Journal of Network Intelligence ©2023 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **8**, Number **3**, August **2023**

3D Animated Group Path Control Based on Improved Gravitational Search

Shuyan Liu*

Department of Film and Television Media
Liaocheng University Dongchang College
Liaocheng 252000, China
563178960@qq.com

Yan Liu

School of Media and Technology
Liaocheng University
Liaocheng 252000, China
liuyan@lcu.edu.cn

*Corresponding author: Shuyan Liu

Received February 21, 2022, revised March 24, 2022, accepted May 20, 2022.

Abstract. *In 3D animation, group intelligence algorithms are often required to control group behaviour in order to improve the efficiency and fidelity of the animation. However, the group motion trajectory needs to satisfy both consistency and individual independence, which places high demands on the group intelligence algorithm. Compared with traditional intelligent algorithms, the convergence ability and convergence efficiency of the gravitational search algorithm are significantly higher. Therefore, this paper proposes to improve the gravitational search algorithm and apply it to the path planning of 3D group animation characters. The proposed method can generate the motion paths of the group; thus realistic animation-aided design can be achieved. Firstly, a combinatorial chaotic mapping is used to generate the initial group for the behavioural characteristics of the group, which improves the group diversity. Secondly, the particle update formulation is improved using an adaptive movement strategy to enhance the ability of the gravitational search algorithm to jump out of the local optimum. Then, real-time collision detection and collision avoidance methods are introduced to solve the individual collision problem. Finally, the path data are imported into Maya software for group motion simulation tests. The experimental results show that the proposed algorithm can realistically simulate large-scale group motion, which greatly improves the efficiency of creating group animation motion paths and is suitable for various complex animation scenes.*

Keywords: group animation; path planning; gravitational search algorithm; chaotic mapping; collision avoidance

1. **Introduction.** Group animation, short for group behavioural animation, is an animation that simulates the behaviour of a group of organisms in a particular environment [1,2,3]. In 2022, Ma et al. [4] solved this problem by proposing behavioural animation, based on the principle that complex animations of groups can be generated automatically by the simple behaviour of each individual and the interaction between individuals. In nature, many species of animals always survive in groups. There are many animals that live in groups like sheep and cattle, such as bee colonies, bird colonies, ant colonies and so on. Each animal in a colony is an individual, but the individuals make up the whole. Through simple group decisions they can generate great power against external threats.

In recent years, group animation has been used extensively in the fields of computer graphics, virtual reality, public safety, film and television technology, and especially in virtual reality. Large-scale group animation in film scenes can create stunning visual effects. Group animation has become an integral part of today's film production. Humans are the most complex and independent intelligences in nature [5,6,7]. There is a clear diversity of human behaviour in everyday life. Each person has a different personality and behavioural style. Therefore, how to realistically simulate the behaviour of people in a realistic environment has been the focus of research in virtual reality technology. Virtual cities enable rational planning of buildings and various facilities.

The simulation of group behaviour in animals in a computer is called group animation and requires making each individual have the same behavioural patterns and goals [8,9]. A group behaviour model, is an abstract and mathematical description of the behavioural characteristics of a group in the real world. Group animation has been widely used in games, film and television animation, military and so on, and it brings people a stunning sensory experience. However, the more individuals there are in a group animation, the more time and effort is consumed by traditional production methods. This is due to the fact that individual behaviour is involuntary, limited and inflexible. Therefore, the behaviour of the individuals in a group needs to be regular and random. The design of path planning methods in group animation has been a hot topic of research for many years [10].

Compared to traditional intelligent algorithms, the convergence capability and convergence efficiency of the gravitational search algorithm(GSA) is significantly higher. Therefore, this paper proposes to improve GSA and use it for path planning of 3D group animation characters to generate individual motion paths to achieve realistic animation-assisted design. The path data of the proposed algorithm is tested in Maya software for collision and group clustering and verified to be effective. The simulation results show that the proposed algorithm allows individuals to automatically overcome inter-group collision problems and blocking problems during group motion.

1.1. **Related Work.** The study of group behaviour animation simulation techniques has a long history in computer graphics [11]. Group motion simulation models can be divided into macroscopic and microscopic models for different requirements of individual detail. Macro studies aim to simulate large-scale group behaviour in real time [12]. A global perspective is taken to extract the common characteristics of a large-scale group and to ignore the description of individual behavioural details in the group. Micro studies treat each person in a group as an individual and give each individual an independent character, thinking and position. Each individual's response to the same thing may reflect great variability because each individual's position is independent of the other [13].

Duives et al. [14] proposed a macroscopic crowd movement model based on continuum dynamics. Jiang [15] studied the aggregation characteristics of pedestrians and proposed a macroscopic pedestrian simulation model containing several adjustable parameters, such as average pedestrian speed, pedestrian density and pedestrian flow. The macroscopic crowd simulation model considers the crowd from a holistic perspective. The macroscopic crowd simulation model sets the same movement characteristics for each individual in the crowd, which can effectively describe the global crowd movement state as a whole, and has a low requirement for computer resources. However, the macroscopic crowd simulation model has difficulty in representing the interactions between each individual in the group and therefore cannot accurately describe the random movement of different individuals when they encounter obstacles. It can be seen that the macroscopic crowd behaviour simulation model is not well suited to the application.

Due to the above shortcomings of macroscopic crowd simulation models, microscopic crowd simulation models have become the focus of simulation research in recent years. In contrast to macroscopic crowd motion models, microscopic crowd motion models treat each person as an individual. Currently, the main microscopic crowd movement models are magnetic field models [16], metacellular automata models [17], queueing theory models [18] and group intelligence models [19]. Collectively, the crowd intelligence model takes into account more comprehensive factors and can simulate the crowd movement process well.

