IEEE International Workshop on Semantic Computing and Systems

Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem

Kangshun Li[1,2], Lanlan Kang[1], Wensheng Zhang[2], Bing Li[3]
[1]School of Information Engineering, Jiangxi University of Science & Technology, China 341000
[2]Institute of Automation, Chinese Academy of Sciences, China 100083
[3]The State Key Laboratory of Engineering Software, Wuhan University, China 430072
likangshun@sina.com

Abstract **Ant Colony Algorithm and Genetic Algorithm (GA), two bionic-inspired optimization algorithms, have great potentials to solve the combination optimization problems, respectively used in solving traveling salesman problem, but there are some shortcomings if only one of them is used to solve TSP. Performance comparative analysis have been done by using ACA and GA respectively in solving TSP in this paper. The experiments show the advantages and disadvantages used only ACA or GA, we can overcome the shortcomings if GA and ACA are combined to solve TSP and get faster convergent speed and more accurate results compared with only using ACA or GA .**

I. INTRODUCTION

Ant Colony Algorithm (ACA) and Genetic Algorithm (GA) enlightened by biologic evolutionism are put forward as a kind of bionic algorithms. But they share some commons and have difference at the same time. Genetic algorithm simulates the evolutionary process of gene, while Ant Colony Algorithm differs from it greatly. In other words, ACA imitates the behaviors of colony composed of simple individuals-society system. The two algorithms have been applied to solve some combinatorial optimization problems and have got many achievements.

In this paper, both GA and ACA will be applied to solve the traveling salesman problem (TSP)[1]. Through the analysis of their advantages and disadvantages are given by means of large numbers of experiments respectively, some future studies are provided as well.

II. BRIEF INTRODUCTION TO TSP

Traveling Salesman Problem (TSP) [1] is the most widely studied problem in combinatorial optimization. The objective is to find a cost-minimal Hamiltonian cycle, i.e. a path through a graph that starts and ends at the same vertex and includes every other vertex exactly once.

The dominating reasons for choosing TSP to test ACA and GA are illustrated as follows:

1) TSP is a shortest path problem; ACA and GA are adapted to solve this kind of problem.

2) It is not an easy task to understand TSP owing to the lack of many technical terms.

3) TSP is a typical combinatorial optimization problem. It is often used to validate a certain algorithm, making the comparison with other algorithms an easy work.

III. THE PRINCIPLE OF ACA AND IT'S METHOD TO SOLVE TSP

ACA [2,3] was firstly proposed by three Italy scholars in 1991, Colorni A, Dorigo M and Maniezzo V., and has received increasing attention from researchers, having been used to solve many difficult problems in optimization of discrete systems. The process mainly utilized the ants' ability to search the best optimization path while searching for foods. As is talked above, ACA made full use of the mechanism of behaviors of real ants in searching for food. While searching for food, real ants use a certain material called pheromone on the paths, which are left by ants, to communicate with other ants. In the trails lack of pheromone ants more or less perform a random walk. However, as soon as they sense a pheromone trails on a path in their vicinity, they are likely to follow that path, thus reinforcing this trail. More specifically, if ants at some point sense more than one pheromone trail, they will choose one of these trails with a probability related to the strengths of the existing trails. This idea has first been applied to the TSP, where an ant located in a city chooses the next city according to the strength of the artificial trails.

The symbols are defined as follows:

978-0-7695-3316-2/08 $25.00 © 2008 IEEE
DOI 10.1109/WSCS.2008.11

If m and n denote the number of ants and the number of cities respectively; $d_{ij}(i, j = 1,2,\cdots,n)$ is referred to as the distance of city i and city j, $\tau_{ij}(t)$ is the amount of pheromone between city i and city j at time t.

