Hierarchy and Adaptive Size Particle Swarm Optimization Algorithm for Solving Geometric Constraint Problems

Wenhui Li
College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
Email: liwh@jil.edu.cn

Mingyu Sun
College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
Email: sunmingyu370@sohu.com

Huiying Li and Bo Fu
College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
Email: {lihuiying@jlu.edu.cn, fobocloud@163.com}

Huichun Li
College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
Email: lihc@mails.jlu.edu.cn

***Abstract*—Geometric constraint problems are equivalent to a series of nonlinear equations which are constraint-meeting. Thus, it is a significant issue to improve solving efficiency of the nonlinear equations. This paper proposes Hierarchy and Adaptive Size Particle Swarm Optimization (HASPSO) algorithm for solving geometric constraint problems, and its aim is to greatly improving solving efficiency. This is the basic idea: according to individual extremum, making a comparison between each particle and its members in the direct next hierarchy, then based on transmission principle, taking the best particle's personal optimal position as its own to do subsequent iterations. Meanwhile, it depends on the natural principle of Fibonacci sequence, by simulating biological reproduction behavior, to make the algorithm adaptively expand its population size from a single individual to appropriate numbers of ones for subsequent hierarchies. If a particle still has not found precision-meeting optimal solution after a T times of iterations, then our approach judge whether the particle is new reproduced individual before the iterations, if so, it continues next T times of iterations, otherwise it produces one new individual in its direct next hierarchy, and reinitializes its position and velocity. HASPSO is able to rank population based on the form of adaptively increasing size by degrees. Theoretical Analysis and experiment show that compared with traditional particle swarm optimization (PSO) algorithm, it can make solution efficiency greatly improved and is an effective method for solving geometric constraint problems.**

Manuscript received
Nation Nature Science Foundation: No 60873147; Science and Technology Development Program of Jilin Province: No 20060527
Corresponding Author: Mingyu Sun (1978-): Male, Ph D. His major research interests include computer graphics and computer image processing.

***Index Terms*—geometric constraint solving, particle swarm optimization, hierarchy, adaptive size**

I. INTRODUCTION

Geometric constraint solving is an important research subject based on constraint solving in recent years, and it is the key to realize geometric modeling [1]. Meanwhile, this subject is also a significant part of Computer Aided Design (CAD) [2].

All solutions of geometric constraint problems are finally boiled down to optimal solutions of mathematical optimization model which is corresponding to equivalent nonlinear equations. As a branch of swarm intelligence evolution computation technology system, PSO has become one of effective way to optimal solution by adopting mathematical optimization model, which is mainly because it has strong global search capability and high local search efficiency. Meanwhile, the strong robustness and the inner parallelism are also important reasons for PSO becomes an effective optimization tool. But for complex multimodal and multidimensional optimization problems, if we again adopt traditional PSO, then, often because the population size is too small, it is easy to drop into local extremum in the later stages of iteration. Otherwise, although the algorithm has strong optimization ability, by then its convergence speed will be significantly reduced.

Chunhong Cao [3] and Hua Yuan [4] have ever put forward adopting Crossbreeding Particle Swarm Optimization Algorithm (CPSO) and Tabu Particle Swarm Optimization Algorithm (TPSO) to solve geometric constraint problems. Compared with PSO,

doi:10.4304/jsw.7.11.2567-2574

these two algorithms have been improved in solving efficiency. Cao's method lessons from the thought of genetic algorithm based on PSO, which depends on a crossbreeding mechanism to increase the individual diversity, however under the premise of diversity can't be satisfied, CPSO usually makes the solving process of problems produce temporary stagnation till the diversity is satisfied by crossbreeding, therefore, it reduces the solving efficiency of problems. Yuan's method combined PSO with tabu search (TS), which, in the solving process, adopts PSO in the early stages and tabu search in the later stages. According to a particular problem, if PSO is able to lock the neighborhood of global optimal solution, then, through subsequent tabu search, the solution will be satisfactory. But TS's solution quality often depends on the selection of some initial values, if given initial value is not a point in the neighborhood of the global optimal solution, then, after employing TS, probably there still appear circulation search due to the tabu length is too short, or fell into local extremum due to the tabu list is too small, which, therefore, makes TPSO stagnate.

This paper, on the basis of PSO and based on the principle of ranking population and adaptively increasing size, put forwards Hierarchy and Adaptive Scale Particle Swarm Optimization Algorithm (HASPSO), its purpose is to improve solving efficiency of complex geometric constraint problems which consist of multivariable constraint relations.

II. GEOMETRIC CONSTRAINT SOLVING

A. Conception

Geometric constraint solving can be understood as automated geometry construction [5], it is a process of deciding location relationships for given geometric elements. Meanwhile, geometric constraint describes semantic features of specific relationships between geometric elements, so the connotation of solving process is a kind of interpretation behavior of semantic features.

