

Application of An Improved Genetic Algorithm with The Search Space Compression in TSP

YI Song^{1,a}, Nini Wei^{2,b}

¹ Department of computer science& Information Engineering, Wuhan Bioengineering Institute, Wuhan, China

² Department of computer science& Information Engineering, Wuhan Bioengineering Institute, Wuhan, China

jznu_song@yahoo.com, wnn19821024@163.com,

Keywords: TSP ; Genetic Algorithm; incomplete evolution; search space compression

Abstract.The Traveling Salesman Problem is a combinatorial optimization problem, the problem has been shown to belong to the NPC problem, the possible solution of Traveling Salesman Problem and the scale of the cities have the exponential relation, so the more bigger of the scale. In this paper, improve the search process of the genetic algorithm by introducing the idea is to compress the search space. The simulation results show that for solving the TSP, the algorithm can quickly obtain multiple high-quality solutions. It can reduce the blindness of random search and accelerate convergence of the algorithm.

Introduction

In the research and application of the information technology, the problem of TSP (Traveling Salesman Problem) is considered to be a standard issue, give an effective solution to the TSP problem has a very high value in the theory of computation and practical applications. This is a typical problem of the optimal combination, because the search space is so larger, solves the TSP more hard, it has been proved to belong to NPC problem.

By introducing the idea to compress the search space in the paper, using the combination of incomplete evolution rules and the search space compression to improve the genetic algorithm search process. By incomplete evolution to obtain a group of near optimal solution, and then compared the near optimal solution contained in the same mode, the search space is divided into one or more optimal solution domain, and local optimization.

Basic strategies and ideas

The mode refers to the same part or structure of the solution about combinatorial optimization problems, we can defined a mode which is the path have the same city connected about the optimal solution of the TSP. The size of the mode is defined as a ratio about the number of variables in the pattern and the scale of the problem, the size of the optimal solution mode can reflects distribution of the optimal solution in the space. In the early stages of search, the solution in the population has the little same part for each other, the small size of the mode of all solutions, with the iteration progresses, the solution in the population have the same part is constantly increase, that the size of mode is increasing. The solution in the population close to the near-optimal solution, the solution contains the same portion to 70% -80%, it contains most of the information of the optimal solution. However, if the optimal solution of multiple dispersed in the solution space, the pattern that constitutes the optimal solution will be very different. The solution in the population close to the different optimal solution has the little same part and the size of the pattern of the whole population will be very small.

Because the solution near the optimal solution contains most of the optimal solution pattern, therefore the solution which is closer to the optimal solution can more to reflect the information of

the optimal solution. Therefore, we can take advantage of information of the near-optimal solutions to estimate the distribution characteristics of the optimal solution within the search space.

Improved genetic algorithm with the search space compression

Genetic operation

Use the order of the Traverse City to represent each individual coding string, in this the allele gene are n cities serial. Such as a six-city's two paths as follows: $T_x = \{1\ 3\ 4\ 5\ 2\ 6\}$, $T_y = \{2\ 1\ 4\ 5\ 3\ 6\}$.

Through by statistics the optimal solution with a large number of the TSP in the literature [3-4], that found the solution have regional characteristics, that is in the optimal solution each city select one from the Max cities as a close neighbor city to visit. Therefore, in solving through establish the adjacency list of cities A_i for each city c_i , the table records the Max city which the distance from the city c_i is the most recent, when traversal select a city directly from A_i as adjacent cities.

According to the principle of the close-by visit with the solution of the TSP, randomly selected two genes i and j , the location of the crossover point after i and j , use the gene between i and j as the cross-regional, for each gene p ($p=i+1, i+2, \dots, j$) in the cross region, obtained the gene q when $t_q^x = t_p^y$ in the individual T_x , obtained the gene r when $t_r^y = t_p^x$ in the individual T_y , and then swap t_q^x and t_r^y , swap t_p^x and t_p^y . The results are T_x' and T_y' , the t_q^x represent the q gene in the T_x' , the t_p^y represent the p gene in the T_y' . Such as $T_x = \{2\ 4\ 1\ 8\ 6\ 7\ 10\ 9\ 5\ 3\}$, $T_y = \{1\ 3\ 2\ 4\ 5\ 10\ 8\ 9\ 6\ 7\}$, selected the location of the crossover point after the third and the fifth gene, the new tour route produce process as follows:

$$T_x = \{2\ 4\ 1\ | 8\ 6\ | 7\ 10\ 9\ 5\ 3\} \rightarrow \{1\ 8\ 2\ | 4\ 6\ | 7\ 10\ 9\ 5\ 3\} \rightarrow \{1\ 8\ 2\ | 4\ 5\ | 7\ 10\ 9\ 6\ 3\} = T_x'$$

$$T_y = \{1\ 3\ 2\ | 4\ 5\ | 10\ 8\ 9\ 6\ 7\} \rightarrow \{1\ 3\ 2\ | 8\ 5\ | 10\ 4\ 9\ 6\ 7\} \rightarrow \{1\ 3\ 2\ | 8\ 6\ | 10\ 8\ 9\ 5\ 7\} = T_y'$$

Use the inversion mutation it is get the gene between the two mutation points to reverse order, thus creating a new path. Randomly selected a gene i , and then randomly selected another city j from the adjacency list of cities A_i of the gene i , use the gene between i and d_j as the mutation-regional, and then reverse order to the each gene in the mutation-regional.

