometric constraints that limit the turnaround path within a 9-meter diameter circular area, ensuring tangency to the disc-in and disc-out points and checking for collisions among neighboring benches.

The optimization process involves:

(1) Population initialization: Generating random candidate solutions for the turnaround path.

(2) Adaptation function calculation: Evaluating the fitness based on path length and collision detection results.

$$Fitness = L + \lambda \times \sum_{i=1}^{n} Penalty(i) \tag{20}$$

Where L is the total length of the path and $Penalty(i)$ is the collision penalty value detected by SAT at the ith time step. If a collision occurs, the fitness value increases.

(3) Selection, crossover and mutation: select individuals with higher fitness and generate the next generation of candidate solutions through crossover operations, while mutation operations (e.g., adjusting the radius of the arc or the position of the center of the circle) with small probability are carried out on some of the solutions in order to enhance the diversity of the population.

(4) Iterative optimization: the genetic algorithm gradually improves the candidate solutions through multi-generation evolution until the fitness value converges to find the global optimal solution. The iteration stops either when the maximum number of generations is reached or when the fitness values in the population no longer change significantly.

3. Results and discussion

In this problem, we begin by establishing the Archimedes solenoidal equation, noting that the dragon's head is situated in the 16th circle A. To determine the positions of each segment over time, we develop an algorithm that iteratively draws circles based on the position of the dragon's head.

The computational framework commences by defining a primary circular boundary with radius $r_1 = 286$cm centered at the dragon's head coordinate system. Through the simultaneous integration of the solenoidal field equations and the geometric constraints of this circular domain, we analytically derive the positional coordinates for the first body segment. These coordinates subsequently serve as the origin for a secondary concentric circle with reduced radius $r_2 = 165$cm where intersection analysis incorporates kinematic updates between the head segment and its adjacent body unit, as shown in Fig 1.

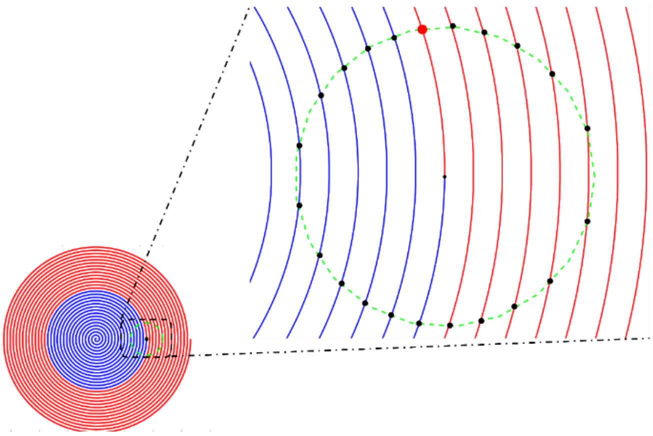

Figure 1. Finding the point of the dragon's body and tail

To resolve solution multiplicity inherent in the geometric configuration space, constraint optimization is implemented via Archimedean solenoidal integration. The search algorithm adopts a radial expansion protocol, systematically evaluating candidate positions from proximal regions near the origin until identifying solutions that satisfy the boundary condition $r_{\text{current}} > r_{\text{center}}$ thereby ensuring continuity in the dragon's articulated motion chain.

We show the computer results for special points, the positions and velocities of the front handle of the dragon head, the front handle of the dragon body and the rear handle of the dragon tail in the 1st, 51st, 101st, 151st, and 201st sections behind the dragon head at 0 s, 60 s, 120 s, 180 s, 240 s, and 300 s: (see Tables 1 and 2 for the results).

