

Genetic Algorithm for TSP Problem

Lijun Sun

Bohai University, Jinzhou, P.R. China

1606684199@qq.com

Keywords: TSP problem; genetic algorithm; elitist strategy; adaptive mutation operator; convergence speed

Abstract. The TSP is a typical NP problem. Therefore it is very important to the optimization for solving TSP problem. The genetic algorithm is one of ideal methods in solving it. However, the traditional genetic algorithm has the problems of slow convergence speed and local optimum. Aiming at the existing problems, the paper improved various operations and operators of traditional genetic algorithm, so that a better solution was put forward to solve the problem of local optimum prematurely. After making the selection operation, the elitist strategy was introduced to ensure the quality of the population gene; using adaptive mutation operator in mutation operation to improve the mutation quality and search result of algorithm. Genetic algorithm has characteristics of accurate result and fast convergence in solving TSP problem.

Introduction

TSP problem also is known as the traveling salesman problem, which is a typical, easy-to-describe problem but difficult to handle [1]. Suppose you have a travel business to visit n cities, you must choose which path to go and you can visit each city only once, and you must return to the original city finally. The target path must have the minimum value among all the paths [2]. The problem has more extensive engineering applications and real-life background, such as the construction of road networks, set up communication nodes of network and the distribution of goods and so on, all of these practical problems can be transformed into TSP problem to solve. Therefore, how to quickly and effectively solve the TSP has high practical value.

Currently, there are many ways to solve this problem, such as exhaustive method, backtracking method, branch and bound method, greedy method and intelligent computing method and so on. Among them, the exhaustive method, backtracking method and branch and bound method are deterministic algorithm based on search, the complexity increases exponentially with the size of the problem; greedy method and intelligent computing method are approximate algorithms, which may not be able to get the shortest path for TSP. In these methods, genetic algorithm has the characteristics of robustness, wholeness and parallelism and others, and it is one of the preferred solutions to solve optimization problems currently. Genetic algorithm is a well-known intelligent optimization algorithm, and it also is the key technology in the field of modern artificial intelligence, a large number of scientific practices had proven that genetic algorithm has high optimization capability relatively in solving TSP problem [3].

Description of the TSP Problem

The essence of TSP is to find the shortest length of the Hamiltonian circuit for a given undirected weighted graph. The problem is a very simple question from the view of above description, which is seeking the shortest path problem. In graph theory terms, there is a undirected graph $G(V,E)$, V denotes the set of vertices in graph, E denotes the set of edges in graph, the distance between the vertices is d_{ij} . TSP is to obtain the shortest path, but you must visit all vertices in V and each vertex only once. It is assumed that there are n cities, so $V=\{v_1, v_2, \dots, v_n\}$, where an access order is $S=(s_1, s_2, \dots, s_i, \dots, s_n)$ and the final s_{n+1} must be equal to s_1 , that you must return to the starting city. Wherein, $s_i \in V$, ($i=1, 2, \dots, n$). Supposing:

$$e_{ij} = \begin{cases} 1, & (i,j) \text{ in the optimal line} \\ 0, & \text{other} \end{cases} \quad (1)$$

The mathematical model of TSP problem is shown as follows:

$$\min D = \sum_{i \neq j} e_{ij} d_{ij} \quad (2)$$

The constraint conditions are:

$$\sum_{i \neq j} e_{ij} = 1, \quad j \in V \quad (3)$$

$$\sum_{j \neq i} e_{ij} = 1, \quad j \in V \quad (4)$$

$$\sum_{i,j \in V} e_{ij} \leq |U| - 1, \quad U \in V \quad (5)$$

U are all non-empty subset of V , $|U|$ is the number of vertices of graph G in the set of U .

TSP problem is a very important optimization problem, which has a widely used in computer science, operations research and engineering and other fields [4]. In theory, it is impossible to obtain the exact algorithm of the optimal solution. You can only get approximate solution of the optimal solution. The main way to solve this kind of large-scale combinatorial optimization problems is to use heuristic algorithms. Genetic algorithm is a classical heuristic algorithm [5].

Genetic Algorithm for Solving TSP Problem

Genetic process is one of the typical iterative processes. Genetic algorithm regards all individuals in a population as objects and uses randomization technique to guide an efficient search for the coding parameter space [6]. Work and basic tasks it must complete consist genetic coding, fitness function, genetic operator, algorithm' parameter and termination condition of algorithm. Among them, the selection, crossover and mutation constitute the genetic operation of genetic algorithm; the parameter coding, the setting of initial population, the design of fitness function, the design of genetic operation, the setting of control parameter, and these five elements are core contents of genetic algorithm. Genetic algorithm for solving the TSP problem contains the following steps:

(1) Coding: any one of Hamilton circuit is a ranking consisted of all vertices of an undirected weighted graph. Chromosome coding has the characteristic of one-for-one, that is, the actual representation of the problem is corresponding to the structure of the chromosome [7]. Coding uses natural number coding, each natural number is the number of one city that represents a gene on chromosome. Suppose there are n cities, and then there are n genes on chromosome, so the sequence number of genes represents a path, such as chromosome (3, 5, 2, 8, 6, 1, 4, 7, 3) represents that you start from the city 3, and pass through the city 5, 2, 8, 6, 1, 4, 7, and finally you must back to the initial city.

