Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 131 (2018) 937–945

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology (ICICT-2018)

Application of a Genetic Algorithm with Random Crossover and Dynamic Mutation on the Travelling Salesman Problem

Jia Xu[a], Lang Pei[a], Rong-zhao Zhu[b,*]

[a]College of Computer Science, Wuhan Qinchuan University, Wuhan and 430000, China
[b]College of Computer and Information Engineering, Hubei University, Wuhan and 430000, China

Abstract

Travelling salesman problem is a combinatorial optimization problem with wide application background and important theoretical value. The traditional method is only suitable for solving small scale travelling salesman problems, thus limiting the application and popularization of such methods. Based on genetic algorithm, the paper proposes an improved strategy combining random crossover and dynamic mutation to increase population diversity and optimize mutation characters. The simulation results show the convergence rate and the optimal solution of the improved algorithm in the paper are obviously superior to the traditional genetic algorithm, the adaptive crossover probability genetic algorithm and the improved selection genetic algorithm, and it provides a new method for the travelling salesman problem.

Selection and peer-review under responsibility of the scientific committee of the 8th International Congress of Information and Communication Technology.

Keywords: Travelling salesman problem; genetic algorithm; random cross; dynamic mutation

1. Introduction

The travelling salesman problem (TSP) is a typical NP-complete problem in combinatorial optimization problems. It is an abstract form of complex engineering problems in many fields [1]. The method of solving the TSP problem has important reference value for solving the complex optimization problem. The traditional solution of TSP problem can be divided into two categories: one is the global search method; the other is the local search method. The former is suitable for solving the small search space problem and obtains the exact solution, such as Dijkstra algorithm and Floyd algorithm. In many cases, the global search method is not suitable for the TSP in large search space due to the time complexity of the algorithm, so the local search method show more advantages. The local

* Corresponding author. Tel.: +1-303-711-8788.
*E-mail address:*93540417@qq.com

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.

Selection and peer-review under responsibility of the scientific committee of the 8th International Congress of Information and Communication Technology
10.1016/j.procs.2018.04.230

search algorithm enlarges the neighborhood size to avoid falling into the local optimal. However, as the neighborhood increases, the algorithm complexity will increase exponentially.

In order to better solve the TSP of large search space, many heuristic algorithms for solving the approximate solutions have been born in succession. Genetic algorithm provides the biological intelligence simulation according to the viewpoint of intelligent generation process. It has distinct cognitive significance, and has parallel computation behavior at the same time. It can solve any kind of practical problems. Therefore, the genetic algorithm thought has penetrated into many disciplines, and it has applied prospect in almost all scientific and engineering problems [2]. However, traditional genetic algorithm has some defects in the crossover operation and fixed mutation probability design, so that the diversity of algorithm population is not enough. When the mutation probability is too small, the algorithm converges too fast and is easy to fall into the local optimal solution. In view of the above problems, scholars have put forward many improved methods to optimize the genetic algorithm. For example, Zhu et al. [3] introduced a new computing method of the adaptive crossover rate, but the method doesn't show he relation with evolutionary algebra. Xiong et al. [4] introduced the adaptive methods for individual variation rate and population number. The adaptive adjustment makes the fine individuals have a smaller rate of variation to continue to evolve, and makes the poorer individuals have the Large mutation rate enhanced the population search ability.

On the basis of previous studies, this paper proposes an improved strategy combining random crossover and dynamic mutation to increase population diversity and optimize mutation characters. The simulation results show the convergence rate and the optimal solution of the improved algorithm in the paper are obviously superior to the traditional genetic algorithm, the adaptive crossover probability genetic algorithm and the improved selection genetic algorithm, and it provides a new method for the travelling salesman problem.

2. Genetic algorithm

Genetic algorithm is an optimization method with simulation of natural selection and genetic mechanism [5]. The algorithm has simulated the principles of biological inheritance and evolution in the nature and it has been formed by the principle of random statistics. The solution procedure is to start from an initial variable group and find the optimal solution of the problem from generation to generation. When the optimal solution satisfies the algorithm convergence or the algorithm reaches the presupposed iteration number, the algorithm is end.