As the application of swarm animation becomes more and more widespread, research on swarm path planning is also increasingly developed. Youssef et al. [20] used genetic algorithms to make simulated annealing algorithms more adaptive. To address the problem that swarm algorithms tend to fall into local optimum, Meshkati and Safi-Esfahani [21] proposed a swarm algorithm based on particle swarm optimisation, which has greatly improved in convergence speed, accuracy and robustness. Tirkolaee et al. [22] proposed an adaptive weighted artificial fish swarm algorithm for the premature convergence problem of path planning. Dulam et al. [23] used an adaptive weighted artificial fish swarm algorithm by modelling the process of panic generation and contagion under disaster situations, a new crowd simulation method was proposed. Wang et al. [24] proposed a path planning method combining data-driven and deep reinforcement learning that not only simulates the movement behaviour of real crowds, but also flexibly adapts to the dynamic changes of simple scenarios. Thilagavathi and Thanamaniet [25] proposed a crowd intelligence algorithm based on the firefly algorithm. Sharma et al. [26] proposed a crowd animation simulation method based on the cuckoo search algorithm (CSA) to improve the intelligence of behaviour control.

Artificial intelligence algorithms have evolved over the decades and a large number of advanced algorithms have emerged in addition to classical algorithms such as particle swarm (PSO) [27]. Current features of group behaviour in group animation include.

(1) Simplicity: each individual intelligence in the whole is relatively simple.

(2) Robustness: the overall group behaviour is largely unaffected by changes in individual behaviour.

(3) Self-organisation: In a group, the movement of individuals may appear random, but as a whole, it is consistent.

(4) Independence: Each individual in the group is independent of each other and behaves differently.

However, as application scenarios become more complex and diverse, the best performing algorithms cannot be used to solve all problems, so targeted improvement measures are constantly required to better solve the various problems that arise. Therefore, this work delves into the gravitational search algorithm. Compared to traditional intelligent algorithms such as PSO and CSA, the convergence capability and convergence efficiency of the gravitational search algorithm have significant advantages. While retaining the strong performance of the gravitational search algorithm, this work improves it.

1.2. **Motivation and contribution.** Although path planning methods for groups in 3D animation have been refined over the years, there are still some shortcomings, for example.

(1) The group animation process is less efficient. In traditional animation, each individual in a group has to plan a route, and the individuals in these routes cannot collide with each other. The traditional animation method is more laborious.

(2) The path planning process lacks interaction. While the designer considers the paths of individuals to achieve collision avoidance, he does not consider the interaction between

the group and the virtual environment and between the group and the group, resulting in a lack of realism in the movement of the group.

(3) Higher requirements for computing resources equipment. The 3D group animation production process requires high-end CPU processors and graphics cards to make more accurate and realistic animation effects, but also greatly increases the production costs [28].

The main focus of this work is on how to solve the group motion blockage and individual collision problems while ensuring the realism and efficiency of group animation. To solve these problems, some group intelligence algorithms need to be further investigated. In contrast to other approaches, group intelligence algorithms have their origins in the simulation of animal evolution and competitive behaviour in nature. It has a strong advantage in solving various process arrangement and behavioural route planning problems. Gravitational search algorithm, as one of the hot research group intelligence algorithms, has strong global search capability and convergence speed. Therefore, this paper improves the gravitational search algorithm and applies it to the path planning of 3D group animation characters in order to achieve realistic group animation behaviour control. The main contributions of this work are shown as follow:

(1) Combinatorial chaotic mapping is used to generate initial groups for group behavioural characteristics and to improve group diversity.

(2) The particle update formula is improved using an adaptive shift strategy to improve the ability of the gravitational search algorithm to jump out of the local optimum, laying a better foundation for subsequent group animation path planning.

(3) Real-time collision detection and collision avoidance methods are introduced to solve the individual collision problem. Finally, the path data were imported into Maya software for group motion simulation tests. Modelling was carried out in the Maya environment and the experimental effects of the improved gravitational search algorithm were analysed so as to verify its effectiveness.

The rest of the paper is organized as follows: Section 2 introduces the basic principle of gravity search algorithm. Section 3 describes the group path control of three-dimensional animation based on improved GSA. Section 4 presents the experimental results and test analysis. Section 5 concludes the paper.

2. **Gravitational search algorithm.**

2.1. **Fundamentals of the Gravitational Search Algorithm.** The Gravitational Search Algorithm (GSA) is an optimization algorithm inspired by the law of gravity as well as Newton's second law [29]. The phenomenon of gravity is shown in Figure 1. Similar to most intelligent algorithms, the GSA algorithm finds the optimal solution by the regular movement of particles, the difference being that GSA's particles move with the help of gravity. During the generation selection calculation, the greater the fitness of the particle, the heavier the mass of the particle and the greater the attraction. The smaller the fitness of the particle, the less the mass of the particle, the less the attraction, and the easier it is to be attracted to and moved by particles with larger masses. Thus, the locally optimal particle of each generation has the heaviest mass and the greatest attraction. The massive particles attract other particles closer and move very slowly themselves. As the algorithm is computed over many iterations, the small masses are pushed by the gravitational force of the large masses until the particle position is output as the global optimum when the convergence conditions are met.

The flow of the gravitational search algorithm is as follows [30].

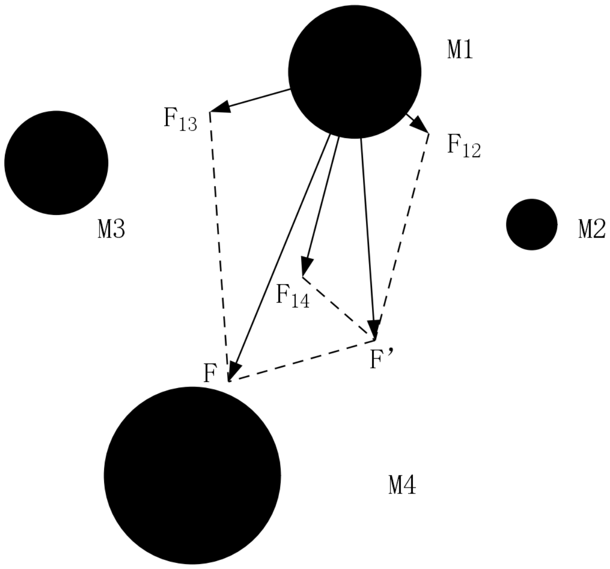

Figure 1. Gravitational phenomenon.

