The Population-Based Optimization Algorithms for Role Modelling and Path Generation in Group Animation

Hong Liu
School of Information Science and Engineering, Shandong Normal University
Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology
Jinan City, P.R.China
Email: lhsdcn@jn-public.sd.cninfo.net

Yuanyuan Li and Hanchao Yu
School of Information Science and Engineering, Shandong Normal University, Jinan City, P.R.China
Email: lyysdnu@126.com yuhanchao@ict.ac.cn

***Abstract*—Traditional animation by key frame techniques takes animators lots of time and vigour to model and simulate behaviours of crowds. For solving this problem, this paper presents a novel group animation generation approach based on population-based optimization algorithms. It is mainly divided into two parts. First, it puts forward a role modelling approach based on dynamic self-adaptive genetic algorithm and NURBS technology. Second, following the introduction to PSO (Particle Swarm Optimization) algorithm, a group path generative approach is presented. It simulates group behaviours, including cohesion and separation, dynamic object tracking and collision avoidance. Finally, a group of shark modelling and path generation images are exhibited as examples.**

***Index Terms*—group animation, role modelling, genetic algorithm, PSO algorithm, path generation**

I. INTRODUCTION

Group animation has been a main topic in the field of computer animation and game. Traditional key frame techniques took animators lots of time and vigour to model and simulate vivid behaviours of crowds or flocks as each individual's behaviour needs to be scripted carefully to meet the following requirement: motion of the whole group should be harmonious while the individuals in the group look like independent and moving stochastically.

One of the important characteristics of group animation is its ability to reduce the workload on the animators. This is achieved by letting a behavioural model automatically take care of the low-level details of the animation, freeing the animator to concentrate on the big picture. Freeing the animator from low-level animation details is even more important when dealing with crowds.

An animator manually creating a crowd animation is overwhelmed not only by the large number of animated entities, but also by the interactions between them. If all tracks are made by animator one by one, the work will be very heavy while the effect is not satisfied.

Population-based optimization algorithms find near-optimal solutions to the difficult optimization problems by motivation from nature. A common feature of all population-based algorithms is that the population consisting of possible solutions to the problem is modified by applying some operators on the solutions depending on the information of their fitness. Hence, the population is moved towards better solution are as of the search space. Two important classes of population-based optimization algorithms are evolutionary algorithms and swarm intelligence-based algorithms.

Genetic algorithm is a kind of evolutionary algorithms that transforms populations of individual objects into new populations using operations patterned after natural genetic operations and fitness proportionate reproduction. Genetic algorithms begin with an initial population of individuals and then iteratively (1) evaluate the individuals in the population for fitness and (2) perform genetic operations on various individuals in the population to produce a new population.

PSO (Particle Swarm Optimization) algorithm is a swarm intelligence-based algorithm and has a dominance to perform better in simulating group actions. Because it was inspired by the activities of social animals or insects in nature, it shows more smooth and genuine as a basis for group animation,.

This paper presents a novel group animation generation approach for role modelling and path generation in group animation. It puts forward a role modelling approach based on dynamic self-adaptive genetic algorithm and NURBS technology. A group of shark design example is illustrated for showing the modelling process. After then, a group animation path generative approach based on PSO algorithm is introduced It simulates group behaviours, including cohesion and separation, dynamic object tracking and collision avoidance. The generative path can be used as the movement track of the group. In

This paper is supported by the National Natural Science Foundation of China (No.61272094) and the Ph.D. Programs Foundation of Ministry of Education of China (No. 20093704110002), Natural Science Foundation of Shandong Province (No. ZR2010QL01) and Shandong Provincial Key Laboratory Project.

doi:10.4304/jsw.9.1.26-36

this way, the animators will save time and be absorbed in the design of the main roles and scenes.

The remainder of this paper is organized as follows. Section 2 is related work in group animation. Section 3 introduces a group animation role modelling approach based on dynamic self-adaptive genetic algorithm and NURBS technology. In section 4, a group animation path generative approach is presented and showed how to use particle swarm optimisation algorithm to generate group animation paths. The last section summarises the paper and gives an outlook for the future work.

II. Related Works

Virtual groups have been studied since the early days of behavioral animation. The seminal work by Reynolds [1] is considered the first one in the field of behavioral animation. It presented a method to animate large groups of entities called boids, which present behaviors similar to those observed in flocks of birds and schools of fishes. Reynolds started from the premise that the group behavior is just the result of the interaction between the individual behaviors of the group members. Therefore, it would suffice to simulate the reasonably simple boids individually, and the more complex flocking behavior would emerge from the interaction between them.

Tu and Terzopoulos [2] created a realistically rich environment inhabited by artificial fishes. The complexity of the under sea life, including interactions between fishes like predators hunting preys, mating, and schooling, was obtained by modeling the behavior of individual fishes: group behaviors emerged as the individuals interacted.

Going even further in the direction of using more realistic models of the simulated entities, Brogan and Hodgins [3] described an algorithm to control the movements of entities with significant dynamics that travel in groups. By significant dynamics, the authors mean that the work is focused on simulating systems whose dynamics are complex enough to have a strong impact on the motion of the simulated entities.

Researches are being conducted to the use of path planning algorithms associated to generation of realistic movements of the found path. Lavalle [4] introduced the concept of a Rapidly-exploring RandomTree (RRT) as a randomized data structure for path planning problems. Choi et al. [5] proposed a model based on a probabilistic path planning and hierarchical displacement mapping to generate a sequence of realistic movements of a human-like biped figure to move from a given start position to a goal with a set of prescribed motion clips. Metoyer and Hodgins [6] proposed a method for generating reactive path following based on the user's examples of the desired behavior. Zhang et al. [7] presented a framed-quadtree based on reversed d* path planning approach for intelligent mobile robot and Gong et al. [8] introduced a multi-objective particle swarm optimization algorithm for robot path planning

More specifically concerning groups' motion, Rodríguez et al. [9] proposed a model using a road map providing an abstract representation of global environment in formation to achieve different complex group behaviors that cannot be modeled with local information alone. Lien and collaborators [10] proposed ways using roadmaps to simulate a type of flocking behavior called shepherding behavior in which outside agents guide or control members of a flock. Foudil et al. [11] designed a system to simulate pedestrian behaviour in crowds in real time, concentrating particularly on collision avoidance. On-line planning is also referred to as the navigation problem.

