The Application of the Genetic algorithm-Ant algorithm in the Geometric Constraint SatisfactionGuidelines

Cao Chunhong[1], Zhang Bin[1], Wang Limin[2], Li Wenhui[2]
[1]Information Science and Engineering, Northeastern University, Shenyang 110004, P.R.China
[2]Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
Chunhongcao_li@163.com, zhangbin@mail.neu.edu.cn, Jeffreywlm@sina.com, liwh@public.cc.jl.cn

Abstract

The constraint problem can be be transformed to an optimization problem. We introduce GAAA (genetic algorithm-ant algorithm) in solving geometric constraint problems. We adopt genetic algorithm in the former process of algorithm so that it can make use of the fastness, randomicity and global stringency of genetic algorithm. Its result is to produce the initiatory distribution of information elements. The latter process of the algorithm we adopt ant algorithm. In the condition that there are some initiatory information elements, we can utilize fully the parallel, feedback and the high solving efficiency. Using random colony in the genetic algorithm, this can not only improve the speed of ant algorithm but also avoid getting in the local best solution when solving the precise solutions. The algorithm has a good effect in not only optimization capability but also time capability.Geometric constraint problem is equivalent to the problem of solving a set of nonlinear equations substantially.

1. Introduction

Geometric constraint solving approaches are made of three approaches: algebraic-based solving approach, rule-based solving approach and graph-based solving approach. One constraint describes a relation that should be satisfied. Once a user defines a series of relations, the system will satisfy the constraints by selecting proper states after the parameters are modified.

The constraint problem can be transformed to an optimization problem. So we can solve the optimization problem by GAAA. The genetic algorithm has the capability of fast and stochastic global searching, but for the utilization of feedback information in the system it is of no effect. When solving reaches to a certain range, it will probably do much useless iteration and the efficiency of getting accurate solutions is low. Ant Algorithm can have constringency on the best route by the accumulating and updating of information elements and has the capability of distributed parallel global searching. So in this paper we combine the ant algorithm with genetic algorithm[1]. The combination of ant algorithm with genetic algorithm is named as GAAA (genetic algorithm-ant algorithm). Its basic idea is to adopt genetic algorithm in the former process of algorithm so that it can make use of the fastness, randomicity and global stringency of genetic algorithm. Its result is to produce the initiatory distribution of information elements. The latter process of the algorithm we adopt ant algorithm. In the condition that there are some initiatory information elements, we can utilize fully the parallel, feedback and the high solving efficiency.

2. Ant Algorithm

2.1 Generation of Ant Algorithm

Swarm Intelligence, namely SI, is a kind of technology of AI [2], which mainly discusses collective activities of swarm composed of many simple individuals.

With the development of bionics in the recent years, scientists take more attention to ants that seemed negligible. In 1991[3], M.Dorigo first brought forward the Ant Algorithm, and then people started the research of ant swarm: how the individuals that is relatively weak and not strong functionally can complete complex work, such as how to search for the best path to acquiring food and returning. Based on the study above on, an efficient kind of evolutional algorithm is developed gradually.

2.2 Principle of the Ant Algorithm

Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06)
Y.Y. Yao, Z.Z. Shi, Y. Wang, and W. Kinsner (Eds.)
1-4244-0475-4/06/$20.00 ©2006 IEEE

As M.Dorigo pointed out in the first article about the Ant Algorithm, the most feature of ant swarm algorithm is the indirect asynchronous contact mode of ants in swarm by the media of strigmergy[4,5]. Ants will leave some chemical matter called strigmergy in the place where they pass when they are searching for food or on the way of returning. The matter can be felt by the after ants in the same swarm, and affect the action of the latter as a signal. Furthermore, the strigmergy that the latter ant leaves can strengthen the intrinsic strigmergy, and the cycle is made in this way. Thus, the more the ants pass the path, the higher probability that the subsequent select the same way is, because of the remaining higher concentration of strigmergy. The procedure will persist until all the ants go the shortest path.[4,6,7,8]

2.3 The model of Ant algorithm

The procedure of ants seeking food is similar with the problem of traveling merchant, so the basic model of Ant algorithm can be illustrated by seeking solution of TSP of n cities.