Step 1: Initialize the group: the position of each particle in the search space is as follows.

$$X_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^n\right), i = 1, 2, \cdots, N \tag{1}$$

Where N is the number of particles, n is the dimension, and x denotes the position of particle i in dimension d.
Step 2: Calculate the initial group fitness.
Step 3: Calculate the particle inertial mass M. The mass of a particle is defined by the fitness $m_i(t)$.

$$m_i(t) = \frac{fit\, t_i(t) + worst(t)}{best(t) - worst(t)} \tag{2}$$

Where: t is the current algebraic, $fit\, t_i(t)$ is the fitness of the particle, $worst(t)$ and $best(t)$ are the minimum and maximum values of the fitness of the group respectively. To simplify the calculation, the mass of the particle is normalized to the inertial mass.

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{3}$$

Step 4: Calculate the acceleration of the particle a. From Newton's second law, we need to find the gravitational force between the particles first.

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{4}$$

Where $M_i(t)$, $M_j(t)$ are the inertial masses of particle i and particle j respectively, ϵ is a minimal constant, $M_{ij}(t)$ is the Euclidean distance between the two particles i and j, and $G(t)$ is the gravitational constant. The value of $G(t)$ varies with the number of selected iterations.

$$[G(t) = G_0 \times \exp\left(-\alpha \times \frac{t}{\text{maxiter}}\right) \tag{5}$$

Where α is the decay coefficient, G_0 is the initial gravitational constant, and maxiter is the maximum number of selected iterations. There is a gravitational ensemble between particles in the group.

$$F_i^d(t) = \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t) \tag{6}$$

Where $rand_j$ is a random number in the interval [0,1]. Finally, according to Newton's second law, the particle acceleration is shown as follows.

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{7}$$

Step 5: Update the particle position: The particle velocity and position update equations are shown below.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{8}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{9}$$

Step 6: Repeat the above steps until the end condition is met. The exact flow of the gravitational search algorithm is shown in Figure 2.

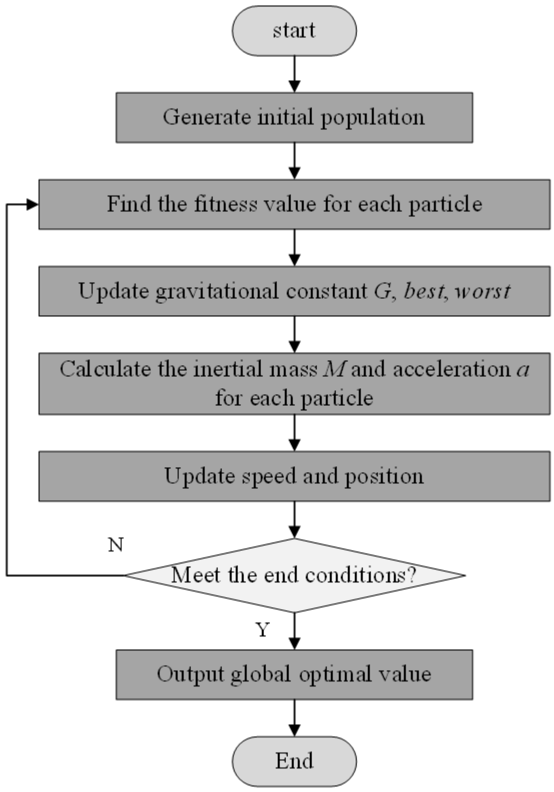

Figure 2. The process of the GSA.

2.2. **Shortcomings of GSA and improvement methods.** Since GSA was proposed, it has received a lot of attention from scholars due to its simple structure, few parameters and strong convergence, etc. GSA has shown more excellent performance in the fields of discrete optimization and deep learning. However, the gravitational search algorithm suffers from the same shortcomings as other intelligent algorithms, such as the poor efficiency of finding the best and the tendency to fall into local optimality when solving complex problems.

The initial group of GSA is simply generated using the rand function. If the deviation between the particles of the initial group and the optimal solution is too large, it will

greatly increase the time for the particles to move towards the optimal solution, which means that the convergence efficiency has a relatively large decrease. In order to improve the traversal of the initial group particles, it is usually necessary to increase the number of initial group particles, but this can seriously affect the efficiency of the operation.
To solve this problem, this work introduces a combined chaos strategy for the initial group. By combining chaos algorithms, the traversal of particles can be effectively improved while the number of group particles remains unchanged, thus improving the algorithm's optimisation-seeking efficiency to a certain extent.
The path of movement of particles in GSA is based on the attraction between particles. The particles are relatively independent of each other and are updated in a fixed way (lack of memorability). As the number of selected iterations increases, the search process of the group slows down. Due to the cumulative effect of the fitness function on the mass, the mass becomes larger during the selection process. At this point, if the global optimum happens to be outside the local search range of the particle, then the particle will not be able to escape the attraction of the local optimum and will attract other particles towards it. In this case, the particle will end its movement prematurely and the search process will stall (losing its global convergence). This is especially true when solving high-dimensional complex optimisation problems, where the particles fall into the local optimum prematurely.
To address the above shortcomings, this work introduces an adaptive movement strategy to improve the particle velocity update formulation based on adaptive theory. By endowing the particle movement memory capability, the algorithm can improve the optimisation finding ability to a certain extent, and thus better escape from the local optimum constraint.