Generally geometric constraint problems are usually equivalent to a series of nonlinear equations which are constraint-meeting, and the corresponding nonlinear equations can be expressed as follows:

$$\begin{cases} f_1(x_1, x_2, \ldots\ldots, x_D) = 0 \\ f_2(x_1, x_2, \ldots\ldots, x_D) = 0 \\ \quad \ldots\ldots \quad \ldots\ldots \\ f_m(x_1, x_2, \ldots\ldots, x_D) = 0 \end{cases} \tag{1}$$

We make solving vector $X = (x_1, x_2, \ldots\ldots, x_D)$, here $x_i (i = 1, 2, \ldots\ldots, D)$ are some unanswered parameters. Thereby constraint solving can be understood as getting a specific X through calculating, and making the X meet (1).

Design problem in essence is one about constraints satisfaction [6], what is more, if (1) has been provided, then we can divide geometric constraint problems into three categories as well-constrained, under-constrained and over-constrained.

Generally, we first need to consider the category which problems belong to according to specific situation, and then do the subsequent solving work.

B. Optimization Solution

For geometric constraint problems, at present there have not yet fully effective and general solving schemes and the existing all kinds of solving schemes have different forms [7]. Its optimization solution is generally at the established levels and based on some given constraint conditions to select the best solution set. However, in practice, some optimization problems are very complex, so we are usually just according to need to work out the solution set which meets given precision. On the basis of the equivalent nonlinear equations to which geometric constraint problems are corresponding, we can structure the following objective function:

$$F(X) = \sum_{i=1}^{m} | f_i(X) | \tag{2}$$

If there exists a vector X, and it meets the equation $F(X) = 0$, so the X meets (1). Therefore the solutions of geometric constraint problems can finally be transformed into that of optimization problems, the equivalent mathematical optimization model can be expressed as follows:

$$MinF(X) = 0 \tag{3}$$

What is more, the termination condition of calculating has to meet the follows:

$$F(X) < \varepsilon \tag{4}$$

Here we take the ε as a precision-meeting threshold, and it is able to make a judgment to whether the optimization routine needs to be terminated. Meanwhile, according to the mathematical optimization model that (1) represents, if we take the principle of transforming equivalent nonlinear equations into a mathematical optimization model, then it can be in the conditions of not having to consider the above-mentioned three constraint categories, to solve conveniently.

III. THE BASIC PSO ALGORITHM

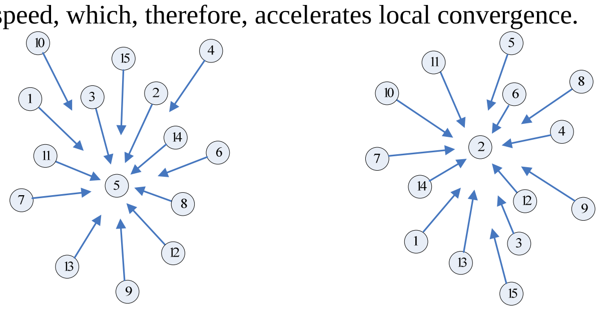
Particle Swarm Optimization (PSO) Algorithm is most early proposed by Kennedy and Eberhart in 1995, who was inspired by research results of artificial life, simulated birds' migration and swarm behaviors in the process of foraging, it is a kind of evolution calculating technology based on swarm intelligence[8]. Because PSO's structure is simple and easy to realize, the method, as in genetic algorithm, has been widely applied to different kinds of more complex optimization problems' solutions [9].

A. Basic Principle

PSO mainly spreads some particles which have optimal characteristics in D-dimensional space and through these particles' search behaviors based on cooperation, to realize problems' solutions. It is able to simulate some animals' population characteristic, hence, there doesn't exist any one complete central director [10]. In the meaning time, the more abundant population diversity is, the greater PSO's ability to deal with a complex problem is [11]. And because PSO has excellent memory characteristic, a particle also has a certain cognitive performance [12].

Generally, a particle can be regarded as a flying individual of no have mass and volume but simple intelligent behavior. And every individual is taken to be a potential solution for particular problems [13]. As flying experience, every individual needs to record the best position through which it has passed itself, and the position is called personal optimal position.., its corresponding fitness value is called individual extremum P_{best}. Besides, to obtain flying experience of the particles within neighborhood, PSO also records the best position through which the whole particle swarm has passed, this position is called global best position Gb, and its fitness value is called global extremum G_{best}.

Meanwhile, every particle has the ability to learn from the best one of population, it is based on its own or neighborhood particles' flying experience to adjust flying speed, which, therefore, accelerates local convergence.



