

The Design and Analysis of TSP Problem Based on Genetic Algorithm and Ant Colony Algorithm

Shuoben Bi, Xueshi Dong, Yan Ma

*School of Computer & Software, Nanjing University of Information Science & Technology, Jiangsu Nanjing
210044, China*

Abstract

This paper firstly makes a brief introduction about TSP problem, Genetic Algorithm and Ant Colony Algorithm, then gives the basic principles and steps of the two kinds of algorithms in solving the TSP problem, does design analysis and experiments of the two kinds of algorithms for solving TSP problem, and draws some useful conclusions: under the experimental conditions, while the population during 5 to 15, the Ant Colony Algorithm for TSP problem is more effective; when the population is 1~2.5 times than cities, it can get better results by using Genetic Algorithm for solving TSP.

Index Terms: Genetic Algorithm; Ant Colony Algorithm; TSP; The shortest path; Combinatorial optimization

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the International Conference on E-Business System and Education Technology

1. Introduction

Traveling salesman problem(TSP) is a well-known one of the problems of mathematical field. Supposing a travel merchant wants to visit n cities, he must select the paths, the restriction is to visit each city only once, and finally go back to the original departure city. Routing objective is to get the distance for the minimum value among all the paths, it is called the shortest path.

There are a lot of traditional methods for solving TSP problem in decades, including exact algorithms such as linear programming methods, dynamic programming, branch and bound methods; approximate optimization algorithms such as nearest-neighbor algorithm, spanning tree method christofides algorithm, hybrid algorithm. In recent years, there are more effective ways for solving the problem to be introduced, such as Genetic Algorithm, Ant Colony Algorithm, the two kinds of algorithms are applied to solve combinatorial optimization problems, which have achieved some better results. This article applies two kinds of algorithms to TSP problem, and does experiment analysis.

Corresponding author:

E-mail address: bishuoben@163.com

2. GENETIC ALGORITHM FOR TSP PROBLEM

Genetic Algorithm (GA) is a calculation model of the simulation of Darwinian's genetic selection and natural selection in biological evolution process. Genetic Algorithm for Traveling Salesman Problem is usually mainly in the following steps:

- 1) Determine the data coding scheme, an initial set of randomly generated individuals constitute the initial population.
- 2) Give an individual assessment of strengths and weaknesses of the adaptation value function, and evaluate each individual's fitness value.
- 3) Whether the algorithm is to meet the convergence criteria, if meet, outputs search results, otherwise, implements the following steps.
- 4) According to the adaptive value, comply the copy operation in a certain manner.
- 5) According to crossover probability, implement the crossover operations.
- 6) According to mutation probability p , do crossover operations, by using inversion mutation. Namely, randomly select two points in chromosomes, and reverse sub-string between them.
- 7) back to step 3.

3. ANT COLONY ALGORITHM FOR SOLVING TSP PROBLEM

Ant colony algorithm (ACO), Dorigo.M from Italy and other scholars firstly proposed in 1996[2]. The algorithm simulates the foraging behavior of ants, There is left pheromone on the traversed path when ants are foraging, while pheromone is volatile over time. The more ants walk on a path, the more pheromone is left; the other hand, high concentrations of pheromone on the path will attract more ants.

Ant Colony Algorithm by simulating the process of searching food is to solve complex optimization problems. Taking TSP as an example, the ant system for solving n-city TSP problem model is as follows:

$$p_{ij}^k(k) = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta(t)}{\sum_s \tau_{is}^\alpha \eta_{is}^\beta(t)} & \text{if } s, j \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases}$$

m is the number of ant, $\tau_{ij}(t)$ and $p_{ij}^k(t)$ respectively represent the amount of information of the road between city i and city j as well as the probability of the transfer of ant k from city i to city j in the time t . After n times moving, each ant completes one cycle, the shortest one is recorded in closed paths walked by all ants, α 、 β respectively, accumulated information of the k th ant in the process of motion and the different roles of inspiration factor when the ants select the path; $\eta_{ij}(t)$ represents the degree of expectation of moving from the city i to city j determined by some heuristic algorithm; tabu_k ($k = 1, 2, \dots, m$) record the cities traversed by k th ant, The list dynamically is adjusted at any time. In addition, the amount of information of the path between cities can be updated by the following style:

$$\tau_{ij}(t+n) = (1-\rho)(t) + \Delta \tau_{ij}, \Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$$

the parameter $1-\rho$ represents the change of information, $\Delta \tau_{ij}^k$ as the increment of the amount of information of k th ant in this cycle of the path ij , $\Delta \tau_{ij}$ indicates the total increment on the amount of information of the path ij of this loop.