2.1. Genetic operation process

Genetic manipulation mainly includes four processes: encoding, selection, crossover and mutation.

The encoding depends on the abstract mode of the real model. The different abstract ways of the same problem correspond to the different coding methods of the genetic algorithm. The encoding method is to abstract the real problem into the digital to facilitate the processing of the computer. After the coding is completed, the genetic algorithm can operate directly on the encoded string.

The selection is an operation with an elimination mechanism. It is based on the survival probability and the survival probability depends on the fitness of the individual. Individuals with high fitness are more likely to survive and are more likely to get mating rights. Individuals with low fitness are easily eliminated.

The crossover is to replace the parts of the parent's individual, thus generating the operation of the new individual. This operation selects two individuals randomly in a pairing library according to certain cross probability. Through cross operation, the searching ability of algorithm can be improved in a leap.

The mutation is an operation that combines local search, selection and crossover. This operation can avoid information loss caused by selection and crossover, maintain individual diversity, ensure the effectiveness of genetic algorithm, and make the algorithm have strong local search ability and prevent premature convergence.

2.2. Genetic algorithm design

The program flow chart of the genetic algorithm is shown in Fig. 1.

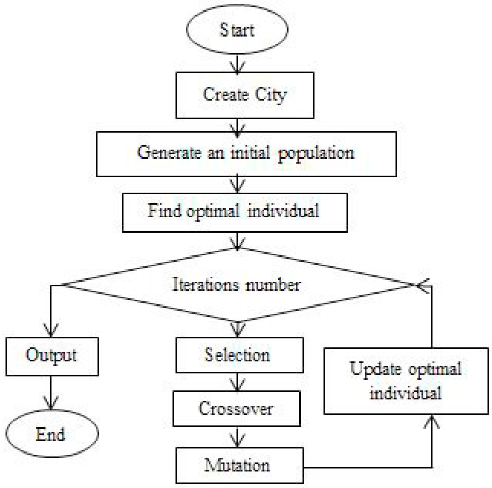

Fig. 1. The program flow chart of the genetic algorithm.

Due to the defects of the traditional genetic algorithms crossover operation and the fixed mutation probability design, the algorithm has many problems. For example, the diversity of population is not enough; when the mutation rate is too small, the algorithm converges too fast and it is easy to get into the local optimal solution. When the mutation probability is too large, the algorithm will become a random algorithm. According to the above problems, this paper proposed a random cross and dynamic mutation (RCDM) genetic algorithm has to increase the diversity of the population and optimize the mutation.

3. Improved genetic algorithm

3.1. Random cross mapping method

In the cross operation, the algorithm first generates one cross points randomly, and the crossover section length is half of the original sequence length. According to the starting position and the crossover section fixed length, the algorithm is probably generate the second intersections, thus break the number limitation of the fixed crossing points and increasing the population diversity.

The specific operation methods are as follows:

The length of the original sequence is n, and the length of the crossover section is m， their relationship satisfies the formula 1:

$$m = \left\lfloor \frac{n}{2} \right\rfloor \tag{1}$$

For example, there are two parent strings A and B, the sequence length n is 9, and the cross segment length m is 4.

A=9 1 4 5 6 7 8 3 2

B=6 8 1 2 3 9 5 4 7

In the random cross mapping method, the random crossover point is different, and the results of exchanging section in A and B are also different. The several cases are discussed as follow:

Case 1: The randomly generated cross starting point is sixth elements; the result of the random cross mapping method is the same as the result of the single-point crossover mapping method.

A=9 1 4 5 6 | 7 8 3 2

B=6 8 1 2 3 | 9 5 4 7

Case 2: A crossing point starts at any location between second and fifth elements; the random crossover mapping method is similar to the partially matched crossover [6] method with 4 crossover length. For example, the starting point is third elements.

A=9 1 |4 5 6 7| 8 3 2
B=6 8 |1 2 3 9| 5 4 7

Because the crossover section length is half of the original sequence, a new intersection point between the sixth elements and seventh elements will be generated, and the intersection point is also the end point of the crossover section.