Rodrigues [12] and Bicho et. al [13] presented a method for crowd simulation based on a biologically motivated space colonization algorithm. The proposed crowd modelling method is free-of-collision and suited to the interactive control of simulated crowds.

Data-driven models are quite recent in comparison with other methods, and aim to record motion in a preproduction stage or to use information from real life to calibrate the simulation algorithms. One example was proposed by Musse et al. [14] described a model for controlling groups' motions based on automatic tracking algorithms and proposed a new model to quantitatively compare global flow characteristics of two crowds [15].

Although there are many approaches for modeling and path generation in computer animation, only a few researchers try to use population-based optimization algorithms for this purpose. This paper presents a novel group animation modelling and path generation approach based on population-based optimization algorithms.

III. The Role Modeling Based on Dynamic Self-Adaptive Genetic Algorithm and NURBS Technology

For the character of a large scale individuals in group animation, the role modelling in group animation by evolutionary algorithm possesses unexampled advantage. Each individual in group have to be different and each of them have to be similar in group animation. Fortunately, evolutionary algorithm is suitable to solve this kind of problem. It is based on simulating the process of natural selection and reproduction on a computer. This technique depends on the specification of a parameterized model that is general enough to allow a wide variety of possible outcomes of interest to the animator. These outcomes are similar with the seed while each of them are different. It just is accord with request of role modelling. The role modelling based on dynamic self-adaptive genetic algorithm and NURBS technology will be introduced at following section.

A. NURBS Technology

Non-Uniform Rational B-Splines encompass almost every other possible 3D shape definition. A NURBS curve is defined as:

$$C(t)=\frac{\sum_{i=0}^{n}\varpi_i P_i N_{i,k}(t)}{\sum_{i=0}^{n}\varpi_i N_{i,k}(t)} \quad (1)$$

Where P_i are the control points, ϖ_i are the weights of P_i, and $N_{i,k}(t)$ are B-Spline basis functions, i=0,1,2,⋯,n.

The recursive definition of B-Spline basis functions $N_{i,k}(t)$ is:

$$N_{i,0}(t)=\begin{cases}1 & t\in[t_i,t_{i+1}]\\ 0 & t\notin[t_i,t_{i+1}]\end{cases} \quad (2)$$

$$N_{i,k}(t)=\frac{t-t_i}{t_{i+k}-t_i}\times N_{i,k-1}(t)+\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}\times N_{i+1,k-1}(t) \quad t\in[t_k,t_{n+1}] \quad (3)$$

In which, t_i are the values of knots, and the knot vector is $T=(t_0,t_1,\cdots,t_{n+k+1})$.

From the definition equation of NURBS curve, it can be seen that the shapes of NURBS curves can be changed by moving control points, altering the weights of control points, and The knot vector is a sequence of parameter values that determines where and how the control points affect the NURBS curve. In this paper, the approach of altering shape is used by moving control points while the structure lines are being adjusted.

There are many creation approaches of NURBS models, such as transshipping to the basic NURBS elements, revolving or lofting to NURBS curves and so on. Therefore, NURBS models can be created by outlining structure lines and then lofting them. In this way, the structure lines of a successful NURBS model are extracted first. Then the extracted structure lines are adjusted. Finally, these structure lines are lofted to generate the other model.

There are both U curves and V curves on NURBS surfaces. U curves are landscape orientation curves while V curves are longitudinal orientation curves on model surface. A NURBS model can be rotten by lofting its U curves or V curves. Therefore, extracting structure curves is to draw out and copy these U and V curves and then adjusted them to form new model.

B. The Coding of Genetic Algorithm

Solving a given problem with genetic algorithm starts with specifying a representation of the candidate solutions. Such candidate solutions are seen as phenotypes that can have very complex structures. There are many coding methods, such as binary coding, grey coding, real coding, symbol coding, tree-structure coding, hybrid coding, and so on. In this paper, the scale factor of structure curves is taken as gene and real coding is used to express scale value. The number of gene bits is decided by the structure curve number of the model.

For example, a model with 7 structure curves can be expressed (0.3) (1.0) (2.2) (1.4) (0.8) (1.9) (1.2), in which (0.3) denotes the scale value of the first curve is 0.3 times of the first curve of the seed.

C. Fitness

Select a current best model by designer, and the fitness of every individual is decided by the similar degree with the best individual. It is calculated according to the radios between the structure curves of an individual with the structure curves of the best individual. The more similar an individual is with the best individual, the higher its fitness value is.

Definition 1 The structure curve radius r_i: The average of the distances between the control points to the center point at the ith structure curve.

Definition 2 The ratio of the current structure curve $Current_i$: The ratio between the radius of the ith structure curve r_i at the current individual and the radius of the first structure curve.

Definition 3 The best ratio of the structure curve $Best_i$: The best ratio between the radius of the ith structure curve r_i at the current individual and the radius of the first structure curve.

$$fitness=\frac{1}{\sum_{i=1}^{n}\frac{Best_i-Current_i}{Best_i}} \quad (4)$$

According to (4), the more similar an individual with the best individual, the higher its fitness value.

D. Self-adaptive Dynamical Adjustment

The performance of genetic algorithm is greatly affected by cross probability P_C and mutation probability P_M. If the unsuitable values of P_C and P_M are used in GA, it is easy to sink into local optimal value early. For solving this problem, an approach of dynamical adjusting to P_C and P_M are presented as following.

First of all, an estimation function for checking whether the current individual has sunk into its local optimal value is defined.