Table 1. Position of some handles at special times

	0 s	60 s	120 s	180 s	240 s	300 s
head y (m)	8.800000	5.799240	-4.084851	2.963648	2.594533	4.420254
body x (m)	0.000000	5.771062	-6.304503	6.094762	5.356724	2.320468
body y (m)	8.363879	7.456700	-1.445588	5.237016	4.821125	2.459623
101st x (m)	2.826376	3.440521	-7.405859	4.359746	3.562074	4.402398
101st y (m)	2.898113	5.673043	5.347815	1.881463	4.931520	6.227856
151st x (m)	-9.922815	8.011180	-7.567440	8.475297	6.368696	3.951199
151st y (m)	10.865581	6.701703	2.413245	1.031091	2.990457	7.054687
201st x (m)	1.804523	8.118303	9.721138	9.421706	8.390546	4.370043
201st y (m)	4.585141	6.592432	10.622660	9.301674	7.479327	7.478716
Tail x (m)	10.712042	9.045155	1.392857	4.215308	6.153372	5.234259
Tail y (m)	-5.337796	7.335988	10.976816	7.410747	3.277765	1.823229

Table 2. Velocity of some handles at particular times

	0 s	60 s	120 s	180 s	240 s	300 s
head (m/s)	0	1	1	1	1	1
1^{st} (m/s)	0	0.999376	0.999225	0.999046	0.998701	0.998103
51^{st} (m/s)	0	0.999206	0.999000	0.998761	0.998399	0.997499
101^{st} (m/s)	0	0.998983	0.999058	0.998395	0.998193	0.997130
151^{st} (m/s)	0	0.998874	0.998896	0.998436	0.997929	0.996829
201^{st} m/s)	0	0.998896	0.998778	0.998410	0.997771	0.996587
Tail (back) (m/s)	0	0.998833	0.998781	0.998434	0.997766	0.996478

Based on the establishment of the collision detection model described above, we use an iterative detection method to solve the problem. The motion trajectories of the dragon's head and each section of the dragon's body are generated by the solenoidal equation and converted to the Cartesian coordinate system. According to different time points t, we can get the motion trajectory and coordinate information of each section of the bench.

Since the overlap detection between benches will lead to a significant reduction in computational efficiency, we use one bench to perform overlap detection with only the closest one (as shown in Fig 2. the first set of overlap rate detection between benches is performed when running for 376 s), which greatly improves the computational efficiency while maintaining a high accuracy rate.

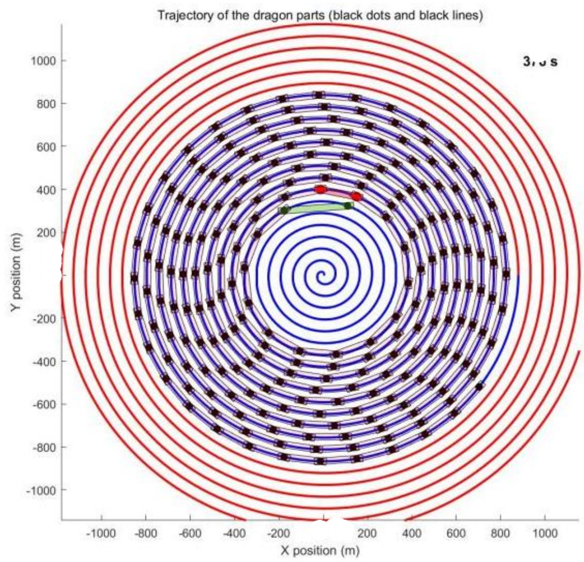

Figure 2. Overlap rate detection between the first set of benches at 376 s of operation

According to the dynamic collision detection model solution based on SAT, the final termination moment of the dragon dance team's disk-in is 413s, which is caused by the collision between the dragon head and the inner test, as shown in Fig 3:

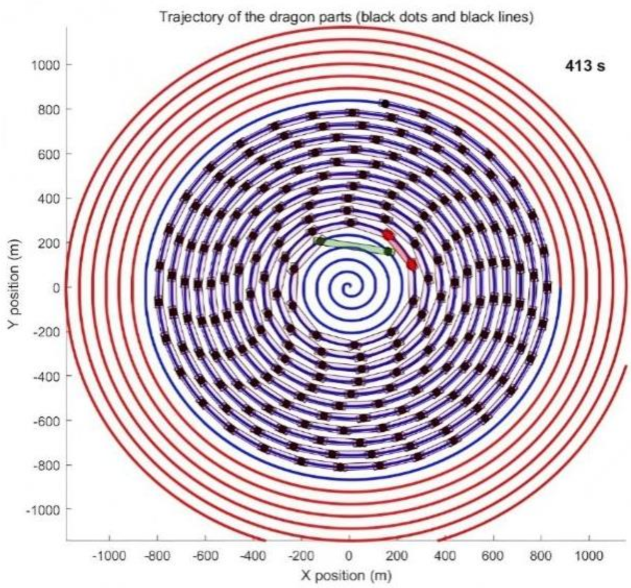

Figure 3. Overlap rate detection between the first set of benches performed at run 376 s

In the model we established above, in the continuous iteration of the improved golden section search algorithm, the pitch trajectory differential equation in each iteration is solved using the Runge-Kutta method to ensure its efficiency and accuracy. Then the positions of the dragon head and various parts of the dragon body are calculated, followed by collision detection to screen the iterative process, and we successfully determine the optimal pitch P so that the bench dragon meets the condition of no collision during the movement [10].

By writing python code to solve we get the result as shown in Fig 4 and Fig 5:

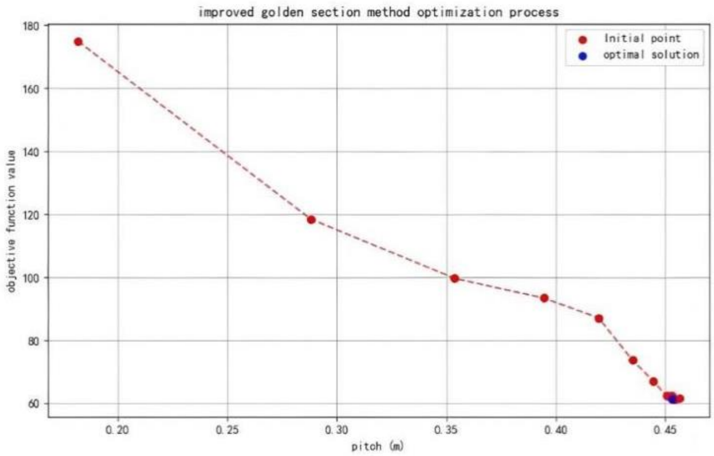

Figure 4. Improved golden section optimisation process

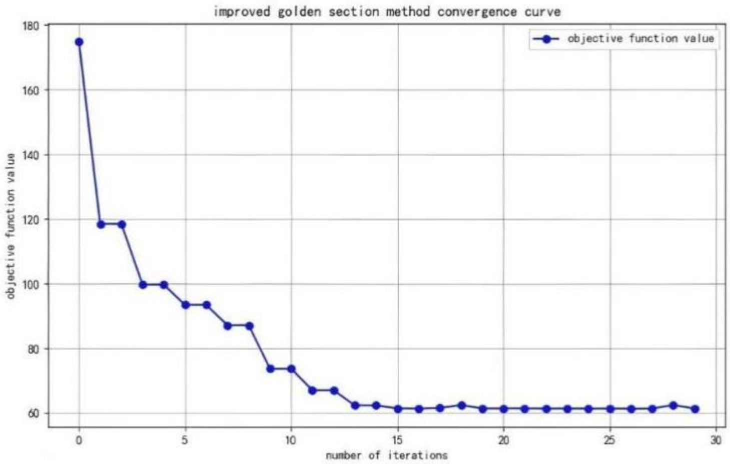

Figure 5. Convergence curve of improved golden section method

It can be seen that the optimum pitch is approximately 0.4529m.

Through the observation of the convergence curve of the improved golden section method, it is found that its objective function tends to be stable to reach the convergence of the efficiency and stability are high.