(a) Individual 5 is the best one (b) Individual 2 is the best one

Figure 1. Basic PSO pattern

B. The Mathematical Model Of PSO

According to basic principles, converting an algorithm into its corresponding mathematical model is the key of solving practical problems. PSO can be regarded as a king of iterative process about self-organized learning. We assume that in D-dimensional solution space, there is a population $S=\{X_1,X_2,......,X_n\}$ which consists of n particles representing a problem's potential solutions. So the mathematical model that every particle follows can be formulated as:

$$V_i^{k+1} = wV_i^k + c_1 r_1 (Pb_i^k - X_i^k) + c_2 r_2 (Gb^k - X_i^k) \quad (5)$$

$$X_i^{k+1} = X_i^k + V_i^k \quad (6)$$

In these two equations, $V_i^k = (v_i^{k1}, v_i^{k2}, ..., v_i^{kd})$ is the ith particle's velocity after k times of iterations, and $X_i^k = (x_i^{k1}, x_i^{k2}, ..., x_i^{kd})$ is after k times of iterations, the particle's position, but $Pb_i^k = (pb_i^{k1}, pb_i^{k2}, ..., pb_i^{kD})$ and $Gb^k = (gb^{k1}, gb^{k2}, ..., gb^{kD})$ are respectively its personal optimal position and global best position, w is as a inertia weight to maintain an individual's movement state, usually $w \subset [0.9,1.2]$ [14], c_1 and c_2 are particles' nonnegative learning factors, normally $c_1 = c_2 = 2$, r_1 and r_2 are certain random number that are read on the interval $[0,1]$.

According to specific problems we have to limit every particle's velocity and position to a range allowed, so $v^j \in [v_{\min}^j, v_{\max}^j]$, $x^j \in [x_{\min}^j, x_{\max}^j]$, $(j = 1, 2, ..., D)$. The purpose is to guarantee particles can find the optimal solution meeting precision in given feasible domain.

Equation (5) is composed of three pieces, and by PSO's principle, they can be described as follows: the first piece represents a particle's ability to maintain its original movement state, it is particle movement's inertia index and mainly plays a role of balancing global and local search. The second piece is particles' self-recognition, which indicates a particle's subsequent behavior also comes from its own existing experience, consequently, makes particles have strong enough global search ability and avoids local extremum at all possible. But the third piece is particles' social cognition; it reflects collaboration and knowledge sharing between them.

IV. The HASPSO Algorithm

When we apply PSO to solve complex multimodal and multidimensional optimization problems, its fast local convergence speed and strong global search capability often cannot be considered all together. On the one hand, the transmission of information between particles is too fast, thus, it will reduce the diversity of individuals, make population trapping the local extremum, and bring premature convergence, on the other hand, if the transmission is too slow, then every individual will be difficult to obtain information of the others, which are far apart from it, in its neighborhood, it will reduce convergence speed, consequently, the algorithm is hard to improve the problems' solving efficiency.

Thereby, following the above shortcomings, this paper proposes Hierarchy and Adaptive Size Particle Swarm Optimization Algorithm (HASPSO) to realize the solution of geometric constraint problems.

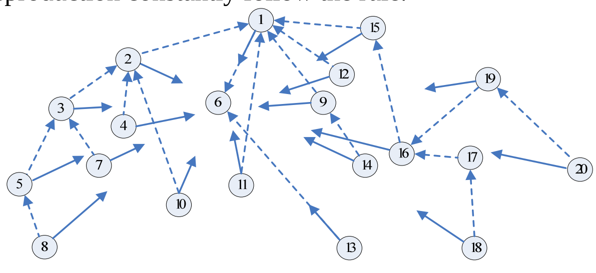
A. Elementary Principles

The realization of HASPSO algorithm mainly follows the following two rules:

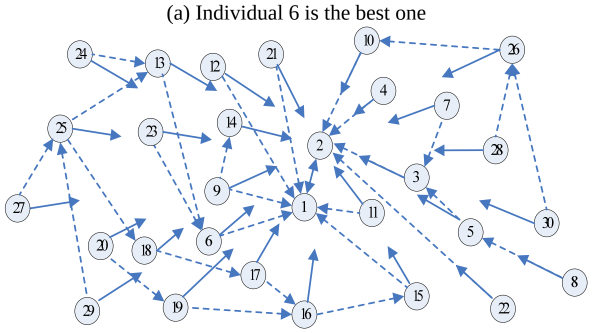
(1) Hierarchies: HASPSO has the ability to rank population. This rule makes every particle compare its individual extremum with that of all others, which are directly subject to the particle itself, in the next hierarchy, and including the particle itself, selects the best personal optimal position to which the best individual extremum is corresponding. And based on transmission principle, the algorithm uses the best personal optimal position that selected as the particle's new one to do subsequent iterations. The aim of using ranking mechanism is making the particles within higher hierarchies have better quality of solutions in a relatively short amount of time; therefore, it can accelerate the algorithm's convergence.