Case 3: A crossing point starts at any location between seventh and ninth elements; the crossover section length is less than half of the original sequence length. Thus the section beyond the scope is supplemented by the section head. For example, the starting point is nine elements.

A=9 1 4| 5 6 7 8 3 |2
B=6 8 1| 2 3 9 5 4 |7

Three elements have been supplemented at the head, and the end point of the crossover section is produced between the third elements and the fourth elements.

3.2. Design of dynamic mutation probability

The mutation probability of the Algorithm in mutation operation is produced dynamically; its size varies with the change of population stability. When the population is unstable, the mutation probability can be very small; when the population is almost stable to the local solutions, the mutation probability will increase, and be more than the mutation probability of the traditional genetic algorithm, the current population produces a large number mutation, thus break away from the steady state.

Set the mutation probability of genetic algorithm is P_1 , the average character of the current generation population is C_{ave} , the best individual character is $C_{\min}$. At the same time, k is an assumed coefficient to regulate the amplitude of the probability change.

The calculation steps of the variable mutation probability are as follows:

Step 1: Through the population situation of the last generation, calculate the sum of all Individual characteristics the present generations and is denoted as C_{all} .

Step 2: Use the population size n and C_{all} to calculate the average individual character and is denoted as C_{ave} .

Step 3: Find optimality individual in the current population and record its traits as $C_{\min}$.

Step 4: calculate the current mutation probability P_2 and the result is provided to the genetic process.

The formula for calculating the current mutation probability is as follows:

$$P_2 = \left(1 - \frac{C_{ave} - C_{\min}}{C_{\min}}\right)^k \quad \left(P_2 < 0\right) \tag{2}$$

where k controls the change amplitude of the probability. When k is 1.0, the probability fluctuation is linear with the difference of C_{ave} and $C_{\min}$; when k is bigger than 1.0, the change amplitude of mutation probability can be greater; when k is 0, P_2 and P_1 are equal.

3.3. RCDM genetic algorithm design

The RCDM genetic algorithm design is as follows:

Step 1: Create an initial City distance matrix by the initial city coordinates. The distance matrix is used to store the distance between cities. Create a class of genetic algorithms.

Step 2: Initialize an initial population by the generation of a continuous single random sequence. Calculate the fitness of a single individual and find the optimal individual.

Step 3: If the algorithm exceeds the total algebra of the specified calculation then go to step 12, or else proceed with step 4.

Step 4: Calculate the current mutation probability P; it is the basis for judging whether the later operation is mutated.

Step 5: Generate an extraction probability by the size of individual fitness. By the extraction probability, the selected fine individuals are stored in the buffer, waiting to cross.

Step 6: Use the random cross mapping method to crossover two adjacent individuals, produce two new individuals. Calculate the characters of two new individuals and compared with their parent characters and the better two individuals will be left as the next generation.

Step 7: Get the random mutation number X and compare with the current mutation probability P to determine whether the algorithm produces mutation.

Step 8: when X is smaller than P then go to step 9, or else go to step 11.

Step 9: If the individual character after mutation is better than the average character of the parent then go to step 10, or else go to step 11.

Step 10: Extract a portion of the next generation individuals to execute mutation operation and then recalculate the new characters of individuals.

Step 11: Record the best individual contents of the current generation, compare its characters with the global optimum value. If the best individual is better than the global optimum value then update the global optimal individual, proceed with step 3.

Step 12: The algorithm outputs the global optimal solution.

4. Simulation design and analysis

This paper uses the traditional genetic algorithm (T-GA), the adaptive crossover probability genetic algorithm [7] (ACP-GA), the improved selection genetic algorithm [8] (IS-GA) and the random crossover and dynamic mutation genetic algorithm (RCDM-GA) to solve the classical TSP problem. Then analysis the simulation results of four algorithms, thus prove the convergence rate and the optimal solution figure of RCDM-GA is better than other algorithms.

4.1. Experimental data

The paper selects the city coordinate data of the literature [9] as a general case to discuss the TSP solution of four algorithms.