Dorigo.M had given three different models known as Ant Cycle System, Ant Quantity System, Ant Density System, their difference is the different ways of expression $\Delta \tau_{ij}^k$.

In the Ant Cycle Sytem model,

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if the } k_{th} \text{ ant uses edge}(i,j) \text{ in this cycle} \\ 0 & \text{otherwise} \end{cases}$$

This experiment of the paper adopts Ant Cycle System model.

4. EXPERIMENT AND ANALYSIS

To compare the validity and usefulness of the two basic algorithms in solving the TSP, this paper implements the program based on MFC of VC++ from the bottom. MFC with the advantages of C++, and Application Wizard allows developers to escape from programming the basic code every time, using ClassWizard and the message mapping makes developers avoid the chaos and lengthy codes.

In order to testify the validity and efficiency of our algorithms, this paper designs the experiment. The parameters of the simulation test are as follows: $\alpha=1$, $\beta=1$, $\rho=0.6$, the number of cities is 20. The greater pheromone intensity Q , the faster convergence of ant colony algorithm, but too fast speed of convergence can affect the global search ability of the algorithm. According to the actual situation, Q generally is taken between 50 ~ 150 [2]. After setting various parameters, the system runs 20 times for each question, statistical results as follows in TABLE I:

TABLE I. The Different Population of Ant Colony Algorithm and Its Required Time (time unit: ms)

Colony numbers	Optical time	Worst time	Average time	Local solutions
1	unknown	unknown	unknown	20
5	6.07	11.82	8.49	14
10	8.68	19.72	12.69	2
15	8.98	21.63	17.53	0
20	11.55	28.10	19.38	0
25	16.63	33.85	24.65	0

The optimal time of TABLE I is the fastest time of obtaining the optical solution in running each problem for 20 times; the worst time is the slowest time of obtaining optimal solution; average time is the average time of running for 20 times; the number of local solution of Ant Colony Algorithm is the numbers of local optimal solution in the solving process. The "unknown" of the table as a local solution of the system in solving the TSP problem. The population of Ant Colony Algorithm and its time for solving TSP as followed in Fig.1.

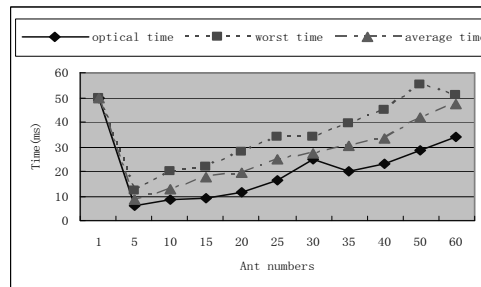


Figure 1. The population of Ant Colony Algorithm and its time for solving TSP

The number of ants in Ant Colony Algorithm is not certain, the number is generally taken 2/3 of cities, the experiment testifies that it can get better results with the number of ants during 5 to 15. From TABLE I and

Fig.1 we can see that the small numbers of ants selected has less impact on the pheromone concentration of each side, having the faster time of the system's computing, but it often falls into local optimal solution; on the other hand, the more numbers it has, the better parallelism the algorithm has, it is more likely to find the global optimal solution, while the larger groups it has, the better diversity the individuals of groups have, the possibility of a local solution is smaller, but the numbers selected of ants are too much, the corresponding computing time will be accordingly increased, so that it would affect the computational efficiency, and increase the complexity of the algorithm. Under normal circumstances, it usually takes the same or close number of ants with nodes, this paper verifies that the number of ants during 5 to 15 Ant Colony Algorithm for solving TSP problem can get better results.