3. **3D animated group path control based on improved GSA..**

3.1. **Research objectives and key issues.** Due to this specificity of group behaviour, the behavioural trajectories of individuals are required to be random and yet must be consistent as a whole in order to ultimately produce naturalistic and realistic group animation effects. Therefore, the study of group path planning in 3D animation requires full consideration of the following factors.

(1) Individuals in a group have their own area of movement and can avoid collisions etc.

(2) The group is unified in the sense that each individual of the whole group moves in the same direction or towards the same goal.

(3) Groups can be divided into multiple subgroups, which can self-organise into new larger groups according to conditions.

(4) Enhancing realism. The trajectory of the individual is independent but also ensures the unity of the whole, in line with the laws of group movement.

(5) Improve group co-ordination. Enhance group intelligence to increase group movement speed.

3.2. **Improved GSA algorithms based on chaotic adaptive theory.** Common chaotic mappings [31] are Logistic, Tent, ICMIC, Bernouilli shift, Chebyshev, Sine.
Logistic mappings are characterised by a high degree of traversal. In addition, the range of values in the logistic mapping is concentrated at the two ends of the search range, and the number of intermediate values is relatively small. The Tent chaos mapping is characterized by a uniform distribution of the range of values in the search range and has a good chaotic perturbation capability. Combining the advantages and disadvantages of

the two chaotic mappings, this work uses a combined chaotic mapping based on Logistic and Tent to implement the initialisation of the group, whose expression is shown as follow:

$$\begin{cases} y_0 = x_0 \\ \text{if} \quad y_0 < 0.5, y_{m+1} = 2y_0 \\ \text{else} \quad y_{m+1} = 2(1 - y_0) \\ tt = \lambda x_0 (1 - x_0) + |y_{m+1}| \\ x_{m+1} = \bmod(tt, 1) \end{cases} \tag{10}$$

Where: x_0 and y_0 are the initial values of the particles, x_{m+1} and y_{m+1} are the updated state values, tt is the state value to be updated, and λ is the logistic chaos coefficient. It can be seen that if the chaos operation is introduced in the process of selecting the initial values of the algorithm, it will increase the particle traversal and improve the convergence speed of the algorithm to a certain extent. A random particle is generated and normalised (the variable is converted to an equivalent value between 0 and 1). Combine the chaos operations to generate the initial group and denormalise it. Calculate the initial group fitness. Move particles and calculate fitness. Determine if the convergence condition is met, if so output the global optimum, otherwise continue to move particles and calculate fitness.

Adaptive means controlling the tendency of variables to change during the solution of a problem according to their changing patterns, so that it has the ability to adjust dynamically. The aim of adaption is to make the system optimal at any given moment. Adaptive algorithms are algorithms that are made to perform better by dynamically adjusting various coefficients.

From the basic principles of the gravitational search algorithm, it is clear that the gravitational search algorithm uses purely particle interaction forces to update the velocity formula. The particle update method is more fixed and lacks memorability. As the number of selected iterations increases, the search process of the group will become slower, or even the search for the best stalls, especially when solving high-dimensional complex optimization problems are prone to the above phenomenon. In order to solve the above problems, this paper refers to the speed update method of PSO algorithm and introduces the global optimal point to achieve speed update as follow:

$$V_i(t+1) = R \times V_i(t) + c_1 \times a_i(t) + c_2 \times (\text{Gbest} - X_i(t)) \tag{11}$$

$$c_1 = -\frac{t^3}{\text{maxiter}^3} + 1 \tag{12}$$

$$c_2 = \frac{t^3}{\text{maxiter}^3} \tag{13}$$

Where Gbest is the global optimal point; c_1 and c_2 are the acceleration factors; and maxiter is the maximum number of iterations to be selected.

3.3. **The process of group path planning.** To explore a wider range of positions, the position of each particle is updated using equation (14).

$$\hat{x}_\iota = \vec{x}_\iota + rand_i\left(\frac{1}{6} \sum_{n \in N_i} \vec{x}_n\right) \tag{14}$$

Where $\hat{x}_\iota$ is the new position of $\vec{x}_\iota$. The new position can be found by calculating the average of the six nearest neighbour positions.

Based on the above improvements, this paper proposes an improved gravitational search algorithm based on chaotic adaptive theory. In the group initialisation phase of the gravitational search algorithm, the initial group diversity is improved by introducing combinatorial chaotic mappings. Enhancing particle traversal helps to improve the convergence efficiency of the algorithm. The global optimal point is then combined with adaptive theory to achieve adaptive velocity update of the particles. This approach gives the particles the ability to memorise, which further improves the convergence efficiency of the algorithm as well as the optimisation finding ability. The process of group path planning is shown in Table 1.

Table 1. The process of group path planning.

Flow of 3D animated group path control based on improved GSA
Inputs: sample group animation data $X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n), 1 \leq i \leq N$; number of particles N .
Outputs: global best value Gbest.
Process:
Step 1: All nodes (agents) are randomly initialized at the beginning.
Step 2: Identify the six nearest neighbours to the data using Euclidean distances.
Step 3: Calculate the particle inertial mass M and acceleration a.
Step 4: Calculate the new position of each particle by using equation (6).
Step 5: The fitness function is used as the objective function to evaluate the fitness of each data.
Step 6: Calculate the quality of the nodes (agents) using the fitness function.
Step 7: Calculate the gravitational force and acceleration of each particle.
Step 8: Adaptive movement of particles to update their position by means of equation (11).
Step 9: Execute Step 1 if the stall occurs on a minimization.
Step 10: Collision between individuals occurs then Step 8 is executed;
Output the best particle with the smallest value of the objective function.

3.4. **Collision detection and methods.** To prevent collisions between individuals, a random collision detection algorithm based on the enclosing box is used. The enclosing sphere of a particle denotes the smallest outer sphere that surrounds the particle, as shown in Figure 3.