(2) Size adaptation: Maintaining the diversity of individuals steadily and harmoniously is an important factor of the algorithm can converge to global optimal solution. According to the natural principals of Fibonacci sequence can reflect harmony and stability, through simulating the behaviors of biological growth and reproduction in nature, HASPSO gradually expands population size of solving, which depends on complexity of specific problems, from the hierarchy where initial single individual is to subsequent hierarchies. If an individual has still not found optimal solution of meeting given precision after iterations T times, then our approach first judge whether the particle is a new reproduced individual before T times of iterations, if so, it continues next T times of iterations; otherwise, it produces one new individual in the next hierarchy to which it belongs directly, and reinitializes its position and velocity. The purpose of adaptively expanding population size base on the rule of Fibonacci sequence is that under the premise of adopting the pattern of hierarchy to ensure faster convergence, because it corresponds with the harmony and stability of biological growth and reproduction in nature, the rule can stably and harmoniously maintain diversity of individuals, therefore it makes the population escape from a local extremum by greatest possibility and increase individuals' global search performance.

The basic principle of HASPSO is show in Fig. 2, (a) is the situation of adaptively expanding individuals from 1 to 20 based on the pattern of hierarchy and according to the rule of Fibonacci sequence; (b) is, under the premise of (a), the situation of adaptively expanding individuals from 21 to 30 after iterations. The dotted lines denote the hierarchy of population and their arrowheads directly point to the previous hierarchy, but the solid lines indicate a kind of converging behavior of population follow a best individual at a particular moment. Because the algorithm itself is based on concurrency mechanism, in the process of solving every time, the order of reproducing individuals is usually different, but in macroscopic aspect, the behaviors of population's reproduction constantly follow the rule.



(a) Individual 6 is the best one



(b) Individual 2 is the best one

Figure 2. HASPSO pattern

B. The Mathematical Model Of HASPSO

In HASPSO, the change of personal optimal position Pb and individual extremum P_{best} is not only from a single individual itself any more, but mainly from the best one which directly belongs to its next hierarchy, also including the individual. By this time, we sign the personal optimal position as Pbt, correspondently, the individual extremum is signed as P_{best_trans}, and so the mathematical model of every particle's velocity iteration can be expressed as follows:

$$V_i^{k+1} = wV_i^k + c_1 r_1 (Pbt_i^k - X_i^k) + c_2 r_2 (Gb^k - X_i^k) \quad (7)$$

In (7), Pbt_i^k is the personal optimal position based on the scheme of hierarchy and adaptive size, the other parameters have the same meanings as that of (5). And also, the equation's first piece and its third piece have the same above-mentioned connotations as that of (5). But the second piece is rightly based on hierarchy, a particle's self-cognition; meanwhile, it can be further described as: with the help of all the other individuals which directly belong to an individual's next hierarchy, it is a rapid promotion of self-cognitive ability. And it is also a transmission effect of an individual is able to acquire more knowledge, through which, HASPSO makes individuals within higher hierarchies have better self-cognitive ability in a relatively short amount of time. Thereby under the premise of ensuring particles have strong enough global search ability and avoid local extremum as much as possible, the transmission effect can improve the algorithm's convergence rate. Although, like PSO, there doesn't exist any complete center controllers in HASPSO, the hierarchal mechanism greatly enhances the synergy between individuals.

In the meaning time, HASPSO's population size n usually depends on the complexity of a specific problem, and in the process of solving the problem; it is increasing by degrees according to the following equation:

$$a_N = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^N + \left(\frac{1-\sqrt{5}}{2}\right)^N\right] \quad (8)$$

Equation (8) is calculation formula of the Nth item in Fibonacci sequence, consequently, as a whole, every T times of iterations the population experiences, the population size always follows the series' rule as $n_1 = 1$, $n_2 = 1$, $n_3 = 2$, $n_4 = 3$, $n_5 = 5$, $n_6 = 8$, ……, to be established, and obviously it is based the series, a obtained result of constantly reproducing new individuals to the subsequent hierarchies.

The only criterion of evaluating whether solution vector meets given precision is objective function's fitness value. For geometric constraint problems, the corresponding objective function is $F(X)$ of (2); its negative function value may serve as the fitness criterion of problem solution, which is able to make a judgment for accepting or rejecting a solution vector.

In HASPSO, the iteration model of particles' position is the same as (6). We assume that the given precision of optimization model, which is corresponding to a specific geometric constraint problem, is ε, the ith individual's fitness value is $fitness(X_i)$, its current position, personal optimal position and personal extremum are respectively Pc_i, Pb_i and P_{best-i}, and its current iteration times is t_{cur-i}, the maximum iteration times of a cycle is T. So every individual's solving process in HASPSO can be described as follows:

Step 1: Initialize the ith particle P_i 's velocity V_i, position X_i, personal optimal position Pbt_i.