Firstly, supposing the distance between city i and j is d_{ij} . The parameter m is the ant number in the ant colony, b_i（t）is the ant number standing in the city I in the t time, then we can get m=$\sum_{i=1}^{n} b_i(t)$. $\tau_{ij}(t)$ is the number of the information elements in the t time. At the beginning the number of information elements is each arc is equal, $\tau_{ij}(t)$=C, C is a constant. In the process of movement, ant determines its direction of removal according to the number of information elements in every arc. p_{ij}^{k}（t）is the probability that ant k moves from point i to point j in the t time.

Here J_k（i）is a city set that the ants in city i can choose. α , β are respectively the importance degree of accumulated information elements , $\tau_{ij}(t)$ is the process of ants movement and the importance degree of heuristic operator $\eta_{ij}(t)$ in the process of choosing routes. η_{ij} is the expectation value from city i to city j and it can be determined concretely by simulating a certain heuristic algorithm. In addition, ant algorithm has

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, \text{ if the kth ant passes ponit(i, j) in this circulation} \\ 0, otherwise \end{cases} \quad (1)$$

memorial function. We can use tab u_k（k=1，2，…,m）to record the cities that ant k passed at that time, tab u_k can be adjusted dynamically according to the evolution process. With the time passed, the information elements left before volatilize gradually. We can use parameter 1-ρ to express the degree of volatilization of information elements. After l time, the ants finish once circulation, the information elements adjust as follows:

Here $\Delta\tau_{ij}^{k}$ is the number of information elements left in the arc by the kth ant in this circulation. $\Delta\tau_{ij}$ is

$$\tau_{ij}(t+l) = \rho \bullet \tau_{ij}(t) + \Delta\tau_{ij} , \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=l}^{m} \Delta\tau_{ij}^{k}$$

the total increment of information elements in arc(i,j) in this circulation.

$$p_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in J_k(i)} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} \text{ if } j\in J_k(i) \\ 0, otherwise \end{cases} \quad (3)$$