1) In initialization, pheromones on all paths are equal, suppose $\tau_{ij}(0) = C$ (C is constant). In the course of movement of the k[th] (where k=1,2,…,m) ant, its transfer direction is determined by the probability of pheromone as a variable in each path, which can be written as:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}\cdot\eta_{is}^{\beta}} & , \quad if \ j \in allowed_k \\ 0 & , \quad otherwise \end{cases} \tag{1}$$

Where η_{ij} means the visibility on the edge(i,j) and it can be calculated by a certain heuristic function, which is often described as: $\eta_{ij} = 1/d_{ij}$, $p_{ij}^{k}(t)$ is the probability that ant k moves form position i to position j at time t. $allowed_k = \{0,1,\cdots,m-1\} - tabu_k$ denotes ant k has been allowed of choice of the cities in the next step ($tabu_k(k = 1,2,\cdots,m)$ is the aggregation that ant k already went across cities. In the beginning there is only one element that is the starting city of ant k , With the development of evolution, element in the $tabu_k$ will be increased). α is the relatively weightiness of track. β is the parameter that determines the influence of the heuristic function.

2) Once all ants have gone through solution construction and local search, the pheromone update procedure is applied to these pheromone values. The update used for the algorithm presented in this paper is a variant of the approach proposed in [8]. It can be written as :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

Where, $\rho \in (0,1)$ means degree of attenuation of $\tau_{ij}(t)$ with the process of time, $\Delta\tau_{ij}$ is the amount of reinforcement in the edge(i ,j), which is defined as:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m}\Delta\tau_{ij}^{(k)}(t) \tag{3}$$

$\Delta\tau_{ij}^{(k)}(t)$ means pheromones that ant k moves form city i to city j in this circle. According to different forms of expression of $\Delta\tau_{ij}^{(k)}(t)$, Dorigo M have ever presented different models, known as Ant-cycle System, Ant-quantity System, Ant-density System. Their formulas are defined as follows (4)～(6) :

In Ant-cycle System：

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} \dfrac{Q}{L_k} , & if \ the \ ant \ k \ uses \ edge(i, j) \\ 0 \ , & otherwise \end{cases} \tag{4}$$

In Ant-quantity System：

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} Q \ , & if \ the \ ant \ k \ uses \ edge(i, j) \\ 0 \ , & otherwise \end{cases} \tag{5}$$

In Ant-density System：

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} \dfrac{Q}{d_{ij}} , & if \ the \ ant \ k \ uses \ edge(i, j) \\ 0 \ , & otherwise \end{cases} \tag{6}$$

Where Q is a constant that denotes the total information of which every ant leaves behind after they travel once around the all cities; L_k is length of paths that ant k have traveled in this cycle; d_{ij} means distance of two cities i and j.

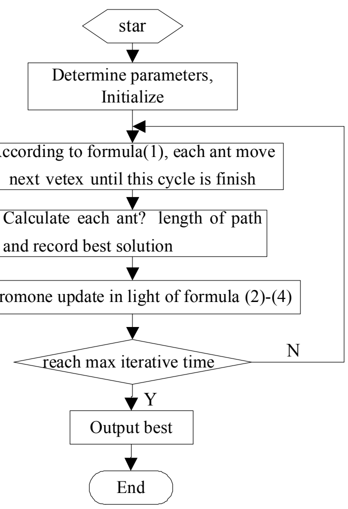

FIGURE1 THE FLOWCHART OF THE ACA

The difference of mentioned above is that Ant-cycle System uses whole information, however Ant-quantity System and Ant-density System use local information. In the process of solving traveling salesman problem (TSP), obviously, the capability of Ant-cycle System is preferable, so this paper adopts Ant-cycle System.

3) Circulating the above steps until the time of travel achieve designated time or haven't appeared better solutions in a certain period of time. The flowchart of the Ant Colony Algorithm is shown in FIGURE1.

IV. GA THE PRINCIPLE OF AND METHOD TO SOLVE TSP

Genetic Algorithm (GA) [4] are an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic selection. The basic concept of GA is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. GAs are search algorithms with random probability and has iterative process of "survival + Detection". Using stochastic technology, GAs carry efficient search in a parameter space encoded.

Genetic operation in GA including three parts—selection, crossover and mutation. Encoding parameters, settings of the initial group and control parameter, the design of fitness function and genetic manipulation comprise the core of GA.