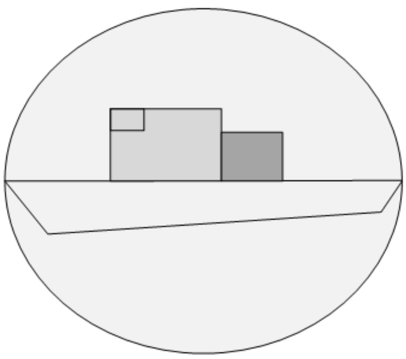

Figure 3. Bounding sphere.

$$R = \{(x, y, z) | (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 < r^2\} \tag{15}$$

Where (x_0, y_0, z_0) denotes the centre of the sphere, which is obtained by calculating the mean value of the coordinates on the particle.and r is the radius. Collision detection of two particles can be judged by the intersection of the enclosing spheres, as follow:

$$\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 < r^2} \leq (r_1 + r_2) \tag{16}$$

Where (x_1, y_1, z_1) and (x_2, y_2, z_2) are the centres of the two enclosing spheres. r_1 and r_2 are the radii of the two enclosing spheres. If the sum of the radii is greater than or equal to the centroid distance, then the two enclosing spheres have collided.

Collision avoidance uses the waitting method and the steering method. Waiting method means that in the event of a potential side or rear-end collision, individuals will wait so that a larger individual can pass first. The steering method means that in the event of head-on, rear-end and side-on collisions, individuals can be steered into other collision-free directions of movement.

4. Experimental results and analysis.

4.1. **Experimental configuration parameters.** In this work, the proposed group path control algorithm is simulated on the Maya game engine. The hardware environment is an Intel(R) CPU I3-12100F@4.3 Ghz processor, 128 GB of RAM, and AMD RX 7900 graphics card. The software environment is Maya 2017 and MATLAB R2016. Three experiments are designed to verify the effectiveness of this algorithm: (i) Experiment 1, simulation experiments of the standard test function of the improved GSA algorithm. (ii) Experiment 2, the simulation experiment of path control of the improved GSA algorithm. (iii) Experiment 3, the validation of the application effect of the improved GSA algorithm under different group size conditions. Experimental parameters: number of particles N=50, maximum iteration maxiter=1000, initial gravitational constant $G_0 = 1$, $\alpha = 30$, acceleration constant $c_1 = c_2 = 1.5$.

4.2. **Experiment I.** The performance of the improved GSA algorithm was tested using four classical test functions and compared with typical GSA [32] and OI-PSO [33].

Each function was run 20 times. For comparison purposes, the parameters used in the GSA algorithm are the same as in the modified GSA. OI-PSO algorithm with inertia weight minimum = 0.4, inertia weight maximum = 0.9 and acceleration constant = 2. The four classical test functions are defined as shown in Table 2.

Table 2. Arguments to test functions

Function Type	Expressions	Search by	Theoretical optimal value
Sphere Function	$f(x) = \sum_{i=1}^{d} x_i^2$	[-80,80]	0
Ackley Function	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e$	[-20,20]	0
Griewank Function	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-250,250]	0
Rastrigin Function	$f(x) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	[-4.06,4.06]	0

This experiment was run 30 times for each of the four test functions using the improved IGSA, GSA and OI-PSO algorithms respectively, and the results of each optimisation were recorded and averaged, retaining five significant digits, and the results are shown in Table 3. It can be seen that the GSA algorithm is inferior to both the OI-PSO algorithm and the Improved GSA algorithm in terms of convergence accuracy. the OI-PSO algorithm outperforms the traditional GSA algorithm, but sometimes it still falls into a local

Table 3. Comparison of optimization results of test functions

Functions		GSA [32]	OI-PSO [33]	Improving GSA
Sphere	Best value	2.0586e-11	2.689le-17	5.3045e-21
	Worst value	1.0589e-07	8.9612 e-16	3.9774e-19
	Average value	4.5506e-08	1.8963e-16	5.7928e-20
Ackley	Best value	1.2952	1.5896e-07	5.5896e-10
	Worst value	1.8411	9.5178e-06	8.5178e-09
	Average value	1.6813	5.3189-07	1.3189e-10
Griewank	Best value	1.1546	0.13586	3.5468e-03
	Worst value	3.8431	0.65685	1.2685e-02
	Average value	2.3125	0.31584	8.3589e-03
Rastrigin	Best value	8.2685	3.2568	0.8321
	Worst value	1.265	6.5874	1.2689
	Average value	9.2168	5.2189	1.1326

optimum. The improved GSA algorithm has improved in accuracy compared with the OI-PSO algorithm, and the optimal values of the improved GSA algorithm are better than the other algorithms, with better performance in finding the optimal value. Therefore, the proposed optimisation algorithm in this paper can show better search accuracy when dealing with multi-dimensional function optimisation problems.

At the same time, to visually compare the algorithm seeking performance, a comparative analysis of the execution efficiency of the above-mentioned algorithms was carried out. The convergence curves of the three algorithms, as shown in Figure 4 to Figure 7.

It can be seen that compared to typical GSA and OI-PSO, the algorithm in this paper takes the least time to obtain the optimal solution, which means the convergence speed is the fastest. This is because the proposed algorithm inherits the powerful optimisation capability of the GSA algorithm, while introducing a combined chaotic mapping to perturb the initial group and guiding the particles to update the speed formula adaptively with the help of the global optimal point Gbest, thus improving the efficiency and accuracy of the optimisation search. A comprehensive analysis shows that the improved GSA algorithm enhances the search speed while ensuring the global optimisation capability.