3. The combination of genetic algorithm with ant algorithm (GAAA)

3.1 The design idea and total frame

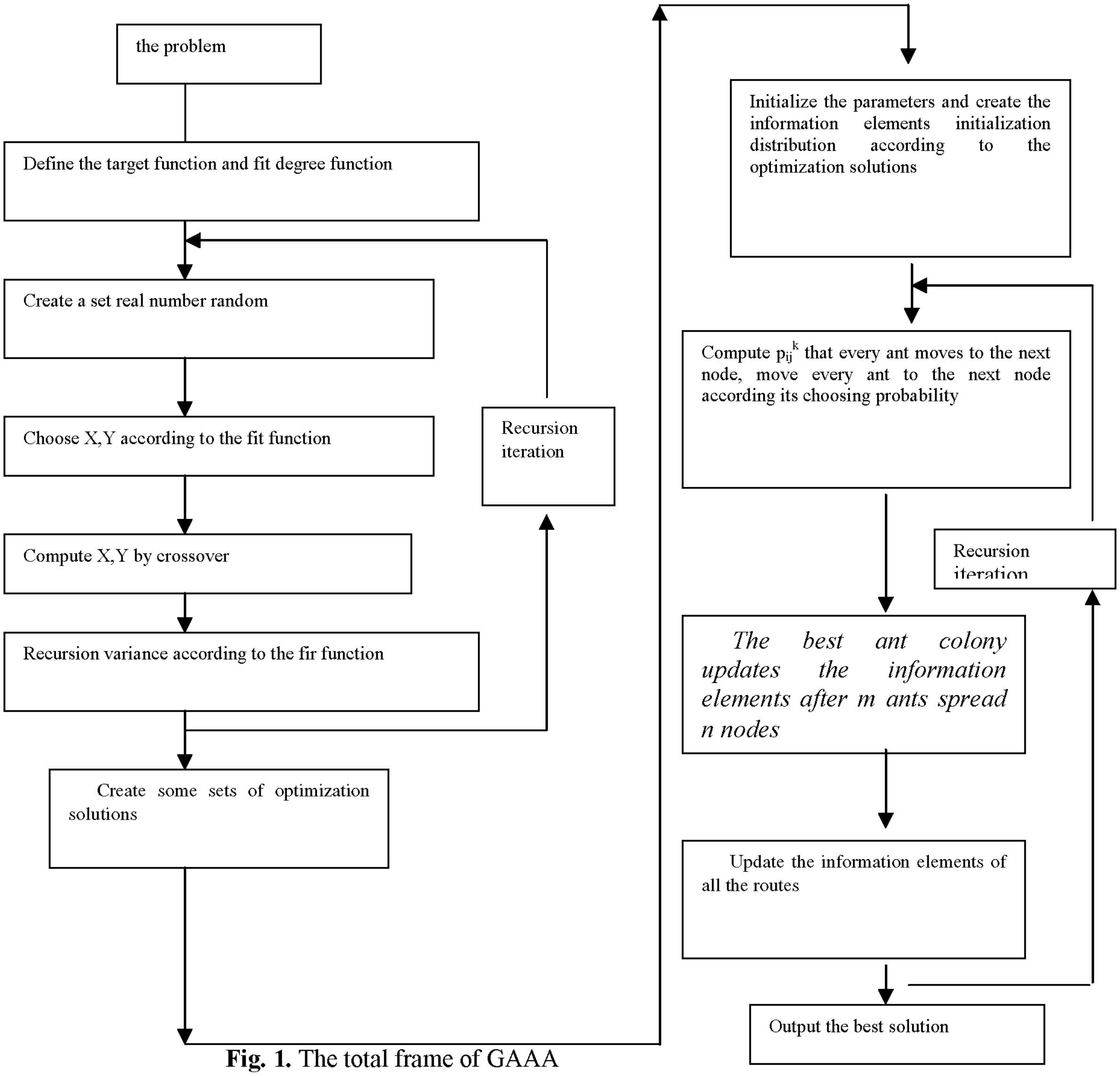

Fig. 1. The total frame of GAAA

We combine of genetic algorithm with algorithm (genetic algorithm-ant algorithm, GAAA). Its basic idea is to adopt genetic algorithm in the former process so that we can make full use of its fastness, randomicity and global constringency characters. Its result created the distribution of the initial information elements. The latter process of the algorithm is to make full use of the ant algorithm, make use of the parallel, feedback and the high efficiency of solving precise solutions in the condition that there are some initial information elements.

3.2 Decision-making variance, constraint condition and optimization model

The constraint problem can be formalized as (E,C)[9], here E= （e_1，e_2，……，e_n），it can express geometric elements, such as point, line, circle, etc; C=（c_1，c_2，…，c_m），c_i is the constraint set in these geometric elements. Usually one constraint is represented by an algebraic equation, so the constraint can be expressed as follows:

X=（x_0，x_1，…，x_n），

X_i are some parameters, for example, planar point can be expressed as（x_1，x_2）. Constraint solving is to

$$F(X_j) = \sum_{1}^{m} |f_i| \qquad \textbf{(4)}$$

get a solution x to satisfy formula (4).

Apparently if X_j can satisfy F（X_j）=0，then X_j can satisfy formula(4)．So the constraint problem can be transformed to an optimization problem and we only need to solve min（F（X_j））<ε. ε is a threshold. In order to improve the speed of ant algorithm, we adopt the absolute value of f_i not the square sum to express constraint equation set. From formula(5) and the solving min（F（X_j））<ε by Ant Algorithm, we can realize it is not necessary m=n, so the algorithm can solve under-constraint and over-constraint problem.

The decision-making variable is x_i, $i \in [0, n]$, the constraint condition is $x_i \in [L_l, L_u]$, $[L_l, L_u]$ is the scope of the decision-making x_j. The equation (5) is the model of the optimization.

$$\begin{cases} f_1(x_0, x_1, x_2, \ldots, x_n) = 0 \\ \quad\ldots \\ f_m(x_0, x_1, x_2, \ldots, x_n) = 0 \end{cases} \quad \textbf{(5)}$$

3.3 Code[10]

In the constraint solving, the individuals are expressed as X_j=（x_0, x_1, …,x_n）, $x_i \in [L_l, L_u]$, $i \in [0,n]$, x_i is the decision-making variable. In order to improve the speed and precision of calculating, we adopt the two-step coding combing the floating point number coding with Green coding. Because the range of decision-making variable is a forecasting value, the range is bigger than the actual range, the searching space is bigger than the actual space and the calculating precision is high, we introduce floating point number coding.

In order to operating the crossover and variance, we code the decision-making variables situating in the crossover point and the variance point more so that we can do more minute work.