We use genetic algorithms for fast convergence and optimal solutions, as shown in Table 3:

Table 3. Optimal solutions for circular arcs

	first circular arc	second circular arc
Coordinates of the center of the circle (x, y)	(-0.76, -1.3057)	(1.7359, 2.4484)
Radius (m)	3.0054	1.5027
Radian difference of the arc (rad)	3.071	3.021
Arc length of the arc (m)	9.0803	4.5403
Total length of the two arcs (m)	13.6206	

(1) Plate-in phase (-100s-0s)

In this phase, the bench dragon keeps moving closer to the center (as shown in Fig 6), and the inner circle shrinks smaller and smaller without affecting the operation, reflecting the ingenuity of the spiral motion.

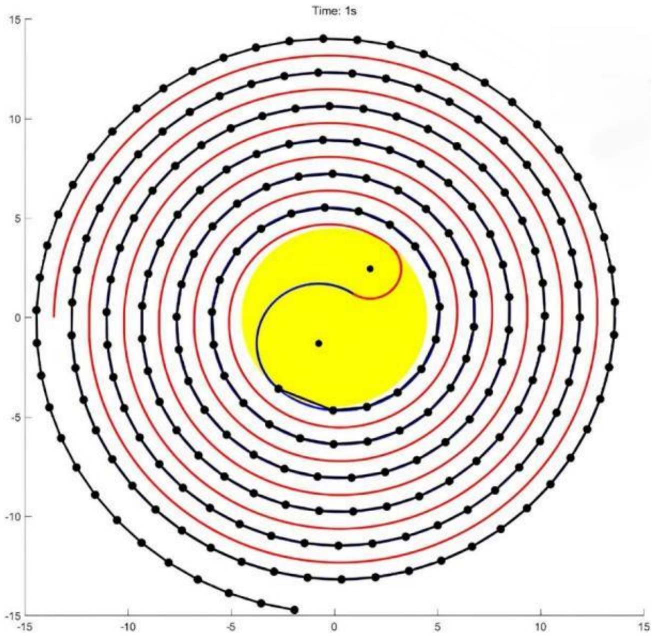

Figure 6. State at disc entry

(2) Turn-around stage (1s-14s)

When the bench dragon enters the beginning of the circular arc, it begins one of the most critical and complicated stages, through the turn-around area in the appropriate range, this trajectory is composed of an S-shaped circular arc, the first stage of the circular arc is large, the second stage of the circular arc is small, so that hundreds of meters of bench dragon can smoothly and quickly change the direction of travel, in this section of the speed and direction of the speed and direction of the obvious differences in the changes occurred.

(3) Disc out stage (15s-100s)

In this stage, it starts to come out of the inner circle through the spiral (as shown in Fig 7), with just the right pitch so that the in and out circles do not interfere with each other and have a good visualization.