The city coordinate information is shown in Table 1 as follows:

Table 1.City coordinate data.

City-Num	City-Cd	City-Num	City-Cd
1	37,52	27	30,48
2	49,49	28	43,67
3	52,64	29	58,48
4	20,26	30	58,27
5	40,30	31	37,69
6	21,47	32	38,46
7	17,63	33	46,10
8	31,62	34	61,33
9	52,33	35	62,63
10	51,21	36	63,69
11	42,41	37	32,22

12	31,32	38	45,35
13	5,25	39	59,15
14	12,42	40	5,6
15	36,16	41	10,17
16	52,41	42	21,10
17	27,23	43	5,64
18	17,33	44	30,15
19	13,13	45	39,10
20	57,58	46	32,39
21	62,42	47	25,32
22	42,57	48	25,55
23	16,57	49	48,28
24	8,52	50	56,37
25	7,38	51	30,40
26	27,78		

In Table 1, there are 51 set data, City-Num expresses the city number, and City-Cd expresses the city coordinate.

In order to determine the algorithm parameters, the paper use a special set of city coordinate data to test. After repeated experiments, we at last select 80 as the city scale, 20 pairs of chromosomes as the experimental data, and the mutation probability is 0.45.

4.2. Analysis of simulation results

According to the city coordinate data, we use four genetic algorithms to solve the optimal solution of the same TSP; the algorithm simulation result is shown as Fig. 2:

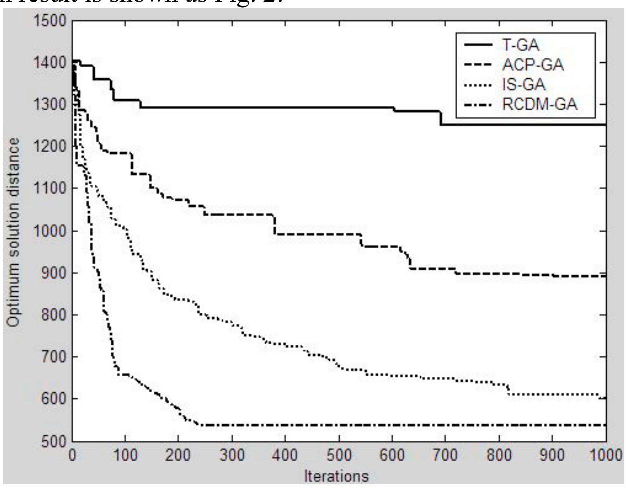

Fig. 2. Algorithm simulation result.

In Fig. 2, the performance of T-GA is not very good for TSP. Along with the continuous growth of the algebraic operation, the distance is shortened, but the effect is not obvious. The optimum solution distance of ACP-GA is obvious smaller than the traditional genetic algorithm; its improvement effect is obvious, but still not the best. IS-GA uses the current optimal solution by a certain proportion to replace the worst individual of the current population, the selection speed of the optimal solution is faster than the first two methods, and the solution is the optimal distance. RCDM-GA can calculate the more excellent solution than other algorithms, and its advantages have been maintained to the end.

In order to further reflect the difference of four algorithms in solving TSP problem, the paper calculates the spent time in finding the optimal solution and the specific value of the optimal solution by each algorithm, so as to analyze the difference between the algorithms more intuitively.

The optimal solution information of the algorithms is shown in Table 2 as follows:

Table 2. Optimal solution information of the algorithms.

Algorithm	Opt-Dis	Per-Dis	Time	Per-Time
T-GA	1145.76	+145.43%	10218	–24.10%
ACP-GA	628.96	+34.73%	9501	–29.42%
IS-GA	513.05	+9.90%	15898	+18.10%
RCDM-GA	466.84	-	13462	-

In Table 2, Opt-Dis expresses the optimal solution distance, Per-Dis denotes the distance difference, and it adopts the percentage mode. Time is the spent time of each algorithm to solve the optimal solution and its unit is millisecond. Per-Time denotes the time difference; it also adopts the percentage mode. Per-Dis and Per-Time of four algorithms are the relative results based on the RCDM-GA. The above data has retained two-bit valid numbers. By comparing with other three algorithms, the optimal solution of RCDM-GA is the best; IS-GA spent the most time for finding optimal solution. Although T-GA and ACP-GA have spent less time than RCDM-GA, but their optimal solution are inferior to RCDM-GA.