Usually, the design of genetic algorithms mainly include following steps:

1) Choose the data coding schemes, and randomly generated a group of initial individuals as the initial colony.

This paper adopts a best natural coding scheme to travel: representation of path. For example, on the assumption that (7，4，9，5，6，1，2，8，10) denotes a path which set out from city seven and passes through city 4—9—5—6—1—2—8—10, finally, returns to city four.

2) Function of fitness is provided to evaluate the individuals and evaluate fitness value of each individual.

To evaluate individuals is to calculate the length of each path, which is also referred to as the fitness. It can be written as:

$$f(x)=\sum_{i=1}^{n-1} d(C_i, C_{i+1}) + d(C_1, C_n)$$

The individual with smaller fitness value means it has a shorter path and the individual might be better as well.

3) Result of searching is exported if algorithm above is satisfied with convergent criteria, otherwise executing followed steps.

4) According to the fitness value, copy the individuals to the next generation by a certain method.

After initial colony being produced, the fitness of each individual in the colony will be calculated. According to the value of each individual, rank the individuals from small to large. Set percentage of choice P , in order to keep the diversity of the colony, select p/2 as good the number of individuals stored into mating pool and then carried out genetic operations such as crossover, mutation.

5) Execute operation of crossover according to crossover probability p_c .

This paper adopts OX operator put forward by Davis. Namely, structuring offspring through choice of a subset of sequences to travel from parents' sequences and save it as relative sequence of the parents' sequences

6) Execute operation of mutation according to mutation probability p_m .

This paper adopts inverse mutation. Namely, the substring between deuces is reversed after choosing deuce randomly.

7) Return step 3)

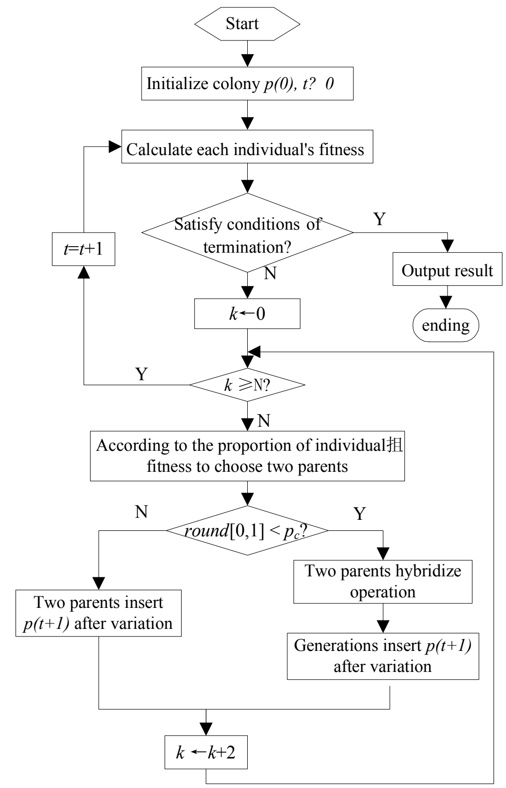

FIGURE2 THE FLOWCHART OF THE Gas

FIGURE2 shows the flowchart of the GA.

V. COMPARATIVE ANALYSIS OF THE TWO ALGORITHMS

This paper adopt ant colony system (ACS) as representation of ACA which is compared with basic GA. Simulative results of two algorithms are brought forward about Traveling Salesman Problem (TSP) with the number of cities 30, 48 and 51 respectively. Two algorithms perform times of 20 and 50 respectively in the same conditions, at the same time, the results of different problems in TSPLIB[6] are put forward. Parameters of algorithms above are as follows:

$\rho=0.1, \alpha=1, \beta=5, m=60, C=10, Q=1000$, $p_c=0.75$, $p_m=0.05$

From above experiments and Table 1 we know that ACA and GA are all bionic probabilistic search algorithms which combine with distributed computing , tend to parallel computing and have more robust. ACA simulates group behaviors between communities and environment made up with simple individual, as well as individuals, viz. social system[5], which may cause unpredictable group behavior, but GA simulate genetic evolutional process. They have common grounds on carrying out Traveling Salesman Problem (TSP):

1) They are all prone to premature convergence, so as to get into local optimum value

2) Algorithms itself in the choice of parameters and its' values can not be rigorous proved in theory, only through the experimental method to determine the optimum combination.