Definition 4 Let x be an individual in current population, $h(x)$ is the fitness value of x, $h_{\max}$ is the fitness value of the best individual in current population, ε is an arbitrary small positive number. The estimation function $k(x)$ is defined as:

$$k(x)=\frac{h(x)}{h^2(x)-(h_{\max}-\varepsilon)^2} \quad (5)$$

Next, a property of the estimation function is proved. Let us differentiate for equation (5) from both side and get:

$$k'(x)=\left(\frac{h(x)}{h^2(x)+(h_{\max}-\varepsilon)^2}\right)'=\frac{h'(x)\left[(h_{\max}-\varepsilon)^2-h^2(x)\right]}{\left(h^2(x)+(h_{\max}-\varepsilon)^2\right)^2} \quad (6)$$

Let $\Omega_1(x^\circ)=\{x\mid h(x)<h(x^\circ)\}$

$\Omega_2(x^\circ)=\{x\mid h(x)>h(x^\circ)\}$

$\Omega_3(h_{\max})=\{x\mid h(x)<h_{\max}\}$

$\Omega_4(h_{\max})=\{x\mid h(x)>h_{\max}\}$

Suppose $x^\circ\in\Omega$ is one local maximum value of $h(x)$. From (5), we get the following property:

When $(h_{\max}-\varepsilon)^2-h^2(x)>0$,

i.e. $h(x)<h_{\max}-\varepsilon$, $k'(x)$ and $h'(x)$ with the same sign, then

(1) If x° is the local maximal value of $h(x)$, then x° also is the local maximal value of $k(x)$, and the attract region of x° is $\Omega_1(x^\circ)\bigcap\Omega_3(h_{\max})$;

(2) If x° is the local minimal value of $h(x)$, then x° also is also is the local maximal value of $k(x)$, and the attract region of x° is $\Omega_2(x^\circ)\bigcap\Omega_3(h_{\max})$。

When $(h_{\max}-\varepsilon)^2-h^2(x)<0$,

i.e. $h(x)>h_{\max}-\varepsilon$, $k'(x)$ and $h'(x)$ with the different sign, then

(1) If x° is the local maximal value of $h(x)$, then x° is the local minimal value of $k(x)$, and the attract region of x° is $\Omega_1(x^\circ)\bigcap\Omega_4(h_{\max})$;

(2) If x° is the local minimal value of $h(x)$, then x° is the local maximal value of $k(x)$, and the attract region of x° is $\Omega_2(x^\circ)\bigcap\Omega_4(h_{\max})$.

From the above consequence, theorem 1 can be gotten. It can be used to estimate whether the current individual is in the global optimum attract region, then dynamically adjust cross probability P_C and mutation probability P_M accordingly.

Theorem 1 Let $h_{\max}-\varepsilon$ be a threshold of the global optimum value (i.e. if $h(x^\circ)>h_{\max}-\varepsilon$, then regard x° as the global optimum value). x^Δ Locates in the global optimum attract region, if and only if $\left[h(x)-h(x^\Delta)\right]\left[k(x)-k(x^\Delta)\right]<0$ is true, $\forall x$ in the adjacent region of x^Δ.

According to theorem 1, we can estimate a current individual x^Δ belongs to the global optimum attract region or the local optimum attract region, and then dynamically adjust P_C and P_M based on the following formula:

$$\begin{cases} P_C=P_C+\alpha\times d\times\dfrac{\left[h(x)-h(x^\Delta)\right]\left[k(x)-k(x^\Delta)\right]}{\left|h(x)-h(x^\Delta)\right|\left|k(x)-k(x^\Delta)\right|} \\ P_M=P_M+\beta\times d\times\dfrac{\left[h(x)-h(x^\Delta)\right]\left[k(x)-k(x^\Delta)\right]}{\left|h(x)-h(x^\Delta)\right|\left|k(x)-k(x^\Delta)\right|} \end{cases} \quad (7)$$

In which, $\alpha=\min\{P_C,1-P_C\}$, $\beta=\min\{P_M,1-P_M\}$, $d=\min\left\{\left|h_{\max}-h(x^\Delta)\right|/(h_{\max}-h_{\min}),\ 1\right\}$, $h_{\min}$ is the fitness value of the worst individual in the population and $h_{\max}$ is the fitness value of the best individual in the population.

The formula 3 is used as following:

If the current individual x^Δ locates in the global optimum attract region, then on the basis of original cross probability P_C, mutation probability P_M and the distance d with the best individual in the population, dynamically decrease the cross probability P_C and mutation probability P_M between 0 and 1, in order to save the individuals located in the global optimum attract region.

If the current individual x^Δ locates in the local optimum attract region, then on the basis of original cross probability P_C, mutation probability P_M and the distance d with the best individual in the population, dynamically increase the cross probability P_C and mutation probability P_M between 0 and 1, in order to avoid sinking into local optimum solution.

The adjusting process is:

(1) For every individual x^Δ in the population, calculate the fitness value $h(x^\Delta)$ and the value of estimate function $k(x^\Delta)$;

(2) $\forall x$ in the adjacent region of x^Δ, calculate the fitness value $h(x)$ and the value of estimate function $k(x)$;

(3) Calculate

$$d=\min\left\{\left|h_{\max}-h(x^\Delta)\right|/(h_{\max}-h_{\min}),\ 1\right\}$$

(4) Let

$$P_C=P_C+\alpha\times d\times\frac{\left[h(x)-h(x^\Delta)\right]\left[k(x)-k(x^\Delta)\right]}{\left|h(x)-h(x^\Delta)\right|\left|k(x)-k(x^\Delta)\right|}$$

(5) Let

$$P_M=P_M+\beta\times d\times\frac{\left[h(x)-h(x^\Delta)\right]\left[k(x)-k(x^\Delta)\right]}{\left|h(x)-h(x^\Delta)\right|\left|k(x)-k(x^\Delta)\right|}$$

The above dynamical adjusting operation can increase the convergence of genetic algorithm and enhance the diversity of the population obviously.

E.The Elitism Strategy

When dynamically adjusting, it is possibly to cause the crossover probability and the variation probability become very large, and thus destroy the outstanding individuals which will appear in the evolution process.

This paper adopts the elitism strategy to retain the outstanding individual. Supposes the population number is n, the strategy is that after every genetic operations, sort the new generated population B by the fitness value of the individual, take front 30 percent invividuals and merge them with the last generation population, get the polulation A. Sorting the individuals in A according to their fitness values, take front 30 percent invividuals in A and compare them with the individuals of the last generation population. If there are k ($k \leqslant 30\%$) individuals belonging to the last generation, randomly eliminate k individuals in A and take the front k individuals merge into A. The elitism strategy make it to be possible to guarantee the current outstanding individuals not to be destroyed by heredity operations. This will be a important guarantee condition for the convergence of the algorithm.

IV. A Design Example to Illustrate the Execution of Genetic Algorithm

In this section, we introduce a shark model design example (Figure 1) to show the modelling process. A complex design can be divided into several part design and then assembled them. For briefly, we only introduce the modelling process of a shark body.