Step 2: According to the given problem, calculate the fitness value based on P_i 's current position.

If $fitness(Pc_i) \le \varepsilon$ **Then**
Terminate the iteration and output Pc_i
Endif

Step 3: Based on fitness value, compare P_i 's current position with its personal optimal position.

If $fitness(Pc_i) < fitness(Pb_i)$ **Then**
$Pb_i = Pc_i$
Endif

Step 4: Based on personal extremum, compare P_i 's personal optimal position with that of the particle P_j belongs to its previous hierarchy.

If $P_{best-i} < P_{best-j}$ **Then**
$P_{best-j} = P_{best-i}$
Endif

Step 5: Based on fitness value, compare P_i 's current position with the population's global best position.

If $fitness(Pc_i) < fitness(Gb)$ **Then**
$Gb = Pc_i$
Endif

Step 6: Judge whether P_i 's iteration has reached the set maximum times of iteration T in a cycle.

If $t_{cur-i} = T$ **Then**
goto Step 7
Else
goto Step 11
Endif

Step 7: Judge whether P_i is a new reproduced one before the T times iterations in a cycle.

If P_i is not a new reproduced one **Then**
goto Step 8
Else
goto Step 9
Endif

Step 8: In the direct next hierarchy of P_i, reproduce a new individual.

Step 9: Reinitialize P_i 's position and its velocity.

Step 10: Set the individual's iteration times to zero, so make it carry through the next cyclic iterations.

Step 11: Update P_i 's current position and its velocity according to (7) and (6).

Step 12: Increase P_i 's current iteration times t_{cur-i} by one in a cycle.

$t_{cur-i} = t_{cur-i} + 1$
goto Step 2

C. The Algorithm Analysis Of HASPSO

To illustrate HASPSO's efficiency is higher than that of PSO, we make the following theoretical analysis:

Assume that there exist a specific geometric constraint problem, and its corresponding nonlinear equations have been turned into a certain optimization model as (3). Meanwhile, assume that the optimal solution meeting given precision is $X_{best} = (x_b^1, x_b^2, ..., x_b^D)$, and at a particular moment, the size of particle swarm is fixed at n, whose purpose is mainly to prove HASPSO of adopting the hierarchy pattern has a faster convergence rate than PSO. Also, we assume that after the ith individual has made k times of iterations, its current position is $X_i^k = (x_i^{k1}, x_i^{k2}, ..., x_i^{kD})$, here $i = 1, 2, ..., n$. If adopt PSO and given the ith individual's personal optimal position is $Pb_i^k = (pb_i^{k1}, pb_i^{k2}, ..., pb_i^{kD})$, well, at the moment, the distance from Pb_i^k to X_{best} is:

$$d_{1i} = \sqrt{\sum_{q=1}^{D} (x_b^q - pb_i^{kq})^2} \tag{9}$$

If we adopt HASPSO to rank the population, so the population's structure is regarded as a constantly growing multi-branches tree logically, and every individual is corresponding to one node. Besides the m individuals which are corresponding to the leaf nodes, the rest of $n-m$ individuals have an opportunity of obtaining personal optimal positions which are from their direct next hierarchies. Therefore, relative to PSO, under the same condition, the corresponding ith individual of HASPSO with that of PSO can be described as follows:

(1) If it is a leaf node of the multi-branches tree, then at this moment, the distance from its personal optimal position Pb_i to the optimal solution X_{best} meeting given precision is d_{1i} of (9).

(2) If it is not a leaf node in this tree, so we can reasonably assume that among the q individuals of its direct next hierarchy, there must exists a best personal optimal position $Pb_i^l = (pb_i^{l1}, pb_i^{l2}, ..., pb_i^{lD})$, here l is iteration times of the individual corresponding to the position. At the same time, Pb_i^l is better than the ith individual's personal optimal position Pb_i^k, therefore, according the principle of HASPSO, we can use Pb_i^l as the ith individual's new personal optimal position, and through this measure, the distance from Pb_i^l to X_{best} is:

$$d_{2i} = \sqrt{\sum_{q=1}^{D} (x_b^q - pb_i^{lq})^2} \tag{10}$$

Because Pb_i^l is better than Pb_i^k, thereby it can be realized that $d_{2i} \le d_{1i}$. For PSO, after $k+1$ times of iterations, the distance from the ith individual to X_{best} is:

$$d_{1i}^{'} = \sqrt{\sum_{q=1}^{D}[x_i^{kq} + wv_i^{kq} + c_1 r_1 (pb_i^{kq} - x_i^{kq}) + c_2 r_2 (gb^{kq} - x_i^{kq}) - x_b^q]^2} \quad (11)$$