3.4 The evaluation method of individual fit degree

The target function is F（X_j）$= \sum_1^m |f_i|$, the value range is（0, max）. But the optimization goal is to search the minimum value of the function, so the fit degree is inverse ration to the function value. We can calculate the individual fit degree Fit(X_j) is as follows:

Fit(X_j)= $1\big/ e^{\beta F(X_j)}$, $0 \le$ F（X_j）$\le$ max, $1\big/ e^{\beta \max} \le$ Fit(X_j) ≤ 1

Theoretically, F（X_j）$\to \infty$, 0< Fit(X_j) ≤ 1

Here β is the adjusting parameter of the fit degree. We can adjust the value of β in the different period of choosing and the change the choosing pressure of the colony in order to avoid the prematurity of the genetic algorithm.

3.5 Decide the running parameters of genetic algorithm

The basic parameters are:
(1) The size of the colony P is: M=200.
(2) The ceasing generation: T=100.
(3)The probability of crossover: P_c=0.5.
(4) The probability of variance: P_m=0.005.

3.6 Calculate the ceasing condition

Condition 1. The constraint is a good constraint and the change is the least every other two generation ($\le \varepsilon$). We define the change every other two generation (j,j+1 tow generations) as $= \left| \sum_{k=1}^{M} F(X_{j,k}) - \sum_{k=1}^{M} F(X_{j+1,k}) \right|$

Here M is the size of the colony space. $X_{j,k}$ shows the calculating result of the j^{th} individual in the k^{th} time.

Condition 2. The constraint system is under-constraint and $|Fit(X_j) - 1| \le \varepsilon$. Because there are infinite solutions in the condition of under-constraint, we need only to solve an arbitrary solution.

The ceasing condition= (The ceasing generation<T) $\cup$ condition 1 $\cup$ condition 2.

3.7 The improvement and the link of ant algorithm in GAAA

Initialization of the information elements: We have got some route information elements by the genetic algorithm, so we initialize the information elements as: $\tau_S = \tau_C + \tau_G$.

Here, τ_C is an information element constant according to the scope of the concrete problem. τ_G is the value of information element transformed from the solving result of the genetic algorithm.

4. The application in geometric constraint solving of Ant Algorithm

The process that Ant Algorithm solves geometric constraint problem is as follows:

Step1: Change the problem into an optimization problem by the way to write the geometric constraints into nonlinear equations.

Step2: Carry on the genetic algorithm according to the left part of the above frame. Put m ants into n nodes, that is intimate parameters, create the information elements initialization distribution according to the optimization solutions.

Step3: Every ant constructs a solution according to the state changing rule step by step, that is to say to get a viable route.

Step4: We change the information element values by local information element updating rule.

$$\tau_0 = \frac{1}{n \cdot L_{nn}} \qquad (6)$$

In the process of constructing solution, ant k changes the information element values in the arc for every arc

$$\tau(r,s) = (1-\alpha) \cdot \tau(r,s) + \alpha \cdot \Delta\tau(r,s) \qquad (7)$$

that the ant passes according updating rule formula (6).

Here, ρ （$0<\rho<1$） is an information element volatilization parameter,

L_{nn} is the route length coming from the closest neighbor area heuristic algorithm; n is the number of eigenpoint.

Step5: If all the m ants finish construct all the solutions, then go to step6, otherwise go to step3.

Step6: Change information element values by the updating rules of global information elements.

When all the m ants create solutions' number is m, the shortest route among them is the best solution in this generation. We update the information element values associated with all the arcs in this route according to formula (8).

$$\Delta\tau(r,s) = \begin{cases} \frac{1}{L_{gb}}, \text{if } (r,s) \text{ in the global best solutions} \\ 0, \text{ otherwise} \end{cases} \qquad (8)$$

Here, α （$0<\alpha<1$） is volatilization parameter.

$$\tau(r,s) = (1-\rho) \cdot \tau(r,s) + \rho \cdot \tau_0 \qquad (9)$$

L_{gb} is the route length of global best solution that has been got till now.

Step7: If the cease condition can be satisfied, we should stop; otherwise nc=nc+1, then go to step2 to do the next evolution. The cease condition can be set by the generation number of evolution or the running time and the lower limit of the shortest route length.