(2) The fitness function: fitness function is the sole criterion to evaluate the new solutions, so the goal of TSP problem is to find the shortest access path. The obtained path is longer, indicating that this path is more severe; on the contrary, the obtained path is shorter, indicating that this path is more favorable. Therefore, this paper uses reciprocal of path distance as the fitness function. The path is longer, the fitness is smaller, so the probability you choose this path is smaller; on the contrary, the path is shorter, the fitness is greater, so the probability you choose this path is greater. Fitness function $F(x)$ is designed as follows:

$$f(x) = \sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_1, v_n) \quad (6)$$

$$F(x) = 1/f(x) \quad (7)$$

Wherein, $d(v_i, v_{i+1})$ represents the path length from city i to next city, d is the length of a Hamilton circuit. The $F(x)$ is fitness function, and its value is greater, indicating its fitness is better.

(3) Selection operation: you can use the extent of individual fitness to determine that the individual should be eliminated or inherited. It simulates the principle of natural evolution of "the survival of the fittest". This ensures that the best individual continues to optimization with the increase of the evolution algebra in the evolutionary process [8]. Select operator is to decide which individuals in parent population, as well as the possibility should be copied or inherited to the next generation. Its main purpose is trying to make high-quality genes to the next generation and

improving the calculation efficiency of algorithm and probability of global convergence. For a given size group $S=\{s_1, s_2, \dots, s_n\}$, the fitness of each individual in group can be drawn as $F(s_j)$ calculated by equation (6) and (7), then the probability of that is chosen as:

$$P_{s_j} = F_{s_j} / \sum_{i=1}^n F_{s_i} \quad j=1,2,\dots,n \quad (8)$$

In addition, this paper uses the elitist strategy. A large fitness value that is elite individual should be joined to each generation and after crossover and mutation operation of this population, if this elite individual is better than the best individual in the evolutionary population, then this elite individual can be retained; if the best individual in the evolutionary population is better than the elite individual, then this elite individual can be eliminated. This approach ensures that the best individual still remains in the population after evolution.

(4) Crossover operation: the crossover and reorganization chromosome in the genetic algorithm is the main method to obtain new superior individuals. At the same time the crossover operation also has a very important effect for the quality of the genetic algorithm [9]. Its purpose is to cross the selected high-quality individuals to get new better individuals. In this paper, the elite individual and new individual after selection make partially mapped crossover operation, and cross-location is generated randomly. Specific steps are:

Step 1 choosing the best individual and a new individual in the pool of chromosome;

Step 2 cross-location is generated randomly;

Step 3 exchanging partial structure in assured cross-location according to the probability to generate new individuals;

Step 4 if the city number duplicate after cross, you can use partially mapped method to eliminate duplication, and retain do not repeat numbers.

This method is preferable to retain the better genes in parent population, so that good chromosomes can be inherited to the next generation.

(5) Mutation operation: mutation operation is a crucial step in the genetic algorithm. It simulates biological evolution to find mutated allele of chromosome in evolutionary process. Mutation operation can improve the local search ability of genetic algorithm and maintain the diversity of the population to prevent prematurity. If mutation probability is too small, so it cannot change the diversity of the population very well; if it is too big, it will destroy the good gene fragments. Therefore, how to obtain the proper mutation probability is very important, so you can use adaptive mutation operator. Adaptive mutation operator is defined as follows:

$$P = \begin{cases} k(F_{\max} - F) / (F_{\max} - F_{\text{avg}}) & 0 < (F - F_{\text{avg}}) / (F_{\max} - F_{\text{avg}}) \leq 1 \\ 0.5 & (F - F_{\text{avg}}) / (F_{\max} - F_{\text{avg}}) < 0 \end{cases} \quad (9)$$

Wherein, $k \in (0, 1)$ and k is the parameter, P is the mutation probability, F_{\max} is the maximum fitness value, F_{avg} is the average fitness value of the population. If an individual's fitness value is greater than the average fitness value, it indicates that the chromosome is better and it should be retained, so mutation probability in this case should not be made too large; if the individual's fitness value is less than the average fitness value, so you should take large mutation probability to be able to eliminate it. The overall flow chart shows in Fig. 1.