3) They have not special requirements to searching space, such as derivative, continuity, concavo-convex and other auxiliary information, at the same time their practical ranges are more widespread.

TABLE1 TWO ALGORITHMS'SRESULTS OF EXPRIMENT ABOUT DIFFERENT TRAVELING SALESMAN PROBLEM

name of problem	times	ACS			GA			TSPLIB
		best	average	worst	best	average	worst	
Oliver30	20	423.74	429.7	432.46	423.74	456.68	502.57	423.74
	50	423.74	428.76	432.45	423.74	425.7	439.84	
att48	20	33780	35595	36534	37880	38833	38894	33522
	50	33780	35533	36534	35633	38541	42458	
Eil51	20	426	428.4	431.42	495	512	536	426
	50	426	428	432.01	494	514	554	

Difference: Ant colony algorithm converges to the optimal path through the accumulation and update of information, but the lack of pheromone in initial stages leads to slower speed of convergence. Genetic Algorithms have rapid global searching capability, but the feedback of information in the system has not been utilized, sometimes leading to do-nothing redundant iteration and inefficient solution.

Therefore, if the size of the cities are more than 30, genetic algorithms' searching capability will gradually decline to a certain extent, when the number of the cities are too big , it can not obtains the optimal solution in finite iteration , because iterative times are too long and unbearable. Here ant colony algorithm is better than genetic algorithm, and it can reach the optimum value. When the size of cities is too big, ant colony algorithm may appear stagnation. Thereby it can not obtain optimum value.

VI. CONCLUSION

In order to overcome respective shortcomings and be form of complementary advantages, the use of randomly search and rapid of genetic algorithms[1], and pros-feedback mechanism and high efficiency characteristic of ant colony algorithm, two algorithms will be amalgamated, and the core ideas are as follows:

Firstly, Better solutions are be produced and pheromones are left behind the paths of its, pheromones in other paths are not changed. And then, ants perform operations of crossover and mutation in accordance with GA after each ant completes once traveling in the light of ACA. This operation may not improve solutions of ant paths, so only improved ant paths are replaced.

Mix algorithm of ACA and GA is applied to TSP and be tested, which is expected to obtain a better new heuristic algorithm, this is future direction for further research. At the same time, some of improved GA or ACA may be brought, such as Max-MIN Ant System (MMAS)[7], Best-Worst Ant System (BWAS) and so on.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (90612010, 60703018), the National Basic Research and Development 973 Program (2004CB318103, 2007CB310800), the National High Technology Research and Development Program of China (2008AA01Z208) and the Research Project of Science and Technology of Education Department of Jiangxi Province([2007]205).

REFERENCES

[1] Shi-Yong Li, Yong-Qiang Chen, Yan Li. Ant Colony Algorithm and its application[M]. Harbin: Harbin Industrial Press, 2004.9

[2] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]. In: The First European conference on Artificial Life, France:Elsevier, 1991:134-142

[3] Dorigo M. Optimization, Learning and Natural Algorithm(in Italian)[M]. Ph.D.thesis, Dipartimento di Elettronica, Politecnico di Mi2 lano, IT, 1992.

[4] J H Holland. Adaptation in Natural and Artificial Systems[M]. Ann Arbor: University of Michigan press, 1975

[5] Jing Feng, Ning Shu. Swarm Intelligent Theory and its Study of Application[J]. Computer Engineering and Application, 2006.17

[6] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp

[7] T Stutzle ,H Hoos. Improvements on the Ant System : Introducing MAX-MIN ant System. In Proceedings of the International Conference in Artificial Neural Networks and Genetic Algorithms, Springer Verlag, Wien. 1997.245-249

[8] Reimann, M.: Combining an exact algorithm with an Ant System for Travelling Salesman Problems, Working Paper, Department of Management Science, University of Vienna, 2003