International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

An Improved Genetic Algorithm for the Traveling Salesman Problem

TIAN Yuan, PING Xue-Liang, BAI Liang-Liang, JIANG Yi
Jiangsu Key Laboratory of Food Manufacturing Equipment and Technology, Jiangnan University;
School of Mechanical Engineering, Jiangnan University, Wuxi 214122, China

***Abstract* – The traveling salesman problem concerns the best way to visit a set of cities and to minimize the length of the tour over all the cities. The problem is a typical NP-complete problem, which is of major importance in real world applications. An effective genetic algorithm is proposed for the problem in this paper, which combines elitist with the novel genetic algorithm. What's more, tournament selection, partially mapped crossover (PMX) and 2-opt mutation are applied to evolve the individuals. Four testes (bays29, att48, kroa100 and pr137) from the TSPLIB are chosen as the text bed. The conducted experiments demonstrate that the combination of partially mapped crossover with 2-opt mutation greatly enhance the ability to find the global optimum. Compared with the simple genetic algorithm and the nearest neighbor algorithm the computation time, stability of global convergence and accuracy of the proposed algorithm are much better.**

***Keywords* – Traveling Salesman Problem, Genetic Algorithm, Partially Mapped Crossover, 2-Opt.**

I. Introduction

Traveling Salesman Problem (TSP) is known to be NP-complete [1]. It was first considered mathematically in the 1930s by Merrill Flood [2] TSP can be formulated as an integer linear program [3].

In its simplest form, TSP involves finding an optimal route for visiting n cities and returning to the point of origin, where the intercity distances are known. Namely, finding a round-trip tour $T=(t_1,t_2,\cdots,t_n)$ to make the follow quantity minimized:

$$f(T)=\sum_{i=1}^{n-1} d(t_i,t_{i+1})+d(t_n,t_1)$$

This quantity is referred to the tour length t_i is the number of the city and $d(t_i.t_j)$ represents the distance from city i to city j.

TSP problem is a classic combinatorial optimization problem which has long been regarded as a representative problem of studying algorithms capability. Since the description of the problem is simple, and its actual model has been widely used in path, network, distribution and other optimization problems. It has attracted the attention of researchers in many areas of its algorithm improvement for a long time.

The Genetic Algorithm (GA) is an adaptive global optimization algorithm that is modeled after the simulation of biological heredity and evolution in the natural environment [4]. Since it has a good global search capability, it is the most effective method to solve various optimization problems.

The novel genetic algorithm with tournament selection, partially mapped crossover (PMX) and 2-opt mutation is addressed in this paper. In the interest of avoiding premature convergence and evolutionary stagnation, the strategy of elitist is applied to the algorithm. In order to evaluate and analyze the performance of the improved genetic algorithm, four experiments (bays29, att48, kroa100 and pr137) from the TSPLIB are carried out to compare it with the simple genetic algorithm and nearest neighbor algorithm.

II. Related Work

A. *Simple Genetic Algorithm (SGA)*

The simple genetic algorithm (SGA) is a classical form of genetic algorithm. In the algorithms the roulette wheel selection, single point crossover and insertion mutation are applied to evolve the individuals [5]. Here is the outline of the simple Genetic Algorithm:

1. Generate random population.
2. Evaluate the fitness of each chromosome in the population.
3. Create a new population by repeating following steps until the new population is complete.

(1) Selection: Select two parent chromosomes from a population by Roulette Wheel Selection;
(2) Crossover: With a crossover probability cross over the parents to form a new offspring by single point crossover;
(3) Mutation: With a mutation probability mutate new offspring at each locus by insertion mutation;

4. Place new offspring in a new population for a further run of algorithm.
5. If the end condition is satisfied, stop, and return the best solution in current population. If not, go to step 2.

B. *Nearest Neighbor Algorithm*

The nearest neighbor (NN) algorithm or so-called greedy algorithm perhaps the simplest and most straight forward TSP heuristic [6]. The key to this algorithm is always visit the nearest city. This algorithm quickly yields an effectively short route. The algorithm on average yields a path 25% longer than the shortest possible path [7].

Given a set of ***n*** points and a query point, ***q***, the nearest-neighbor (NN) problem is concerned with finding the point closest to the query point. These are the steps of the algorithm:

Nearest Neighbor, $O(n^2)$

1. Select a random city.
2. Find the nearest unvisited city and go there.
3. Are there any unvisited cities left? If yes, repeat step
4. Return to the first city.