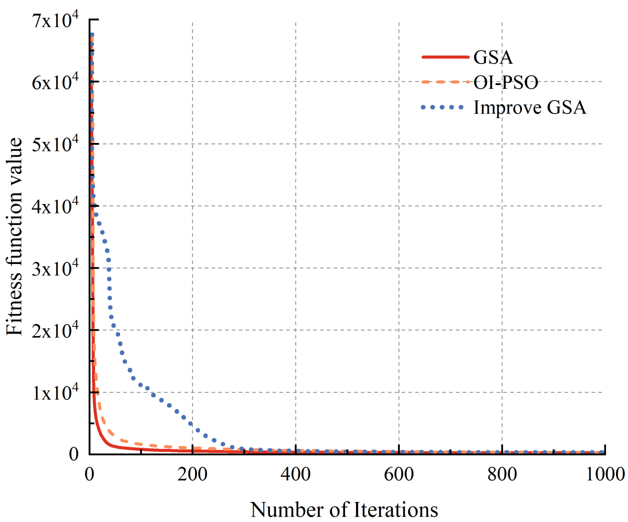

Figure 4. Sphere function

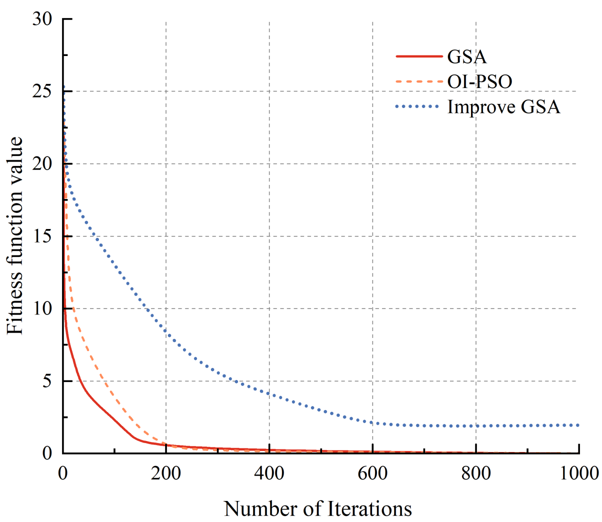

Figure 5. Ackley function

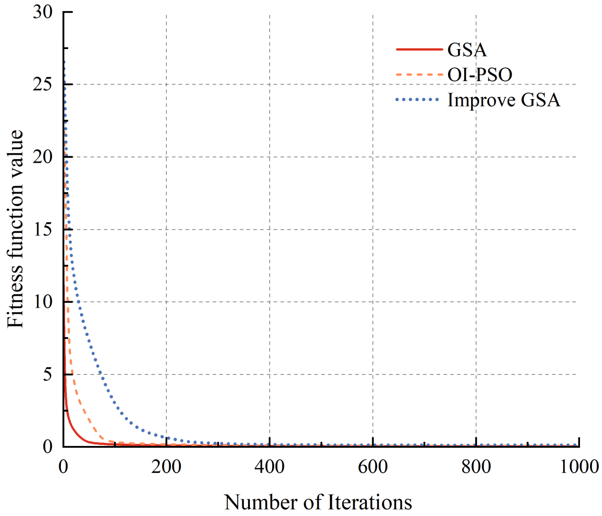

Figure 6. Griewank function

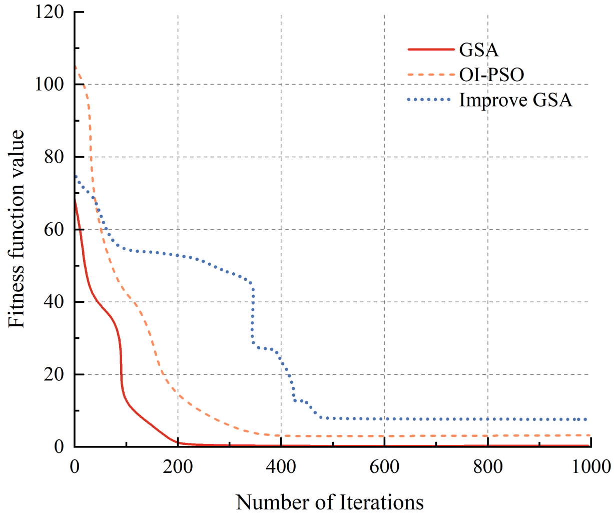

Figure 7. Rastrigin function

4.3. **Experiment II.** Improved GSA algorithm for path control simulation experiments. Firstly, this work meshes the virtual scene, setting the starting point, the end point and the obstacle locations. Then, the improved GSA algorithm is used to generate editable paths. The simulation results of the path generation before and after the improvement are shown in Figure 8 to Figure 11.