But for HASPSO, after $k+1$ times of iterations, the distance from the ith individual to X_{best} turns into:

$$d_{2i}^{'} = \sqrt{\sum_{q=1}^{D}[x_i^{kq} + wv_i^{kq} + c_1 r_1 (pb_i^{lq} - x_i^{kq}) + c_2 r_2 (gb^{kq} - x_i^{kq}) - x_b^q]^2} \quad (12)$$

Also, $d_{2i} \le d_{1i}$, so $d_{2i}^{'} \le d_{1i}^{'}$. We can learn that if adopt PSO, then, based on concurrency mechanism, at any time of after every individual has experienced different times of iterations respectively, the average distance from every individual to X_{best} is:

$$\overline{d_1} = \frac{\sum_{i=1}^{n} d_{1i}^{'}}{n} \quad (13)$$

If we use HASPSO, then at this moment, under the premise of fixed population size, the average distance from every individual to X_{best} is:

$$\overline{d_2} = \frac{\sum_{i=1}^{n} d_{2i}^{'}}{n} \quad (14)$$

Accordingly, $\overline{d_2} \le \overline{d_1}$. Thereby we know that compared with PSO, under the same conditions, if there exist transmission behaviors of particles' personal optimal position, then in the optimizing process of every time of iteration, adopting hierarchy pattern can accelerate the convergence of entire population obtains optimal solution.

In PSO, the third piece of velocity iteration equation, which is regarded as a social cognitive system, is able to reflect the cooperation and knowledge sharing between individuals, and indicate that every individual has the ability to learn from the best one in population. Here we firstly neglect the first and second piece, and examine any two individuals X_i and X_j, $i, j \le n$, so when they are initialized randomly, the distance from X_i to X_j is:

$$d_{i,j}^0 = \sqrt{\sum_{q=1}^{D}(x_i^{0q} - x_j^{0q})^2} \quad (15)$$

Label 0 represents the state of the two individuals still has not begun to iterate. So at this time the velocity iteration equation is evolved into as follows:

$$V_i^{k+1} = c_2 r_2 (Gb^k - X_i^k) \quad (16)$$

And in the process of iterations many times, without loss of generality, on average, we can approximately and legitimately consider that X_i and X_j have the same $c_2 r_2$, so after k times of iterations, the distance between them turns into as follows:

$$d_{i,j}^k = \sqrt{\sum_{q=1}^{D}[(x_i^{0q} - x_j^{0q})\prod_{p=1}^{k}(1 - c_2 r_2)]^2} \quad (17)$$

In (17), $0 \le c_2 r_2 \le 2$ when $c_2 = 2$, therefore, we can get the result as follows:

$$d_{i,j}^k \le d_{i,j}^0 \quad (18)$$

The above result tells us after k times of iterations, the distance from X_i to X_j is reduced. Here we give the definition of individual similarity.

Definition: The similarity of individual X_i and X_j is defined as follows:

$$Si = e^{-d_{i,j}} \quad (19)$$

In (19), e is the base of natural logarithm, $d_{i,j}$ is the distance from X_i to X_j.

At this moment, if we reconsider the first and second piece of velocity iteration equation, then because the third piece also participate in the iterative behaviors, finally, on the whole, it will reduce the distance between individuals, and gather the population to one point or its very small neighborhood. This means the similarity of individuals has been increased, which causes the principle of individual diversity is decreased, so it easily makes the search drop into local extremum or appear stagnation.

When adopting PSO, the more abundant population diversity is, the greater its ability to solve complex problems is, but at this moment, the learning between individuals usually has a certain blind obedience to nature, which makes the similarity increased and it is one of the major reasons for gradually decreasing the principle of population diversity.

HASPSO in this paper, after an individual's every T times of iterations, it reproduces a new one in its direct next hierarchy. In which the iteration times T can be set different values according to a specific problem. Therefore, to a larger extent, HASPSO is able to increase the distance between the new reproduced individuals and the original ones, which means it reduces the similarity between individuals and blind obedience of learning, and in the search process, it also improves the principle of population diversity to a larger extent, for avoiding dropping into a local extremum or stagnation. Meanwhile, at individual behavior level, the mechanism above-mentioned can be explained as follows: assume that there has existed an individual to be produced, and the individual first observes the search status of the one which belongs to its direct previous hierarchy, and according to the search status, decides whether to participate in the optimization. So, at this moment, HASPSO has a certain ability to observe and judge.

Generally, Fibonacci sequence can truly reflect the stability and harmony of a class of things in the nature, which are in the process of their growing and reproducing, what is more, through based on the particular property that the series has, it can reduce the conflicts between things as much as it does.