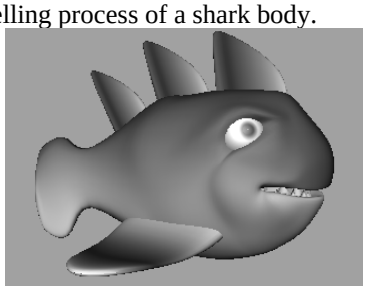

Figure 1. A shark model

Step 1 Initializing the population of chromosomes.

Create a model by animator or import a model from model base, as figure 1 and divided its body model as figure 2.

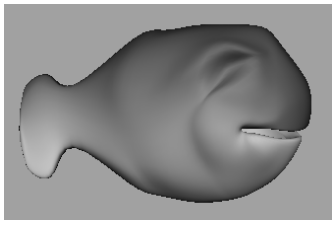

Figure 2. The body of the shark model

(2) Obtain the structure curves number is oparm_num from both U and V orientations by getAttr(Obj.spanU) and getAttr(Obj.spanV) functions.

(3) Draw out the structure curves from both U and V orientations by Duplicate Curve function. There are 16 extracted structure curves at V orientations and they are shown as figure 3.

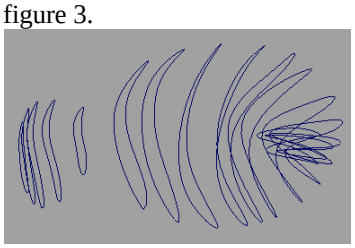

Figure 3. The extracted NURBS curves

(4) Obtain the control point positions of every structure curves and save them in two-dimensional array ep_point[M][N]. There are 8 control points at every structure curve and they are shown as figure 4.

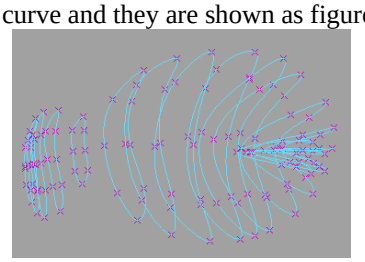

Figure 4. The control points

(5) Reconstitute the structure curves by curve function and the data information of the control points in array ep_point[M][N].

(6) Take a random number between 0 and 1 as scale rate and select a control point randomly at the structure curve, use scale function to scale the structure curves one by one.

(7) Loft the adjusted structure curves to form a new model.

(8) Repeat (5) to (7) until the number of new models to appointed speed size.

Step 2 Count the fitness for each individual in the population according to equation (4).

Step 3 Form a new population according to each individual's fitness based on elitism strategy.

Step 4 Perform crossover, mutation operations on the population.

(1) Crossover

The primary reproductive operation is the crossover operation. The purpose of this operation is to create two new models that contain genetic information inherited from two successful parents. The main crossover ways of real coding based genetic algorithm include signal point and multi-points crossover. A multi-points crossover way is used in this example.

The coding of two parents:

Left: （1.0） （1.13） （0.91） （0.95） （1.08） （0.99） （**1.16**） （0.92） （0.85） （0.92） （1.19） （**0.97**） （0.94） （1.11） （1.12） （1.03）

Right:（1.0）（0.90）（1.10）（0.96）（1.00）（0.98）（**1.00**）（1.14）（1.11）（0.87）（1.09）（**0.87**）（1.12）（1.01）（1.17）（1.06）

Two crossover points (7 and 11) are selected. After crossover, the coding of two children are gotten as following.

Left:（1.0）（1.13）（0.91）（0.95）（1.08）（0.99）（1.00）（0.92）（0.85）（0.92）（1.09）（0.97）（0.94）（1.11）（1.12）（1.03）

Right:（1.0）（0.90）（1.10）（0.96）（1.00）（0.98）（1.16）（1.14）（1.11）（0.87）（1.19）（0.87）（1.12）（1.01）（1.17）（1.06）

The NURBS curves of two parents and two children after crossover can be seen as figure 5 and figure 6.

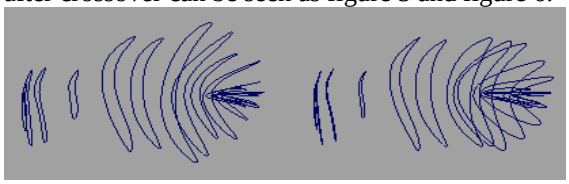

Figure 5. The NURBS curves of two parents

Figure 6. The NURBS curves of two children after crossover

The model of two parents and two children after crossover are shown as figure 7 and figure 8.

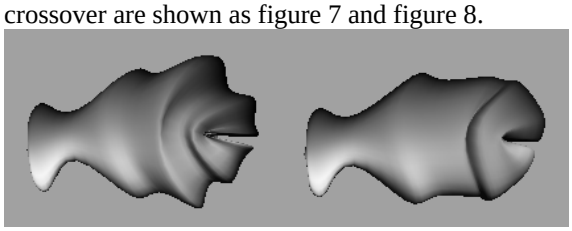

Figure 7. The models of two parents

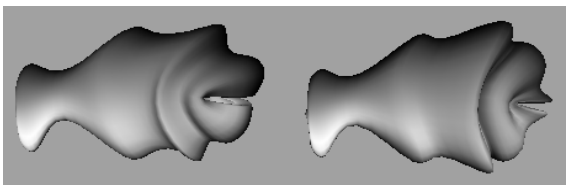

Figure 8. The models of two children after crossover

(2) Mutation

The mutation operation is used to enhance the diversity of trees in the new generation thus opening up new areas of ‘solution space’. It works by random selecting multiple structure curves in a single parent and change their scale radios accordingly.

The coding of two parent：

（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）

Two mutation points (5 and 7) are selected. After mutation, the coding of two children are gotten as following.

（1.0）（1.0）（1.0）（1.0）（1.364）（1.0）（1.069）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）（1.0）

The NURBS curves of the parent and the child after mutation can be seen as figure 9 and figure 10.