To effectively validate the impacts of the population for solving TSP, the paper does experiment simulation of Genetic Algorithm for solving TSP problem. In the experiment, crossover probability of Genetic Algorithms is usually during 0.5 to 1, the verification shows that 0.75 can get better results; mutation probability is generally taken from 0 to 0.2, this article chooses 0.2; the number of cities is 20. The different populations of Genetic Algorithms for solving TSP with the corresponding computing time is shown in Fig.2.

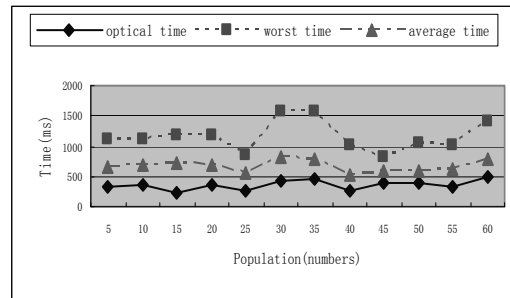


Figure 2. The population of Genetic Algorithm for Solving TSP and required time

Based on the pattern theory and implicit parallelism, if the population size is greater, genetic operators are dealing with more models to generate a meaningful modules and it has higher chance to gradually evolved into the optimal solution, so the shorter time it will spend in obtaining the optimal solution. However, on the other hand, if the the numbers of ant increase, the number of the calculation and assessment and individual fitness value as well as the computing will increase, the effectiveness of the algorithm would be reduced, in order to reduce the time consumed by the system, we should choose the appropriate population size.

From Fig.2 it can be seen, in the above given parameters, when Genetic Algorithm for solving the TSP problem, the system is easy to fall into local optimal solution, when running many times, the worst time and the optimal time of the optimal solution fluctuate greatly, and the discipline of the population and the time is not obvious. In this paper, by experimental verification, the population of algorithm is 1 to 2.5 times than cites, the time of the Genetic Algorithm for solving TSP problem is less, the chart shows that the numbers of ant during 20 to 50, the experiment can obtain good results.

5. CONCLUSIONS

Genetic Algorithm and Ant Colony Algorithm are valid search algorithms, they have own advantages and disadvantages in solving TSP of combinatorial optimization problems. Ant Colony Algorithm converges to the optimal path through the accumulation and update of pheromone, but the lack of initial pheromones leads to slow speed of the algorithm. In this paper, through experimental verification, the number of ant during 5 to 15, Ant Colony Algorithm for TSP problem can get better results; Genetic Algorithm with rapid global search capability, but the system does not use the feedback information, this often leads to redundancy inaction iteration, there are shortcomings of the slow convergence and large amount of arithmetic operations, In this paper, through experimental verification, the number of population as 1 to 2.5 times than cities, Genetic Algorithm for TSP problem can get better results. We can draw the following conclusions by the analysis: a

theoretical study of the selection of algorithm parameters and the analysis of algorithm convergence are needed; the two kinds of algorithms have their own advantages and disadvantages, to strengthen the purpose of search algorithms, heuristic strategies should be introduced, it can give clear guidance to the search process; we should also deeply study the improvement of two kinds algorithms and considering the effective combination with other intelligent algorithms and so on.

References

- [1] Pan Zhengjun, Kang Lishan, Chen Yuping. Evolutionary Computation[M]. Beijing: Tsinghua University Press, 1998,5(in chinese).
- [2] Duan Haibin. Ant colony algorithm theory and its applications [M]. Beijing: Science Press , 2005: 112-116(in chinese).
- [3] Rudolph C. Convergence properties of canonical Genetic Algorithms[J]. IEEE Trans on Neural Networks, 1994, 5(1): 96-101.
- [4] Thomas S, Holger H H, MAX-MIN ant system[J]. Future Generation Computer Systems, 2000, 16(8): 889-914.
- [5] Cao Luyin, Luo Bin, Qing Minghao. A Genetic Algorithm for Finding Shortest Paths[J]. Journal of Hefei University of Technology (Natural Science), 1996, 19 (3): 112-116(in chinese).
- [6] Wang Ying, Xie Jianying. An Adaptive Ant Colony Algorithm and Simulation[J]. Journal of System Simulation, 2002,2 :39-47(in chinese).
- [7] Xu Jingming, Cao Xianbin, Wang Xufa. Polymorphic Ant Colony Algorithm [J]. Journal of University of Science and Technology of China, 2005,35 (1) :59-65(in chinese).