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

III. Proposed Algorithm

The proposed algorithm is an improvement and innovation based on the Genetic Algorithm. In the algorithms the tournament selection, partially mapped crossover, 2-opt mutation and elitist are applied to evolve the individuals. The framework of the improved genetic algorithm consists of the following steps:

1. Initialization: Generate the population of M individuals randomly.
2. Evaluate the fitness of each chromosome in the population
3. Selection: First copy the best individual (or a few best individuals) to new population, then find N individuals randomly and copy the highest fitness one to new population.
4. Crossover by partially mapped crossover: Choose M*Pc/100 pairs of individuals randomly and produce an offspring from each pair of individuals.
5. Mutation by inversion mutation: Choose Pm% of individuals randomly and improve them by inversion mutation.
6. If the end condition is satisfied, stop, and return the best solution in current population, if not, go to step 2.

A. Chromosome Coding

In genetic algorithms, a chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. Coding is the genetic representation of feasible solution in Genetic algorithm, it is the first step of genetic algorithm, directly affects the selection, crossover and mutation operators. In this paper, permutation encoding is used to represent the cities, the chromosome is a string of numbers, which represents the traversal order of the cities. Taking six cities as an example, code string 123456 indicates that the salesman starts traveling from city 1, sequentially go to city 2,3,4,5,6, and finally return to city 1.

B. Population Initialization

Population initialization is used to generate candidate solutions, it can affect the convergence speed and also the quality of the final solution. For TSP the most commonly used method to generate candidate solutions is random initialization. Since permutation encoding is used to encoding the problem, the generation of initial population using 'randperm' function. This function can generate a random permutation matrices as the initial solution of the problem.

C. Objective Function and Fitness Function

The traveling salesman problem concerns the best way to visit a set of cities and to minimize the length of the tour over all the cities. Thus the objective function of the TSP can be designed as the total length of the tour:

$$f(T) = \sum_{i=1}^{n-1} d(t_i, t_{i+1}) + d(t_n, t_1)$$

Since the fitness function is used to summarize how close a given design solution is to achieving the set aims. The fitness function designated as the derivative of the objective function, namely $fit = \frac{1}{f(T)}$.

D. Partially Mapped Crossover (PMX)

Partially mapped crossover is performed in the center region to yield the following *transitionary* offspring, the crossing region is chosen by selecting two crossing sites[8].

Given two parents *s* and *t*, PMX randomly select two crossing sites – like 2-point crossover. First, yield the transitionary offspring by exchange the values between two crossing sites of *s* and *t*. To keep the string a valid chromosome the cities between the two crossing sites are filled by cross-referencing with the parent of the alternate chromosome. Take 7 allele as an example to explain the crossover technique. Fig.1 shows how PMX works parents *transitionary* offspring offspring

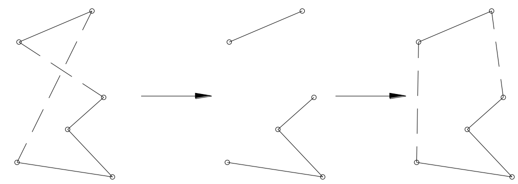
parents		transitionary		offspring
A = 14x256x37	→	A~=14x371x37	→	A' =64x371x25
B = 26x371x54		B~=26x256x54		B' =31x256x74

Fig.1. an example of PMX

Where 'x' marks the crossing site.

E. 2-opt Mutation

2-opt is a simple local search algorithm for solving the combinatorial optimization problems. In order to improve the performance of the algorithm, using 2-opt instead of a simple mutation operator. The main idea of it is to reorder a route to make it shorter. The basic step of 2-opt is to delete two edges from a tour and reconnect the remaining fragments of the tour by adding two new edges[9]. Fig.2 shows how a single 2-opt move works.



identify two edges remove them add two new edges

Fig.2 single 2-opt move

A 2-opt move is the same as inverting a subsequence of cities in the tour. Here is an example for the 2-opt mutation: Assuming a solution to the current problem x = {a, b, c, d}, and assuming that the optimal solution of x is $x_{\min}$. Apply x with 2-opt mutation will produce an additional 6 solutions: {a, b, d, c}, {b, a, c, d}, {c, b, a, d}, {d, b, c, a}, {a, c, b, d}, {a, d, c, b}, then the algorithm examines all the possible solutions and applies the best exchange, namely take the best solution of the tour as the new optimal solution $x_{\min}$.

IV. Experimental Results and Performance Comparisons

Both methods, simple genetic algorithm (SGA), nearest neighbor algorithm (NN) and the newly proposed one, are implemented as matlab programs. Four testes (bays29, att48, kroa100 and pr137) from the TSPLIB[10] is chosen as the text bed. This library is provided and maintained by the Research Group on Discrete Optimization at Heidelberg University. As it is true for many problems in the TSPLIB pool, the problems used in our work have known optimal solutions.