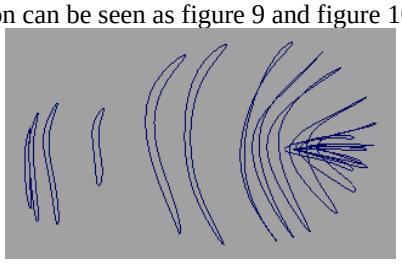

Figure 9. The NURBS curves of parent

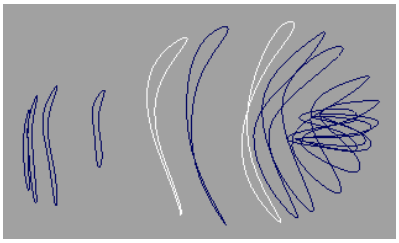

Figure 10. The NURBS curves after mutation

The models of the parent and the child after mutation are shown as figure 11 and figure 12.

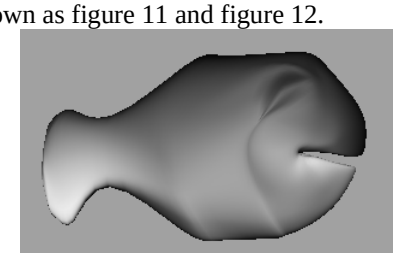

Figure 11. The model of parent

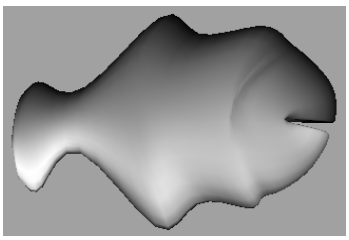

Figure 12. The model after mutation

Step 5 If the procedure doesn’t been stopped by the animator, go to step 2.

This process of selection and crossover, with infrequent mutation, continues for several generations until the animator stop it. Then the detail design will be done by animators with human wisdom.

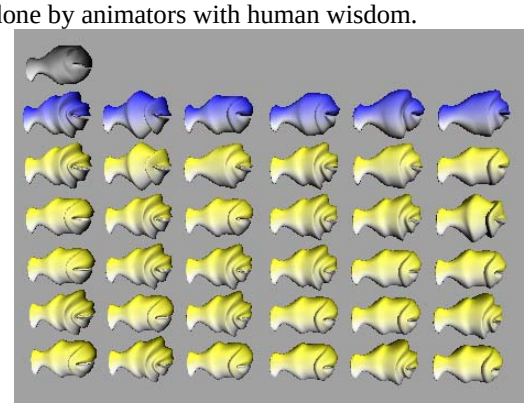

Figure 13. The fish bodes after five generation evolution

Figure 13 shows some generated fish bodes after five generation. The generated components and designed components by animators are classified and saves in a SQL sever based database. These components then are assembled and treated by animators to form a group of roles, and to be used at the following group animation.

V. Group Animation Path Generation Based on PSO Algorithms

A.Particle Swarm Optimization (PSO) Algorithm

Although the size of group is very large, the bird flocks can harmonious achieve the sudden change of direction, group separation and cohesion without collision. This phenomenon inspires many researches in related domain[16,17]. Social animals or insects in nature often exhibit a form of emergent collective behavior known as 'flocking'. The flocking model is a bio-inspired computational model for simulating the animation of a flock of entities. It represents group movement as seen in the bird flocks and the fish schools in nature. In this model, each individual makes its movement decisions on its own according to a small number of simple rules that it reacts to its neighboring members in the flock and the environment it senses. These simple local rules of each individual generate a complex global behavior of the entire flock.

Particle swarm optimization (PSO) was originally introduced by J. Kennedy and R. Eberhart [Kennedy and Eberhart, 1995] in 1995 as an optimization technique. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. Each individual in PSO is assigned with a randomized velocity according to its own and its companions' flying experiences, and the individuals are then flown through hyperspace. In the Standard PSO model, each individual is treated as a volumeless particle in the D-dimensional space, with the position and velocity of ith particle represented as $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$ and $V_i = (V_{i1}, V_{i2}, \cdots, V_{iD})$. The particles move according to the following equation:

$$V_{id} = w * V_{id} + c_1 * rand(\cdot) * (P_{id} - x_{id}) + c_2 * Rand(\cdot) * (P_g - X_{id}) \quad (8a)$$

$$X_{id} = X_{id} + V_{id} \quad (8b)$$

where c1 and c2 are positive constant, rand() and Rand() are two random functions in the range of [0,1]. Parameter w is the inertia weight introduced to accelerate the convergence speed of the PSO. Vector $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ is the best previous position (the position giving the best fitness value) of particle i called pbest, and vector $p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ is the position of the best particle among all the particles in the population and called gbest.

In PSO algorithm, a particle decides where to move next, considering its own experience is the memory of its best position, and the experience of its most successful neighbour. At each iteration, the particle pbest with the best fitness in the local neighbourhood and the current particle are combined to adjust the velocity alone each dimension, and that velocity is then used to compute a new position for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position is considered the cognition component, and the portion influenced by the best in the neighbourhood is the social component.

Particle swarms, in common with many population based algorithms, have a dominance to perform better in simulating group actions. Because it was inspired by the activities of social animals or insects in nature, and as a basis for group animation, it shows more smooth and genuine.

The basic group animation consists of four basic steering behaviours:

(1) Cohesion: Steering the individuals in the group to collect to a position;
(2) Separation: Steeling the individuals in the group to separate from one position;
(3) Dynamic object track: Steering group toward the tracking object and keep a suitable distance each other;
(4) Collision avoidance: Steering group remains in close proximity while avoiding collisions with other members of the group and with obstacles in the environment.

The path generation of these basic behaviours based on PSO algorithm will be introduced at the following sections one by one.

B. Cohesion and Separation

In natural world, cohesion and separation are most general behaviours. The acquiring foods together and attacking heterogeneous objects are typical clustering behaviour, and getting away from natural enemy is common separate behaviour. In the animation with large scene, these two behaviours are the actions with high frequency. In our animation environment, an improved

PSO algorithm is used for generating path with more actual effect [18].

In natural biologic collecting, when the individual arriving in target point, it always moves around one position and will not concentre to one point. According to this principle, we enact a threshold for controlling the individuals' movement around target point (see (9)). When the individual arrives in the arrange that takes the target point as circle of center and threshold as radial, it will make stochastic movement in the circle.

$$f = \sqrt{[t_x - (o_x + rand(th))]^2 + [t_y - (o_y + rand(th))]^2 + [t_z - (o_z + rand(th))]^2} \quad (9)$$

Where f is fitness value, t_x,t_y,t_z is the current position of particle, o_x,o_y,o_z is the position of the target point, th is inputted threshold value.