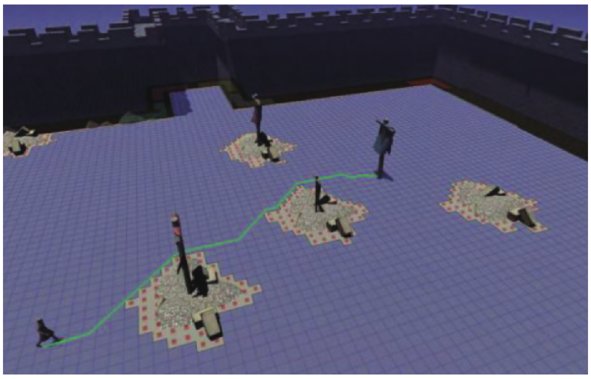

Figure 8. Path 1 generated by GSA

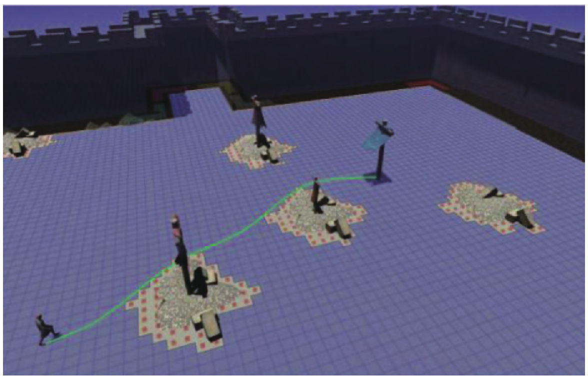

Figure 9. Path 1 generated by improved GSA

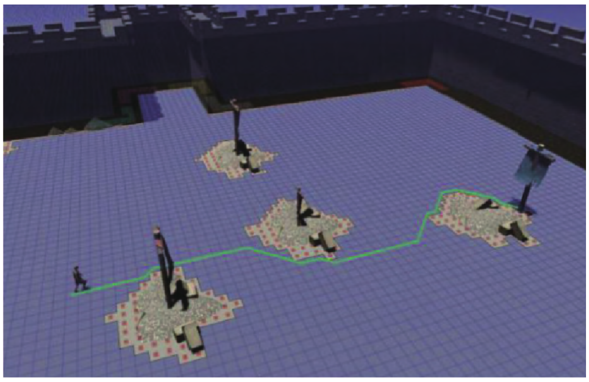

Figure 10. Path 2 generated by GSA

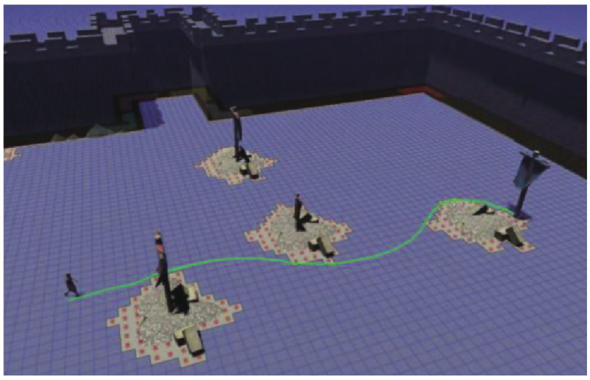

Figure 11. Path 2 generated by improved GSA

It can be seen that the group path control algorithm proposed in this work is able to solve the problems of folded lines, large deflection angle and unsmoothness in the paths well.

4.4. **Experiment III.** Validation of the application of the improved GSA algorithm under different group size conditions.
In order to demonstrate the effectiveness of the group path control algorithm for group animation applications, this work verifies the effectiveness of the improved GSA algorithm for different group sizes, and the simulation results are shown in Table 4.

Table 4. Comparison of optimization results of test functions

Group size	Initial speed/ms^{-2}	Path length/m	Group average speed at obstacle point/ms^{-1}	Number of collisions	Time/ms
100	1	100	1.00	2	6.79
500	1	100	0.90	5	6.83
1000	1	100	0.86	7	6.86

It can be seen that groups of different sizes plan the same path and have the same pathfinding time. The larger the group size, the greater the group density at the obstacle point, resulting in an increase in the number of collisions (resulting in motion blockage). However, the improved GSA algorithm reduces the average group speed during obstacle avoidance and effectively reduces the number of collisions. From the experimental results, it can be seen that the group path control algorithm proposed in this work is intelligent and efficient.

5. **Conclusion.** In this paper, an improved GSA algorithm based on chaos adaptive theory is proposed and applied to group path control in 3D animation. Firstly, a combinatorial chaos mapping is used to generate the initial group for the group behavioural characteristics and to improve the group diversity. Secondly, an adaptive movement strategy is used to improve the particle update formula to enhance the ability of the gravitational search algorithm to jump out of the local optimum, laying a better foundation for the subsequent group animation path planning. Then, real-time collision detection and collision avoidance methods are introduced to solve the individual collision problem. Finally, the path data is imported into Maya software for group motion simulation tests. The results of the standard function tests in the experimental part show that the algorithm proposed in this paper is advanced, and the animation example tests prove the feasibility of the algorithm. In the future, a perceptual model will be added to the animated examples to add emotional conditions for each individual, which could improve the behavioural fidelity generated by group path planning, and therefore further research will follow to address this aspect.

Data Availability. The data used to support the findings of this study are included within the article.

Conflicts of Interest. The author declares that there is no conflict of interest regarding the publication of this paper.

Funding Statement. This work is supported by project of Liaocheng university undergraduate teaching reform research(311162212) and Liaocheng social science planning (NDYB2022171).

Project Fund. Social Science Planning Research Special Project of Shandong Province (No. 21CLYJ13)

REFERENCES

[1] F. Q. Zhang, T.-Y. Wu, and G. Y. Zheng, “Video salient region detection model based on wavelet transform and feature comparison,” *EURASIP Journal on Image and Video Processing*, vol. 2019, 58, 2019.

[2] F. Q. Zhang, T.-Y. Wu, J.-S. Pan, G. Y. Ding, and Z. Y. Li, “Human motion recognition based on SVM in VR art media interaction environment,” *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[3] Y. R. Ma, Y. J. Peng, and T.-Y. Wu, ”Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning,” *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[4] T.-Y. Wu, C.-W. Lin, Y.Y. Zhang, C.-H. Chen, “The Grid-Based Swarm Intelligence Algorithm for Privacy-Preserving Data Mining,” *Applied Science*, vol. 9, no. 4774, 2019.

[5] Y. Tian, Y. Li, L. Pan, and H. Morris, “Research on group animation design technology based on artificial fish swarm algorithm,” *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 2, pp. 1137-1145, 2020.

[6] N. Hoyek, C. Collet, F. Di Rienzo, M. De Almeida, and A. Guillot, ”Effectiveness of three-dimensional digital animation in teaching human anatomy in an authentic classroom context,” *Anatomical Sciences Education*, vol. 7, no. 6, pp. 430-437, 2014.

[7] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, ”Effectiveness of Animation in Trend Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1325-1332, 2008.