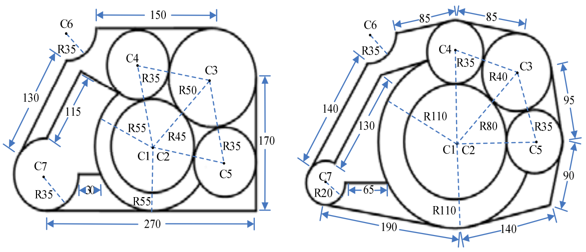
Therefore, the series is also called Golden Mean sequence. As a mathematical ratio relation, normally, there exists the golden ratio $\varphi = 1.6180339887498948482......$, which is an unlimited circulating decimal. As $N \to \infty$, the ratio of the series' every term and the one directly in front of which is as follows:

$$\lim_{N \to \infty} a_{N+1} / a_N = \varphi \tag{20}$$

Because the population size only can be rounded in number, if according to the principle of Fibonacci sequence, HASPSO adaptively increases the size, then its ratio of population size adjacent two generations is the nearest to golden ration after every T times of iterations, in the meaning time, the principle also is most able to have the stability and harmony of increasing population size, accordingly it makes HASPSO most have the stability and harmony of maintaining individual diversity.

V. Solving Experiment

To verify HASPSO's effectiveness of solving geometric constraint problems, herewith, in this paper we give an example associated with this type of problem. Fig. 3 shows two complex geometric shapes which consist of some basic geometric elements meeting corresponding constraint relation, in which, (a) is the original design draft, but (b) is according to (a), under the premise of changing the geometric elements' size and being still able to meet the constraint relations, the changed design pattern respectively adopting PSO, CPSO, TPSO, HASPSO to solve it.



(a) The original design draft (b) The changed design pattern

Figure 3. HASPSO's Solving Example of Geometric Constraint Problems

Meanwhile, for the given solving example by Fig. 3, under the premise of meeting identical given precision, setting exactly same parameter values and equipping with identical software and hardware configuration, we, on the basis of time efficiency, repeatedly solve 100 times to compare HASPSO with PSO, CPSO and TPSO. Furthermore, the solution precision of corresponding optimization model is 1×10^{-9}, the individuals' parameter setting is $w = 0.9$, $c_1 = c_2 = 2$, the programming environment of aforementioned these algorithm is Microsoft Visual Studio 2005, their running environment is Microsoft Windows XP Professional, and the hardware configuration is DELL Corporation Intel (R) Core (TM)2 Duo CPU E7200 @ 2.53GHz, 1GB of RAM. As we adopt HASPSO, its every individual's iteration times in a cycle are set for 500 when a new one is produced in its direct next hierarchy. And the population size of original PSO, CPSO and TPSO is specified as 30, this is because PSO's population size is generally between 10 and 40, Carlisle et al compared different sizes and suggested that 30 is a reasonable choice [15]. So, that of CPSO and TPSO is also 30. Their comparison results are listed in Table 1, in which, the obtained average, best and worst solving time are all other statistical results but stagnation phenomenon.

TABLE I.
The Performance Comparison of HASPSO, PSO, CPSO and TPSO

Algorithm Name	Average Time (s)	The Best Time (s)	The Worst Time (s)	Number of Stagnation
PSO	58.245	27.031	91.364	82
CPSO	90.246	14.362	117.725	none
TPSO	46.263	21.273	74.214	74
HASPSO	3.943	3.459	4.265	none

It can be seen when we adopt HASPSO to realize the solutions of complex and multidimensional geometric constraint problems, which has the efficiency difficult to achieve for PSO, CPSO and TPSO. When we adopt PSO, because it is difficult to guarantee the individual diversity, the algorithm usually causes more stagnation phenomenon. And when we adopt CPSO, however, the worst solving time is lengthened, the main reason is that in relatively short period of time, under the premise of difficultly guaranteeing individual diversity, it appears temporary stagnation phenomenon till the crossbreeding mechanism makes individual diversity be satisfied as mentioned earlier, which, thereby, lengthens the average solving time. As we adopt TPSO, because it combines the tabu search strategy, compared with PSO, its convergence velocity has been improved, but the strategy cannot make TPSO escape from a local extremum and individual diversity is still not ensured, therefore, it also usually appears more stagnation phenomenon.

VI. Conclusion

Hierarchy and Adaptive Size Particle Swarm Optimization is able to rank its population, and on the basis of transmission principle, makes individuals in higher hierarchies have better self-cognitive ability, which, therefore, can further accelerate its convergence. And HASPSO is based on the principle of Fibonacci sequence, through simulating biological growth and reproduction behaviors, to adaptively increase population size. The mechanism guarantees the stability and harmony of individual diversity, which makes it can avoid stagnation phenomenon and escape from local minima by greatest possibility. The theory analysis and experimental results show that adopting HASPSO to solve geometric constraint problems can overcome the shortcoming of such PSO and its comprehensive performance is better than PSO to a large extent.