4.3. Optimal solution figure

According to the optimal solutions of above algorithms, the paper generates the optimal solution figures of each algorithm, so as to analysis the gap of four algorithms more intuitively.

In the optimal solution figures, the solid black spots represent each different city, and all the cities in table 1 have one to one solid point in the map. The number below the black spot is the number of the city. The black line is the connection between the cities. The sum of all the connections length is the best solution calculated by the current algorithm. The sum of the optimal solution length of each algorithm corresponds to the optimal solution value in Table 2.

The optimal solution figures of T-GA and ACP-GA are shown as Fig. 3(a) and (b).

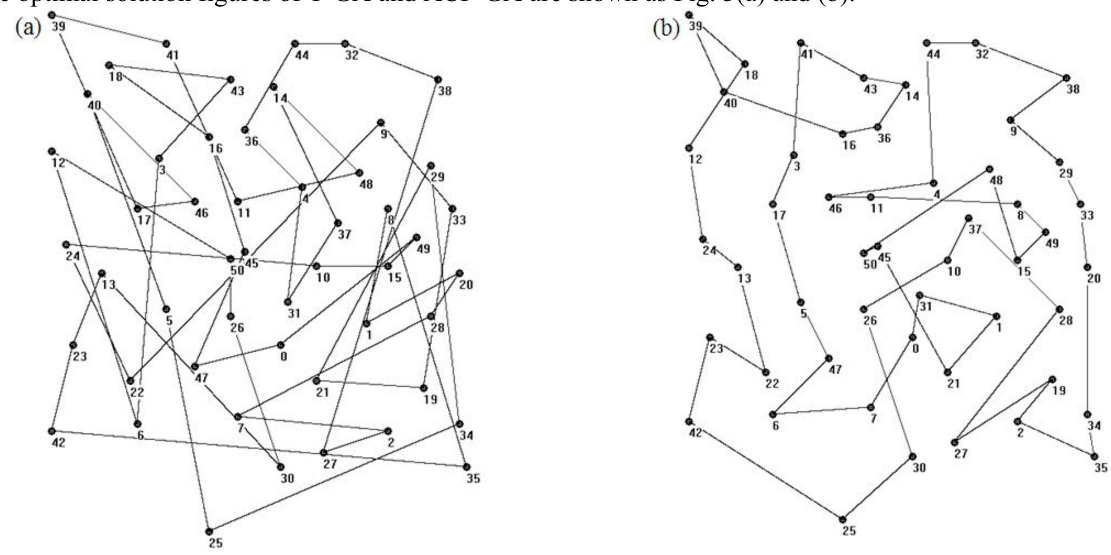

Fig. 3. (a) Optimal Solution Figure of T-GA; (b) Optimal Solution Figure of ACP-GA.

Fig. 3(a) has so many cross segments, the effect is obvious not good for solving TSP problem; Fig. 3(b) is compared with the Fig. 3(a) has been greatly improved, but there are still many cross segments, thus affecting the quality of the solution.

The optimal solution figures of IS-GA and RCDM-GA are shown as Fig. 4(a) and (b).