Algorithm processes

Step 1: according to the scale of the problem to select the appropriate parameters.

Step 2: execute the incomplete evolution by the above genetic operation to generate a set of near-optimal solutions.

Step 3: the size of mode of the multiple near-optimal solutions is obtained by calculating, if these solutions contain the size of same mode is larger, then the search space as an optimal solution domain, go to the Step5, and otherwise go to Step4.

Step 4: according to the size of the same mode of the solution, the search space is divided into a number of different optimal solution domains, extract the identical pattern of the solution in the optimal solution domain, go to Step5, and execute the parallel optimization in a different optimal solution domain.

Step 5: after this use a temporary city to instead of the pattern in the optimal solution domain, and then update the adjacency list of cities A_i , the remaining non-pattern variables use the complete evolutionary algorithm to optimize.

Simulation experiment

The algorithm has been tested with the classic problem berlin52 and kroA100 as an example[5], for the two test problems we respectively gave 10 tests, take the crossover probability is 0.9, take the mutation probability is 0.1, the maximum number of iterations is set to 5000, according to the

steps described in 2.2 to evolve, take the current six best solution as a near-optimal solutions in experiment, calculate the size of mode. The table 1 lists of the results about the calculation, m means the numbers of the optimal solution domain is divided.

table 1 experimental results

<i>problem</i>	<i>best solution of the TSPLIB</i>	<i>best solution in the paper</i>	<i>evolution algebra</i>	<i>average deviation</i>	<i>the average value of ρ</i>	<i>m</i>
berlin52	7542	7544	52~132	0.03%	0.71	1
kroA100	21282	21289	76~279	0.3%	0.45	1

From the table we can find the average value of ρ respectively reach to 0.71 and 0.45 of the two problems, that is said respectively get the six near-optimal solutions have the same city path reach to 71% and 45%, therefore we can conclude that the two problems are only divided into an optimal solution domain, then use a temporary variable to instead the same path, and the other is not the same path to do a problems respectively get the optimal path by testing shown as the figure1 and figure2. complete optimized until you find the optimal solution. The berlin52 and the kroA100

Use the basic genetic algorithm (GA) to test the two problems with the same initial parameters, when test the berlin52 problem GA found the local optimal solution 7865 in the 1500 generation, however the proposed algorithm in the paper found the global optimal solution between the 52 generation and the 132 generation, its convergence obviously accelerate; when test the kroA100 problem GA found the global near-optimal solutions 21798 in the 4096 generation, however the proposed algorithm in the paper found the global optimal solution between the 76 generation and the 279 generation, its convergence is also obviously accelerate.

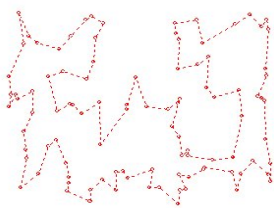


Figure1 the path of berlin52

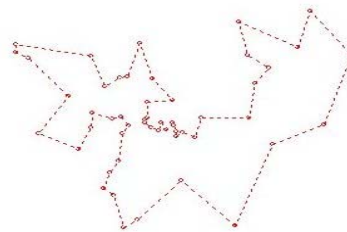


Figure 2 the path of kroA100

Conclusion

By the experiment of TSP, the improved genetic algorithm use the search space compression can reduce about 30% to 40% of the city handling capacity. It greatly reduced the scale of the problem and gets a better quality of the solution, and then accelerates the convergence to the optimal solution.

Acknowledgment

This paper is supported by National Natural Science Foundation of Hubei(No.2011CD297) and Wuhan Education Projects Foundation(No.2009K100)

References

[1] M. Li, L. Wu, K .B. Zhang, Comparative study of several algorithms for traveling salesman problem, Journal of Chongqing University Of Posts And Telecommunications. No 5,(2008),pp. 124-126

[2]Y .C. Gao, L.M. Liu, X. Wei, Delomposing Algorithm Bsed on Space Partition, Journal of System Simulation. No 16, (2009),pp.113-117

- [3] S .Y. Zeng,L .S. Kang,L .X. Ding, A new method of evolutionary algorithm for mixed-integer Nonlinear optimization Problem,Journal of Wuhan University (Natural Science Edition), Vol.46,No. 5,(2000) ,pp.554-558
- [4] J. Zhang,Y. Liu, F.L .Lu,Parallel Research and Implementation of Ant Colony Algorithm to Solve Problem of TSP,Journal of Computer Technology and Development, Vol.21,No. 5, (2011),pp. 72-74
- [5] L .Y.Jia, C.H. ZHOU ,Strategy for Generic Algorithms to Slove TSP Problem Quickly,Journal of Computer Engineering,Vol.34,No.5, (2008),pp.174-175