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

The simple genetic algorithm and the improved genetic algorithm were run with the same settings: 86.9% crossover, the population size of 500, 15% elitism, 0.7% mutation. Both the above two algorithms and the nearest neighbor algorithm terminate after 1000 generations. Table 1 to 4 shows the experimental results of the proposed approach and the other two algorithms tested on three TSP problems.

Table 1: Comparison of algorithm performance test on bays29

Algorithm	bays29(2020)	
	Best path length	time(s)
SGA	9074.1	74
NN	9964.8	60
Improved GA	9076.9	65

Table 2: Comparison of algorithm performance test on att48

Algorithm	Att48(10628)	
	Best path length	time(s)
SGA	33600.6	216
NN	39236.9	168
Improved GA	33588.3	174

Table 3: Comparison of algorithm performance test on kroa100

Algorithm	Kroa100(21282)	
	Best path length	time(s)
SGA	22453.0	1266
NN	24698.5	702
Improved GA	21363.0	726

Table 4: Comparison of algorithm performance test on gr137

Algorithm	Gr137(69853)	
	Best path length	time(s)
SGA	77266.3	2106
NN	93013.9	1037
Improved GA	71444.8	1362

The experimental results of the texts in table 1 to 4 shows that the proposed improved GA performs preferably for these tested problems. Although its optimization results deviate from the optimal solution, the optimization effect of the proposed algorithms is superior to the other two algorithms, especially when the count of cities are more. In the case of kroa100 and gr137 tests the experimental results of the proposed algorithm are superior to the other two algorithms, which indicates that the combination of Partially Mapped Crossover with 2OPT greatly enhance the ability to find the global optimum. Compared with simple genetic algorithm, the best path length and time consuming are both significantly improved. Although the consumption of time of the proposed algorithm is more than the nearest neighbor algorithm, the best path length was significantly better than the nearest neighbor algorithm.

Conclusion

An effective genetic algorithm is proposed for solving the travel salesman problem in this paper. The 2-opt heuristic operator provides a powerful way to jump out the local optimum, which promotes the improved genetic algorithm produce good results. Furthermore, the combination of partially mapped crossover (PMX) and the 2-opt heuristic operator reduces runtime without destroying the basic power of the algorithm, which greatly enhance the ability to find the global optimum. It is possible for other genetic algorithms to benefit from such a policy.

A future project will compare the proposed algorithm with the 3opt swaps and simulated annealing with regard to their runtimes and other solution characteristics.

References

[1] Gail, D.D., Reinaldo, J.M., Gary, E.W., Meta-RaPS.: A Simple and Effective Approach for Solving the Traveling Salesman Problem. Transportation Research Part E. 2005, pp.115-130
[2] Lawler E L, Lenstra J K, Rinnooy Kan A H G, et al. The Traveling Salesman Problem; A Guided Tour of Combinatorial Optimization[J]. 1985.
[3] Papadimitriou C H, Steiglitz K. Combinatorial optimization: algorithms and complexity[M]. Courier Corporation, 1998.
[4] Li L, Zhang Y. An improved genetic algorithm for the traveling salesman problem[M]//Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques. Springer Berlin Heidelberg, 2007, pp.208-216.
[5] Vose M D. The simple genetic algorithm: foundations and theory[M]. MIT press, 1999.
[6] Nilsson C. Heuristics for the traveling salesman problem[R]. Tech. Report, Linköping University, Sweden, 2003.
[7] Johnson, D.S. and McGeoch, L.A.. "The traveling salesman problem: A case study in local optimization", Local search in combinatorial optimization, 1997, pp.215-310.
[8] Goldberg D.E., Lingle R.J.: Alleles, Loci and the TSP. Proceedings of the First International Conference on Genetic Algorithms and Their Applications. 1985, pp.154-159.
[9] Djordjevic M, Tuba M, Djordjevic B. Impact of grafting a 2-opt algorithm based local searcher into the genetic algorithm[C]//Proceedings of the 9th WSEAS international conference on Applied informatics and communications, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point. 2009, pp. 485-490.
[10] http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/

Author's Profile

TIAN Yuan
was born on July 22, 1991. She is pursuing her M.E. in Mechanical Design and Theory in Jiangnan University, Wuxi, Jiangsu, China. Her research includes computer science mechatronics.
Email:teaya10@126.com

Dr. PING Xueliang
is a professor in the school of Mechanical Engineering, Jiangnan University. He focus on the research and teaching in the fields of mechatronics and robotics.
Email: ping@jiangnan.edu.cn