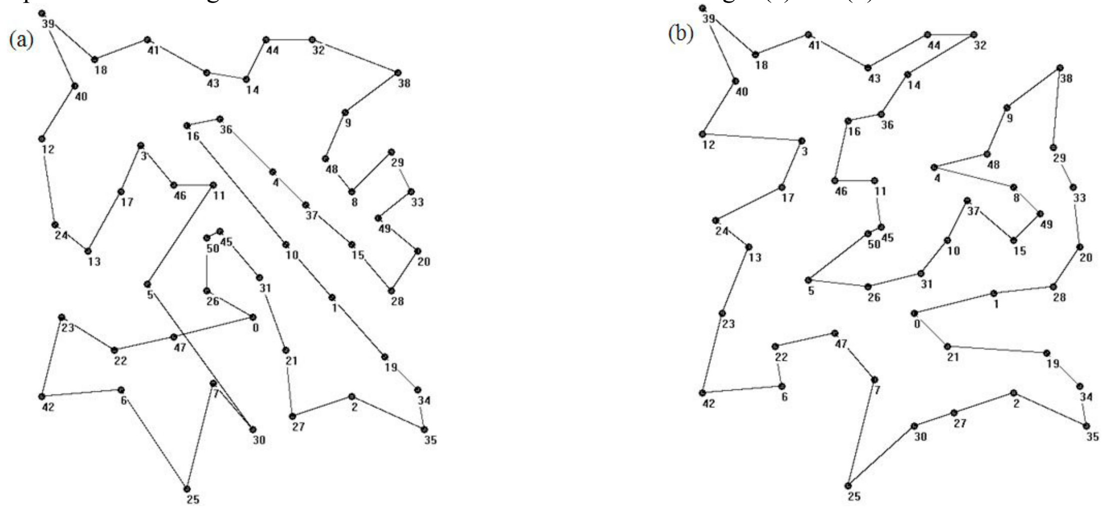

Fig. 4. (a) Optimal Solution Figure of IS-GA; (b) Optimal Solution Figure of RCDM-GA.

The cross segments in Fig. 4(a) are significantly reduced and the algorithm optimal solution becomes better. Fig. 4(b) doesn’t exist cross segments; its optimum result is the best of four algorithms.

In summary, RCDM-GA can find the more excellent solution than other three algorithms; its advantages have been maintained to the end. The optimal solution figure of RCDM-GA is better than other algorithms.

5. Conclusions

TSP is the most classical existence of all combinatorial optimization problems, and it has become one of standard problems to test a new combinatorial optimization algorithm. For solving TSP, the paper proposes an improved genetic algorithm based on random crossover and dynamic mutation to increase population diversity and optimize mutation characters. The simulation results show the convergence speed and the optimal solution of RCDM-GA are obviously better than the traditional genetic algorithm, the adaptive crossover probability genetic algorithm and the improved selection genetic algorithm. In the next research, the paper will further verify the applicability of the algorithm on other complex problems, so as to prove the algorithm itself has universal advantages, and has a certain popularization and application.

Acknowledgements

This paper is sponsored by the guiding project of the science and technology research program of the Hubei Provincial Education Department (No.B2017409).

References

1. Kang Zhou, Xiao-li Qiang, Xiao-jun Zhou, Jin Xu. Algorithm of TSP. Computer Engineering and Applications; 2007. 43 (29): 43-47.
2. Ji-ke Ge,Yu-hui Qiu, Chun-ming Wu, Guo-lin Pu. Summary of genetic algorithm research. Application Research of Computers; 2008. 25 (10): 2911-2916.
3. Jin-jun Zhu, Kai Gao, Wan-zhen Zhou. The Application of Genetic Algorithm in Data Mining. Computer Engineering and Applications; 2003. 39 (17): 203-206.

4. Jun Xiong, Dun-tang Gao, Si-dan Du, Qin-hong Shen. Genetic algorithm with mutation probability and population size adaptation. Journal of Southeast University (Natural Science Edition); 2004. 34 (4): 553-556.
5. Cheng-zhi Cao. Artificial intelligence technology. Beijing: Tsinghua University Press; 2010.
6. Yan-chun Liang. The theory and application of swarm intelligent optimization algorithm. Beijing: Science Press; 2009.
7. Lan Wang. Research of the convergence of GA based on self-adaptive crossover probability and self-adaptive mutation probability. Journal of Yunnan Normal University (Natural Science Edition); 2010. 30 (3): 32-37.
8. Gui-gen Miu. Matlab Programming for Solving TSP Based on Genetic Algorithm. Journal of Chuxiong Normal University; 2011. 26 (3): 5-10.
9. Xing-xin Liao. Application and Research of Ant Colony Algorithm in Travelling Salesman Problem. Master's degree thesis of Sichuan University; 2006.