The HASPSO that this paper put forwards, , when a particle produces new individuals, its iteration times are fixed in a cycle. In other words, this particle is difficult to effectively judge its optimal reproduction time. Therefore, according to a problem and the corresponding solving process, making an effect judgment for a particle's optimal reproduction time, which can further maintain individual diversity, is a key subject that we are going to research in future.

REFERENCES

[1] Michelucci Dominique, and Foufou Sebti, "Geometric constraint solving: The witness configuration method," CAD Computer Aided Design, 2006, vol. 38, pp. 284-299.

[2] E. B. Simon Thierry, "A particle-spring approach to geometric constraints solving," 26th Annual ACM Symposium on Applied Computing, SAC 2011. Association for Computing Machinery, 1515 Broadway, 17th Floor, New York, NY 10036-5701, United States, 2011, pp. 1100-1105.

[3] Chunhong Cao, Yongjian Zhang, and Wenhui Li, "The application of crossbreeding particle swarm optimizer in the engineering geometric constraint solving," Chinese Journal of Scientific Instrument, 2004, vol. 25, pp. 397-400.

[4] Hua Yuan, Wenhui Li, and Na Li, "Tabu-PSO algorithm for solving geometric constraint problem, " Microelectronics & Computer, 2010, vol. 27, pp. 26-29.

[5] Xiaoshan Gao, and Kun Jiang, "Survey on geometric constraint solving", Journal of Computer-Aided Design & Computer Graphics, 2004, vol. 16, pp. 385-396.

[6] Chunhong Cao, Limin Wang, and Dazhe Zhao, "The research based on the discrete cellular ant algorithm in the geometric constraint solving", Tien Tzu Hsueh Pao/Acta Electronica Sinica, 2011, vol. 39, pp. 1127-1130.

[7] Mathis Pascal, and E. B. Simon Thierry, "A formalization of geometric constraint systems and their decomposition", Formal Aspects of Computing, 2010, vol. 22, pp. 129-151.

[8] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", proc. In: IEEE Service Center ed. IEEE International Conference on Neural Networks, Perth, Australia, 1995. Piscataway: IEEE Press, 1995, pp. 1942-1948.

[9] Sangwook Lee, Haesun Park, and Moongu Jeon, "Binary particle swarm optimization with bit change mutation, " IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2007, vol. E90-A, pp. 2253-2256.

[10] Dian Palupi Rini, Siti Mariyam Shamsuddin, and Siti Sophiyati Yuhaniz, "Particle Swarm Optimization: Technique, System and Challenges," International Journal of Computer Applications, 2011, vol. 14, pp. 19-27.

[11] Daneshyari Moayed, and G. Yen Gary, "Cultural-based multiobjective particle swarm optimization," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, vol. 41, pp. 553-567.

[12] Yi Wan, Chunhong Cao, and Changsheng Zhang, "The geometric constraint solving based on hybrid differential evolution and particle swarm optimization algorithm, " 2010 International Conference on Intelligent Control and Information Processing, ICICIP 2010. IEEE Computer Society, 445 Hoes Lane-P.O.Box 1331, Piscataway, NJ 08855-1331, United States, 2010, vol. 1, pp. 692-695.

[13] Chunhong Cao, Changsheng Zhang, and Limin Wang, "An improved particle swarm optimization algorithm for geometric constraint solving problem," 2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010. IEEE Computer Society, 445 Hoes Lane-P. O. Box 1331, Piscataway, NJ 08855-1331, United States, 2010, vol. 4, pp. 1835-1838.

[14] Shi Yuhui, and Eberhart Russell, "Modified particle swarm optimizer," Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, ICEC'98, IEEE, Piscataway, NJ, United States, 1998, vol. 4, pp. 69-73.

[15] A. Carlisle, and G. Dozier, "An off-the-shelf PSO," Proceedings of the workshop on particle swarm optimization, 2001, vol. 1, pp. 1-6.

Wenhui Li was born in 1961. He is a Professor and doctor supervisor in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His major research interests include computer vision, image processing and computer graphics.

Mingyu Sun was born in 1978. Since 2010, He has been working on the PhD degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His major research interests include computer graphics and computer image processing.

Huiying Li was born in 1978. she is a Lecturer of Jilin University, China. She received the PhD degree in Computer Science and Technology from Jilin University, China, in 2008 respectively. Her research interests include computer algorithms and simulation, image processing and 3D modelling.

Bo Fu was born in 1983. Since 2009, He has been working on the PhD degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His research interests include computer image processing and computer graphics.

Huichun Li was born in 1987. Since 2009, He has been working on the M.E. degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. Her research interests include digital image processing and virtual reality technique.