The fitness f is counted by the distance between current position of the particle and one position closed with target point to avoid all particles congregate to one point.

Cohesion Algorithm Based on PSO

Step 1 Initialization: initialize the number of particle
the max number of group overlapping MAX;
the position of target point o_x, o_y, o_z;
threshold value th;
the arrange of each axis of original particle;
accelerate constant c_1, c_2;

Step 2 While（the overlapping number < MAX）do
For every particle
{
(1) Count the fitness value (new pbest) of the new position for every particle;
If new pbest >old pbest
Then pbest=new pbest;
(2) Change gbest according to updated pbest in (1);
(3) Update the speed of every particle, and limited them in V_{max};
(4)Update the position of every particle;
}

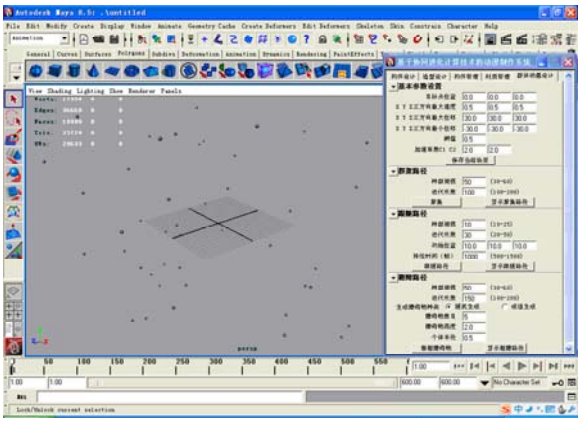

Figure 14. Initial states of the group

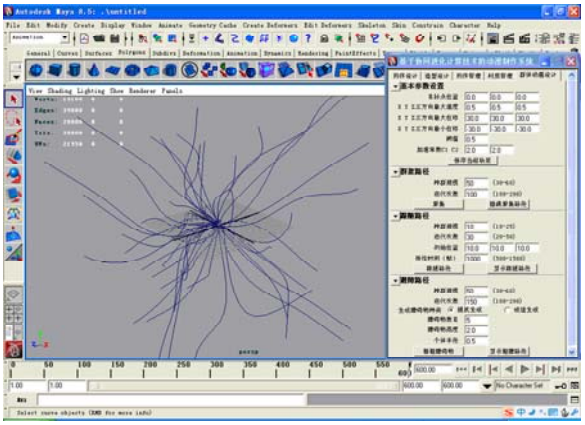

Figure 15. The generative cohesion path

Figure 16. One generated shark cohesion image according to the path in figure 15

Simulated results are as shown in figure 14 and figure 15. Figure 16 is generative shark cohesion according to the path in figure 15. If the different parameters are inputted, various paths with different directions and different types will be generated. The separation takes the opposite fitness to cohesion and is omitted here.

C. Dynamic Object Tracking

Dynamic object tracking behavior among the individuals is also a kind of typical group action in the natural world, such as flight of wide gooses, follow among shoal and so on. Track is an interesting action and can exhibit group intelligence well. The tracking movement in animation can show special effect.

The dynamic object track is implemented by the following process. First of all, erecting initial parameters, and then following the pass points of tracking object, taking tracking object as target position and overlapping MAX times. After that, updating the target position by current position of tracking object,

For each passed point of the tracked object, we take that point as the target position of the individuals of tracking group overlapping MAX times, and then update the target position of the tracking group to the current position of the tracked object, repeat PSO algorithm until the tracked object stop. For every first overlap after updating, the gbest and pbest are initialized as rational larger values for erasing the affect of the last target point.

Dynamic Object Tracking Algorithm Based on PSO

Step 1 Initialization
Iitialize the number of particle
the max number of group overlapping MAX;
the position of target point O_x, O_y, O_z;
threshold value th;
the arrange of each axis of original particle;
accelerate constant C_1, C_2;

Step 2 FOR each passed point of tracked object
WHILE（the overlapping number < MAX）DO
{
IF（the overlap related to this passed point is the first overlap）
THEN { Initialize global best value *gbest* and the every personal best value *pbest*, give them bigger values, such as 10000;}
FOR every particle
{
(1) Count the fitness value (new pbest) of the new position for every particle;
If new pbest >old pbest
Then pbest=new pbest;
(2) Change gbest according to updated pbest in (1);
(3) Update the speed of every particle, and limited them in V_{max};
(4) Update the position of every particle;
}
}

Step 3 Finish the path planning and output the generated pathes.

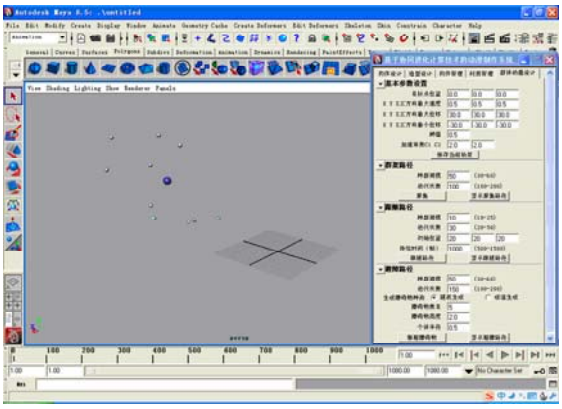

Figure 17. Initial state of dynamic object tracking

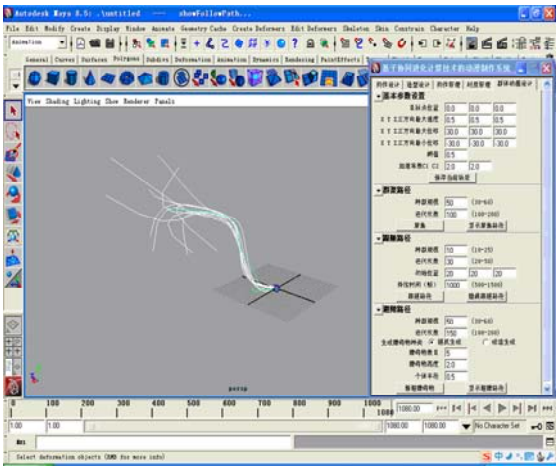

Figure 18. One generated tracking path

Figure 17 and figure 18 are simulated initial state and generative paths while figure 19 is a generated tracking image according to the generated path. If the different start position and target position are set, the different tracking points are produced and various tracking paths will be generated.