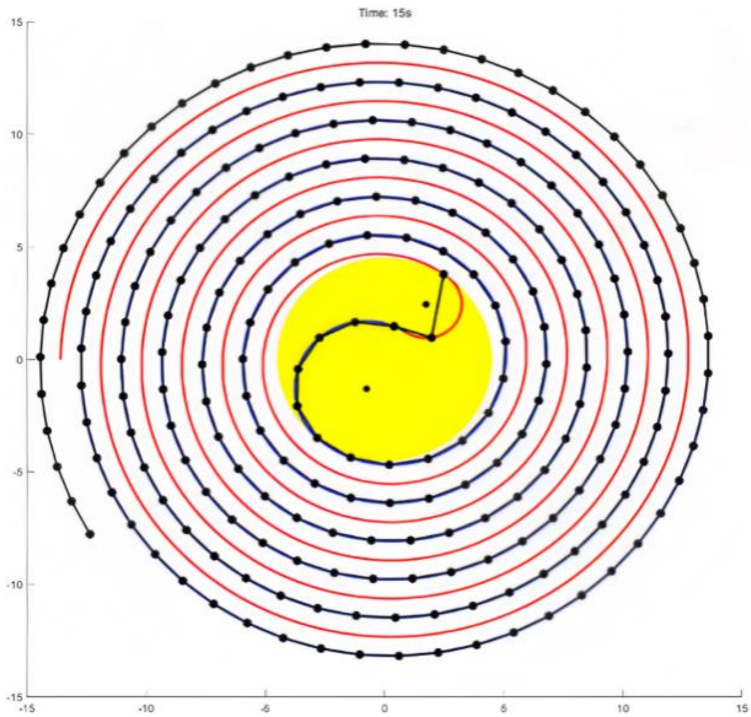

Figure 7. Condition at disc out

4. Conclusion

This study addresses the challenges of dynamic path planning and real-time collision detection in traditional Bench Dragon dance performances by proposing an integrated optimization framework that synergizes an enhanced golden section search algorithm, collision detection based on the Separation Axis Theorem (SAT), and a genetic algorithm. Through rigorous simulation experiments and numerical analysis, the framework demonstrates high efficiency and robustness in coordinating large-scale rigid-chain movements. Key findings include: (1) The enhanced golden section search algorithm, incorporating dynamic search space adjustment, converges to an optimal spiral pitch of 0.4529 m, reducing the total spiral path length to 13.6206 m while ensuring collision-free trajectories via SAT-based real-time detection. (2) The genetic algorithm further optimizes turnaround paths by minimizing the total arc length to 13.0206 m under geometric continuity constraints. (3) Velocity control precision is validated, with the dragon head maintaining a stable speed of 1 m/s and segmental velocity deviations below 0.003 m/s. However, idealized assumptions—such as rigid linkages and 2D motion—limit the framework's applicability in dynamic multi-physics scenarios. Future research will prioritize real-time dynamic modeling, 3D motion extensions, and multi-agent collaborative optimization to advance practical applications in swarm robotics and automated logistics. This work establishes both theoretical foundations and technical pathways for digitizing traditional cultural performances, bridging academic innovation and engineering practicality.

References

[1] Fukaya N, Yamane K, Masuda S, et al. Four-axis adaptive fingers hand for object insertion: FAAF hand [C] // 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024: 13996-14003.

[2] Yang Z, Wang T, Ji X, et al. Bench dragon motion simulation based on fourth-order RK algorithm and ABM prediction [C] // 2024 International Seminar on Artificial Intelligence, Computer Technology, and Control Engineering (ACTCE). IEEE, 2024: 320-325.

[3] Noroozi M, Mohammadi H, Efatinasab E, et al. Golden search optimization algorithm [J]. IEEE Access, 2022, 10: 37515-37532.

[4] Deng W, Zhao H, Zou L, et al. A novel collaborative optimization algorithm in solving complex optimization problems [J]. Soft Computing, 2017, 21: 4387-4398.

[5] Maia J G R, Vidal C A, Cavalcante-Neto J B. Transformation semantics: An efficient approach for collision detection [C] // 2006 19th Brazilian Symposium on Computer Graphics and Image Processing. IEEE, 2006: 94-104.

[6] Liu X, Jiang D, Tao B, et al. Genetic algorithm-based trajectory optimization for digital twin robots [J]. Frontiers in Bioengineering and Biotechnology, 2022, 9: 793782.

[7] Liu Q, Huang G, Zhang X, et al. Speed planning and interpolation algorithm of Archimedes spiral based on tangential vector [J]. International Journal of Precision Engineering and Manufacturing, 2024, 25 (11): 2235-2248.

[8] Zhang Y, Pang D. Research on path planning of mobile robot based on improved ant colony algorithm [C] // 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2022: 558-563.

[9] Jose C, Vijula Grace K S. Optimization based routing model for the dynamic path planning of emergency vehicles [J]. Evolutionary Intelligence, 2022, 15 (2): 1425-1439.

[10] Zhai W, Tao D, Bao Y. Parameter estimation and modeling of nonlinear dynamical systems based on Runge–Kutta physics-informed neural network [J]. Nonlinear Dynamics, 2023, 111 (22): 21117-21130.