In the step3, ant process can be seen as an agent and its task is that it can return the starting point after visiting all the vertexes and then create a loop. Supposing the node that the ants are standing is r, the nodes that need to be visited can be expressed by set $J_k(r)$. Then the ant k moves from point r to point s according to the state-changing rule in formula (10) and chooses the next node according to the different probabilities.

$$S = \begin{cases} \arg\max_{u \in J_k(r)} \{[\tau(r,s)]^{\beta}, \text{if } q \le q_0 (utilize) \\ S, \text{ otherwise(exploiture)} \end{cases} \qquad (10)$$

Here $\tau(r,s)$ is the information element that the true ants spread on the way and is defined as a plus real number. Its value changes constantly while running. It is used to express the motivity that ant moves from point r to point s. $\eta(r,s)$ is a heuristic function to evaluate ant moves from point r to point s and its value usually can be got by the reciprocal of the distance, that is to say $\eta(r,s) = \frac{1}{d_{rs}}$. The parameter $\beta > 0$ can be used to describe the importance of heuristic function. The parameter q_0 ($0 \le q_0 \le 1$) determines the relatively importance of utilizing and expliotureing. Utilizing is to take the best route and exploitureing is to choose the route S according to the rule that the probability is higher if the concentration is bigger. The parameter q is a random number got from [0,1]. When $q \le q_0$, the best route is choose according to formula (10), otherwise according to the probability of formula (11).

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s) I \eta(r,s)]^{\beta}}{\sum_{z \in J_k(r)} [\tau(r,z) I \eta(r,z)]^{\beta}}, \text{若} s \in J_k(r) \\ 0, otherwise \end{cases} \qquad (11)$$

5. Application Instance and Result Analysis

Table 1. The result of GAAA

α	β	ρ	The convergence of	GAAA evolution

			the solution	generation
1	1	0.8	convergent	30+11
1	2	0.8	convergent	30+10
2	1	0.8	convergent	30+16
2	2	0.8	convergent	30+13
2	3	0.8	convergent	30+21
3	3	0.8	convergent	30+19
3	2	0.8	convergent	30+13
5	2	0.8	convergent	30+9
5	3	0.8	convergent	30+11
3	5	0.8	convergent	30+9

The table 1 is the result of the GAAA to the pratical constraint problems.

6. Conclusion

We can get the conclusion: The algorithm in this paper has a good effect in not only optimization capability but also time capability. The algorithm uses random colony in the genetic algorithm. This can not only improve the speed of ant algorithm but also avoid getting in the local best solution when solving the precise solutions. It is not necessary that the number of equations m is equal to the number of variable n when translating the equations into the optimization problem, so the algorithm can deal with over- and under-constraint problems. The combination of genetic algorithm with ant algorithm can reduce the parameters in the ant algorithm and avoid many blindfold experiments.

References

[1]Ding Jianli, Chen Zengqiang, Yuan Zhuzhi. On the Combination of Genetic Algorithm and Ant Algorithm. Journal of Computer Research and Development, 2003, 40(9): 1352-1356.

[2]Huang Lan.The reseach on Vehicle Routing Algorithm of Logistics Distribution. The doctoral dissertation in Jilin University，2003：50-55.

[3] Wen Wenbo, Du Wei An Abstract on the Ant Colony Algorithms. Automation in Petrochemical Industry，2002，1：19-22

[4] Colorni A. Distributed optimization by ant coloniest[R]. Proc.of 1st European Conf. Artificial Life.

[5]Dorigo M, Gianni Di Caro, Thomas Stutzle. Ant algorithms [J]. Future Generation Computer System，2000，16：5-7.

[6]Dorigo M, Vittorio Maniezzo， Alberto Colorni. The Ant System :optimization by a colony of cooperating agents[J]. IEEE Transactions on system, Man, and Cyberneticss-Part B, 1996，26（1）：1-13

[7]Dorigo M，Luca Maria Gamberdella. Ant Colony System：A Cooperative Learning Approach to the Traveling Salesman Problem[R].TR，IRIDIA，1996

[8]Walter J， Gutjahr. A graph- based Ant System and its convergence[J]：Future Generation Computer System，2000，16：837-888.

[9]Liu Shengli, Tang Min, Dong Jinxiang. Two Spatial Constraint Solving Algorithms. Journal of Computer-Aided Design & Computer Graphics，2003，15(8)：1011-1029

[10]Liu Shengli, Tang Min, Dong Jinxiang. Geometric Constraint Satisfaction Using Genetic Simulated Annealing Algorithm. Journal of Image and Graphics，2003，8(8)：938-945