Figure 19. One genarated shark tracking image

D. Collision Avoidance

To run as a group, animals must remain in close proximity while avoiding collisions with other members of the group and with obstacles in the environment. Collisions frequently happened in a group action but we have to avoid their occurrence. For escaping collisions in path layout, after the radius and numbers of barriers are erected, two kinds of barriers are generated by randomly and grouping. These barriers locates in the area of the target point and the maximal shift of coordinate x ,y and z. Considering the cost of operations and requirement of real time, the following approach is presented for collision avoidance.

Suppose the radius of the collision object O is r, the current speed of particle i is v(vx,vy,vz) and current location is p(tx,ty,tz). The overlapping of this time can get the estimated next location p'(tx',ty',tz'). Counting the distance d between p' and the center of collision object O. If d< the distance from the center of O to the boundary of collision area，then the collision is possible happened. Therefore, the direction of particle should be changed and recount the next location.

The estimated next location p'(tx',ty',tz') is gotten by the following procedure.

Let vector $\vec{u}$ is the vector from the current position p of the particle i to the center of the collision object O and vector $\vec{k}$ is the vector from the current position p of the particle i to the estimated next position p'. The unit vector $\vec{o}$ should be coplanar with vector $\vec{u}$, $\vec{k}$ and perpendicular to $\vec{u}$. According to the right hand rule and the related rules for cross product, vector $\vec{t}$ and $\vec{m}$ are gotten.

$$\vec{t} = \vec{k} \times \vec{u} \tag{10}$$

$$\vec{m} = \vec{u} \times \vec{t} \tag{11}$$

Let $\vec{o} = \vec{m} / \left|\vec{m}\right|$ and the module of the speed

$\left|\vec{v}\right| = \sqrt{(vx^2 + vy^2 + vz^2)}$. By moving distance $\left|\vec{v}\right|$ from the unit vector $\vec{o}$, the next position p'(tx',ty',tz') is solved.

In figure 20, square frame denotes the collision object, and dashed line denotes the boundary of collision detection while triangle denotes the next position of the particle. We can see from figure 20, the next position of the particle i has entered the area of collision, and the moving direction and the next position should be changed.

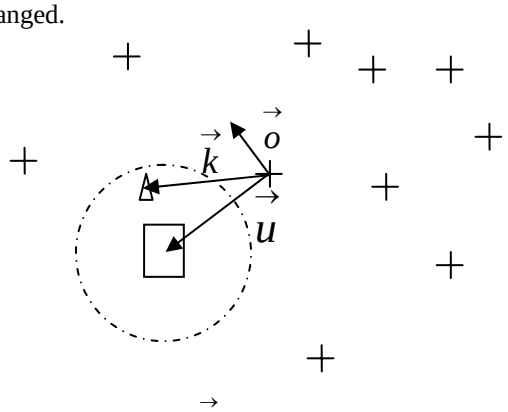

Figure 20. Unit vector $\vec{o}$ and the next position p'(tx',ty',tz')

Collision avoidance algorithm based on PSO

Step 1 Initialization
Initialize the number of particle
the max number of group overlapping MAX;
the position of target point O_x, O_y, O_z;
threshold value th;
the arrange of each axis of original particle;
accelerate constant C_1, C_2;

Step 2 WHILE（the overlapping number < MAX）DO
FOR every particle
{
(1) Count the fitness value (new pbest) of the new position for every particle;
If new pbest >old pbest
Then pbest=new pbest;
(2) Change gbest according to updated pbest in (1);
(3) Update the speed of every particle, and limited them in v_{max};
(4) Count the next position of the particle;
If (the position of the particle has entered the area of collision)
Then
Recount the next position of the particle
according to equation (10) and (11).
}

Step 3 Finish the path planning and output the generated pathes.

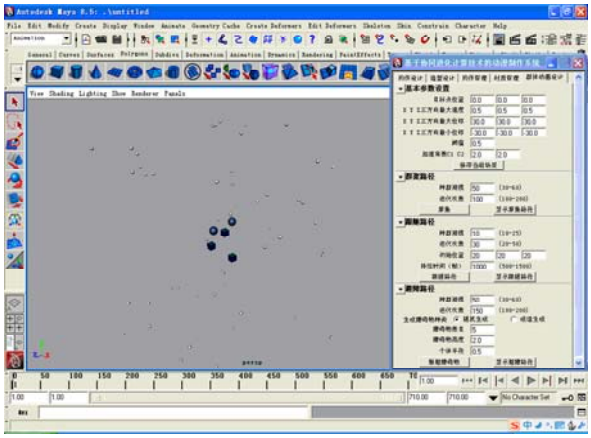

Figure 21. Initial state of collision avoidance

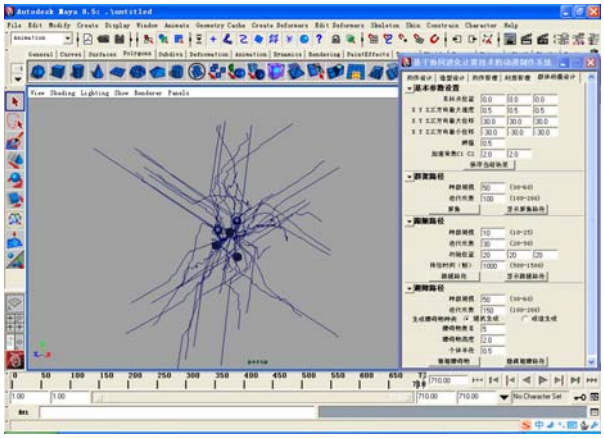

Figure 22. One generated collision avoiding path

Figure 21 and figure 22 are simulated initial state and generated paths according to collision avoidance paths. Figure 23 is a collision avoidance image according to generative path showing in figure 22. It shows that the algorithm can generate the collision avoidance paths successful.