Conclusion

TSP problem is a well-known NP problem, but also one of the most classic problems in computer science and optimization problem. It is widely used not only in transportation, production, national defense, biotechnology, computers and other fields, but also it provides an ideological method platform for various algorithms in discrete optimization. Thus it has important theoretical significance to find a practical and effective algorithm, and it also has important practical value at the same time [10]. Genetic algorithm is an intelligent search algorithm that can simulate biology evolution, which is commonly used in solving the TSP problem. According to the basic principle of genetic algorithm and the characteristics of TSP problem, a genetic algorithm for TSP was designed. For the shortcomings of traditional genetic algorithm, the elitist strategy was introduced in the paper

to retain good individuals; and adaptive mutation operator was used in the mutation operation to retain high quality individuals and eliminate the bad individuals, so that mutation quality and algorithmic search result can be improved. The algorithm proposed in this paper not only has strong global search capability, but also has fast convergence speed. In short, you are able to develop genetic algorithm from coding and the design of genetic operator for the TSP problem [11].

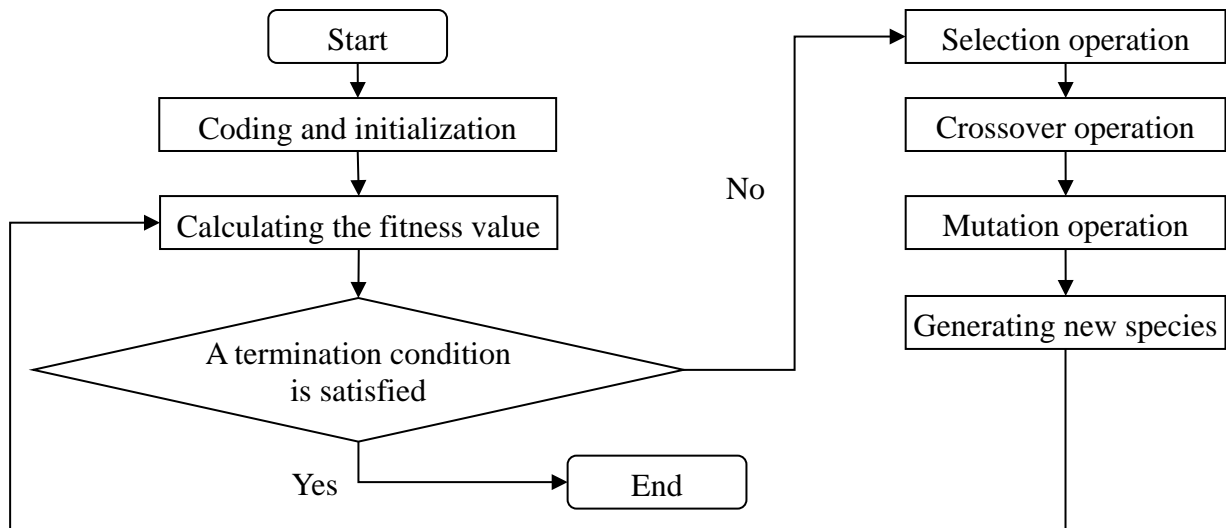


Fig .1. Flow chart of genetic algorithm

References

- [1] F. Guo, Y. Chen, "Improved Genetic Algorithm for TSP," *Modern Computer*, vol. 30, no. 17, pp. 48-51, 2014.
- [2] Baidu Encyclopedia, "TSP Problem," <http://baike.baidu.com/view/45957.htm?fr=aladdin>, 2014-11-28.
- [3] L. P. Shi, "An Improved Genetic Algorithm to Solve TSP Problem," *Experimental Technology and Management*, vol. 31, no. 7, pp. 61-64, 2014.
- [4] H. H. Liu, C. Cui, J. Chen, "An Improved Genetic Algorithm for Solving Travel Salesman Problem," *Transactions of Beijing Institute of Technology*, vol. 33, no. 4, pp. 390-393, 2013.
- [5] T. Jiang, M. J. Chen, Y. Huang, "Genetic Algorithm for Constrained Optimization TSP," *Application Research of Computers*, vol. 25, no. 5, pp. 1323-1325, 2008.
- [6] S. B. Bi, X. S. Dong, Y. Ma, "Design and Analysis of TSP Problem Based on Genetic Algorithm and Ant Colony Algorithm," *Journal of Wuhan University of Technology*, vol. 32, no. 16, pp. 89-92, 2010.
- [7] Y. Yan, "The Improved Genetic Algorithm and Estimation of Distribution Algorithms to TSP Problem," Master's degree of Chongqing University of Technology, 2013.
- [8] Q. F. Wang, D. F. Yuan, D. L. Liang, "Greedy Genetic Algorithm for Solving TSP," *Manufacturing Automation*, vol. 35, no. 2, pp. 71-74, 2013.
- [9] M. H. Yao, N. Wang, L. P. Zhao, "Improved Simulated Annealing Algorithm and Genetic Algorithm for TSP," *Computer Engineering and Applications*, vol. 49, no. 14, pp. 60-65, 2013.
- [10] H. Z. Xue, "Application and Improvement of Genetic Algorithm in Solving TSP," Master's degree of Changan University, 2006.
- [11] H. Z. Li, J. H. Yang, "Application in TSP Based on Genetic Algorithm," *Computer Knowledge and Technology*, vol. 6, no. 3, pp. 672-673, 2010.