[8] C. M. Chen, K.-H. Wang, T.-Y. Wu, E. K. Wang, ”On the security of a three-party authenticated key agreement protocol based on chaotic maps”, *Data Science and Pattern Recognition*, vol. 1, no. 2, pp. 1-10, 2017.

[9] C. M. Chen, L. L. Xu, K.-H. Wang, S. Liu, T.-Y. Wu, “Cryptanalysis and Improvements on Three-party-authenticated Key Agreement Protocols Based on Chaotic Maps”, *Journal of Internet Technology*, vol. 19, no. 3, pp. 679-687, 2018.

[10] C. M. Chen, W.C. Fang, S.A. Liu, T.-Y. Wu, J.-S. Pan, K.-H. Wang, “Improvement on a Chaotic Map-based Mutual Anonymous Authentication Protocol”, *Journal of Information Science & Engineering*, vol. 34, no. 2, pp. 371-390, 2018.

[11] M. Hanif, “The Development and Effectiveness of Motion Graphic Animation Videos to Improve Primary School Students’ Sciences Learning Outcomes,” *International Journal of Instruction*, vol. 13, no. 4, pp. 247-266, 2020.

[12] T.-Y. Wu, X. N. Fan, K.-H. Wang, J.-S. Pan, C.-M. Chen, “Security analysis and improvement on an image encryption algorithm using Chebyshev generator”, *Journal of Internet Technology*, vol. 20, no. 1, pp. 13-23, 2019

[13] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, “State-of-the-art crowd motion simulation models,” *Transportation Research Part C: Emerging Technologies*, vol. 37, no. 11, pp. 193-209, 2013.

[14] Y. Q. Jiang, “Macroscopic Modeling of Pedestrian Flow Considering the Herd Behavior,” *Applied Mechanics and Materials*, vol. 444-445, pp. 906-911, 2013.

[15] C. Amanulla, A. Wakif, and Z. Boulahia, “Numerical investigations on magnetic field modeling for Carreau non-Newtonian fluid flow past an isothermal sphere,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, no. 9, 86, 2018.

[16] O. Okwuashi and C. E. Ndehedehe, “Integrating machine learning with Markov chain and cellular automata models for modelling urban land use change,” *Remote Sensing Applications: Society and Environment*, vol. 5, no. 21, pp. 100461-100473, 2020.

[17] J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella, and J. Rius, “A queuing theory model for cloud computing,” *The Journal of Supercomputing*, vol. 69, no. 1, pp. 492-507, 2014.

[18] Z. Sabir, M. A. Z. Raja, M. Umar, and M. Shoaib, “Neuro-swarm intelligent computing to solve the second-order singular functional differential model,” *The European Physical Journal Plus*, vol. 135, no. 6, 2020.

[19] H. Youssef, S. M. Sait, and H. Adiche, “Evolutionary algorithms, simulated annealing and tabu search: a comparative study,” *Engineering Applications of Artificial Intelligence*, vol. 14, no. 2, pp. 167-181, 2001.

[20] J. Meshkati and F. Safi-Esfahani, “Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing,” *The Journal of Supercomputing*, vol. 75, no. 5, pp. 2455-2496, 2018.

[21] E.-B. Tirkolaee, A. Goli, and G.-W. Weber, “Fuzzy Mathematical Programming and Self-Adaptive Artificial Fish Swarm Algorithm for Just-in-Time Energy-Aware Flow Shop Scheduling Problem with Outsourcing Option,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 11, pp. 2772-2783, 2020.

[22] R. Dulam, K. Furuta, and T. Kanno, “Consumer Panic Buying: Realizing Its Consequences and Repercussions on the Supply Chain,” *Sustainability*, vol. 13, no. 8, 4370, Apr. 2021.

[23] H.-Z. Wang, Y. Wu, G.-Y. Min, J. Xu, and P.-C. Tang, “Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach,” *Information Sciences*, vol. 498, no. 14, pp. 106-116, 2019.

[24] M. D. Thilagavathi and D. A. Thanamani, “Scheduling in High Performance Computing Environment using Firefly Algorithm and Intelligent Water Drop Algorithms,” *International Journal of Engineering Trends and Technology*, vol. 14, no. 1, pp. 8-12, 2014.

[25] N. Sharma, O. Prakash, K. Sengar, S. Chaudhury, and A. Singh, “The relation between emotional intelligence and criminal behavior: a study among convicted criminals,” *Industrial Psychiatry Journal*, vol. 24, no. 1, 54, 2015.

[26] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, “An Overview of Variants and Advancements of PSO Algorithm,” *Applied Sciences*, vol. 12, no. 17, 8392, 2022.

[27] J. Berndt, “Anime in Academia: Representative Object, Media Form, and Japanese Studies,” *Arts*, vol. 7, no. 4, p. 56, 2018.

[28] E. Rashedi, E. Rashedi, and H. Nezamabadi, “A comprehensive survey on gravitational search algorithm,” *Swarm and Evolutionary Computation*, vol. 41, no. 11, pp. 141-158, 2018.

[29] E. Momeni, A. Yarivand, M. B. Dowlatshahi, and D. J. Armaghani, “An efficient optimal neural network based on gravitational search algorithm in predicting the deformation of geogrid-reinforced soil structures,” *Transportation Geotechnics*, vol. 26, no. 16, 100446, 2021.

[30] E. Varol Altay and B. Alatas, “Bird swarm algorithms with chaotic mapping,” *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1373-1414, 2019.

[31] K. M. Ali and M. Khan, “Application Based Construction and Optimization of Substitution Boxes Over 2D Mixed Chaotic Maps,” *International Journal of Theoretical Physics*, vol. 58, no. 9, pp. 3091-3117, 2019.

[32] Y. Song, J. Yang, R. Xie, and Y. Wu, “Dynamic positioning model of offshore oil drilling platform based on OIPSO algorithm,” *Future Generation Computer Systems*, vol. 124, pp. 168-173, 2021.