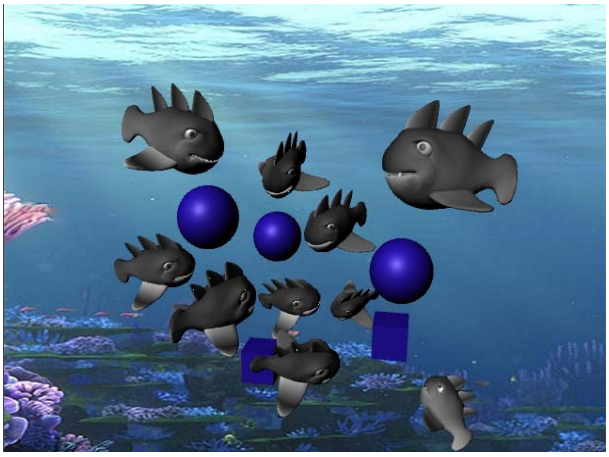

Figure 23. One collision avoidence image according to generative path

V. CONCLUSIONS

In this paper, we propose a role modeling approach based on dynamic self-adaptive genetic algorithm and a group path generating approach based on PSO algorithm for group animation. First, the backgrounds are studied. Second, a role modelling approach based on dynamic self-adaptive genetic algorithm and NURBS technology is put forward and a group of shark design example is illustrated for showing this modelling process. Third, a group path generative approach to simulates group behaviours is presented. The experiment was made in a simulation environment that is able to simulate the role modelling and path generation in animation produce process [19] . An animation named "Fancy Dress Party at the sea floor" has been made in this environment and the algorithms introduced in this paper has been practised .

Although looking simple, the approach emplores a feasible and useful way in an animation generation. It can enhance the fidelity and vitality of computer

animation, and reduce the work intensity of animators remarkably.

Future research will include: i) extending the work in this paper, to model the more complex role; ii) increasing in the system, the population characteristics for the customer classification and evaluation mechanisms for the animator's creation. These improvement will be studied in the very near future.

REFERENCES

[1] C. Reynolds, "Flocks, birds, and schools: a distributed behavioural model," *Computer Graphics*, vol.21, pp.25–34, 1987.

[2] X. Tu, D. Terzopoulos, "Artificial fishes: physics, locomotion, perception, behavior," In: *Proceedings of SIGGRAPH 1994*, pp. 43–50, NY, USA, 1994.

[3] D. C. Brogan, J. K. Hodgins, "Group behaviors for systems with significant dynamics," *Autonomous Robots*, vol. 4, pp. 137-153, 1997.

[4] S. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *Technical Report TR98-11*, Dep. of Computer Science, Iowa State University, 1998.

[5] M. G. Choi, J. Lee, S. Y. Shin. "Planning biped locomotion using motion capture data and probabilistic roadmaps," *ACM Trans. Graph.* vol. 22, pp. 182–203, 2003.

[6] R. A. Metoyer, R.A., J. K. Hodgins, "Reactive pedestrian path following from examples," *The Visual Computer*, vol. 20, pp. 635–649, 2004.

[7] Q. Zhang, J. C. Ma, W. Xie, "A framed-quadtree based on reversed d* path planning approach for intelligent mobile robot," *Journal of Computers*, vol 7, pp. 464-469,2012.

[8] D. W. Gong, J. H. Zhang, Y. Zhang, "multi-objective particle swarm optimization for robot path planning in environment with danger sources," *Journal of Computers*, vol.6, pp. 1554-1561,2011.

[9] S. Rodríguez, J. M. Lien, N. M. Amato, "A framework for planning rotation in environments with moving obstacles," In: *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, pp. 3309-3314, November 2007.

[10] J. M. Lien, S. Rodríguez, J. P. Malric, N. M. Amato, "Shepherding behaviors with multiple shepherds," In: *Proceedings of the IEEE Inter. Conf. on Robotics and Automation*, pp. 3402-3407, 2005.

[11] C. Foudil, D. Noureddine, C. Sanza, Y. Duthen, "Path finding and collision avoidance in crowd simulation," *Journal of Computing and Information Technology*, vol. 3, pp. 217–228, 2009.

[12] R. Rodrigues, M. Paravisi, AdI. Bicho, L. P. Magalhães, C. R. Jung, S. R. Musse, "An interactive model for steering behaviors of groups of characters," *Applied Artificial Intelligence*, vol.24, pp. 594–616, 2010.

[13] A. I. Bicho et. al. "Simulating crowds based on a space colonization algorithm," *Computers & Graphics*, vol.36, pp. 70-79, 2012.

[14] S. R. Musse, C. R.Jung, J.C.S. Jacques, A. Braun, "Using computer vision to simulate the motion of virtual agents.," *Computer Animation and Virtual Worlds*, vol. 18, pp.83–93, 2007.

[15] S. R. Musse, V. J. Cassol, C. R. Jung, "Towards a quantitative approach for comparing crowds," *Computer Animation and Virtual Worlds*, vol. 23, pp. 49-57, 2012.

[16] S. Gao, Z. Y. Zhang, C. G. Cao, "Particle swarm optimization algorithm for the shortest confidence interval problem," *Journal of Computer*, vol. 7, pp.1809-1816, 2012.

[17] S. Song, B. Lu, L. Kong, J. J. Cheng, "A Novel PSO Algorithm Model Based on Population Migration Strategy and its Application,", *Journal of Computer*, vol. 6, pp. 280-287, 2011.

[18] H. Liu, S. J. Xu, "Group animation path generation based on particle swarm optimisation," In: *Proc. of the 14th Int. Conf. on CSCW in Design*, pp. 37-42, Fudan University, Shanghai, China, 2010.

[19] H. Liu, H. C. Yu, Y. Y. Li, Y. L. Sun, "A role modelling approach for crowd animation in a multi-agent cooperative system," In: *Proc. of the 15th Int. Conf. on CSCW in Design*, pp. 304-310, Lausanne, Switzerland, 2011.

Hong Liu was born in 1955, is now a Professor of computer science in the School of Information Science and Engineering, Shandong Normal University. She received PhD degree from the Chinese Academy of Sciences in 1998. Her main research interests include computational intelligence and cooperative design.

Yuanyuan Li was born in 1986. She received M.S. degree from Shandong Normal University, and now is a Ph.D. student in Tongji University. Her main research interests include evolutionary algorithm and group animation.

Hanchao Yu was born in 1986. He received M.S. degree from Shandong Normal University, and now is a Ph.D. student in the Chinese Academy of Sciences. His main research interests include evolutionary algorithm and group animation.