2009 Fifth International Conference on Natural Computation

An Improved Greedy Genetic Algorithm for Solving Travelling Salesman Problem

Zhenchao Wang, Haibin Duan, and Xiangyin Zhang
School of Automation Science and Electrical Engineering, Beihang University
Beijing, 100191, P R China

e-mail: wzc4884@163.com

Abstract—**Genetic algorithm (GA) is too dependent on the initial population and a lack of local search ability. In this paper, an improved greedy genetic algorithm (IGAA) is proposed to overcome the above-mentioned limitations. This novel type of greedy genetic algorithm is based on the base point, which can generate good initial population, and combine with hybrid algorithms to get the optimal solution. The proposed algorithm is tested with the Traveling Salesman Problem (TSP), and the experimental results demonstrate that the proposed algorithm is a feasible and effective algorithm in solving complex optimization problems.**

Keywords-***Genetic algorithm(GA); Greedy algorithm; Improved greedy genetic algorithm(IGAA); Traveling Salesman Problem; Base point***

I. INTRODUCTION

Genetic algorithm (GA) was firstly put forward by J. Holland in 1970s to study the self adaptation behavior of natural system [1]. It is a random search algorithm and global optimization algorithm. It has strong fault tolerance and strong robustness. It is also easy to combine with other methods in optimization, which is suited to solve complex combinatorial optimization problem system.

However, genetic algorithm is too dependent on the initial population, is easy to premature convergence, and has insufficient capacity of local optimization.

Greedy algorithm, at the time of solving the problem, always makes the choice which at present appears to be the best. In other words, it is not optimal while the whole is taken into account, and what it has got is the best in only the local optimal solution.

The Travelling Salesman Problem (TSP) [2] is one of the most known combinatorial optimization problems. These problems require very usually a colossal amount of computing resources to be solved efficiently because the solutions space that should be explored gains exponentially when the size of the problem increases[3].

In this paper, in order to overcome the disadvantages of genetic algorithm and greedy algorithm, an improved greedy genetic algorithm (IGGA) is proposed. Genetic algorithm is too dependent on the initial population. IGGA improved the greedy algorithm and made it search based on the base point. The experiments show that the algorithm reduced the number of cross-routes, and strengthened the capacity optimization. TSP which is a famous classic NP problem and a typical combinatorial optimization problem is often used to verify the validity of intelligent optimization algorithm. Through solving the TSP problem, and compared with other algorithms, it demonstrates that IGGA is a feasible and effective algorithm to solve complex combinatorial optimization problems.

II. 2. MAIN PROCESS OF GA AND GREEDY ALGORITHM

A. The main principle of GA[3]

●Selection. Selection is to evaluate each individual and keeps only the fittest ones among them. In addition to those fittest individuals, some less fit ones could be selected according to a small probability. The others are removed from the current population.

●Crossover. The crossover recombines two individuals to have new ones which might have a better performance. Formula (1) shows the single-point crossover of the binary-coded genetic algorithm as follows.

$$\begin{aligned} &101 \ {}_{\Delta}\underline{101} \rightarrow 101\ \underline{010} \\ &010 \ {}_{\Delta}\underline{010} \rightarrow 010\ \underline{101} \end{aligned} \tag{1}$$

●Mutation. The mutation operator induces changes in a small number of chromosomes units. Its purpose is to maintain the population diversified enough during the optimization process. Formula (2) shows the mutation of the binary-coded genetic algorithm as follows.

$$101\underline{1}10 \rightarrow 101\underline{0}10 \tag{2}$$

B. The basic idea of greedy algorithm

From the initial solution to a successive approximation solution of one problem, it achieves as quickly as possible to a better solution. When the algorithm cannot continue to move forward, the algorithm stops. The algorithm has the following questions: (1) it cannot guarantee that the final solution is the best; (2)it cannot be applied to seek the maximum or minimum solution of the problems;(3) it can only seek to the scope of the feasible solution which satisfy certain constraints.

III. THE IMPROVED GREEDY GENETIC ALGORITHM

A. The basic idea of the proposed algorithm

This paper presents an improved greedy genetic algorithm (IGGA), which makes greedy algorithm search for a one-

978-0-7695-3736-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ICNC.2009.504

way based on the base point. In this way, it can improve the traditional greedy algorithm, strengthen its ability of global optimization while ensuring that their advantage in terms of speed, generate more good gene fragments at the same time, and speed up the generation of good population. Through the experiments, in solving the CHN31, the average route length of the initial population is between $4*10^4 \sim 5*10^4$,which comes from the initial population that generated randomly ,while it is between $2*10^4 \sim 2.4*10^4$,which comes from the initial population generated by the greedy algorithm based on the base point.However,the best solution is between $1.5*10^4 \sim 1.6*10^4$ .In addition, the genetic algorithm in the early evolution of population should be conducted to ensure that the search is overall in order to avoid premature convergence. At the evolution of the late, while approaching the optimal solution, the search should be focusing on the local scope but not the global scope, as far as possible, to improve the accuracy. Therefore, this paper quoted the self-adaptive crossover and mutation strategies [4].

$$P_c = \begin{cases} P_{c\_\max} - \left(\dfrac{(P_{c\_\max} - P_{c\_\min})}{it\max}\right) * iter......f' > f_{avg} \\ P_{c\_\max}...f' <= f_{avg} \end{cases} \qquad (3)$$

$$P_m = \begin{cases} P_{m\_\min} + \left(\dfrac{(P_{m\_\max} - P_{m\_\min})}{it\max}\right) * iter......f > f_{avg} \\ P_{m\_\min}...f <= f_{avg} \end{cases} \qquad (4)$$

P_c : The crossover probability

$P_{c\_\max}$: The maximal cross probability

$P_{c\_\min}$: The minimal cross probability

P_m : *The mutation probability*

$P_{m\_\max}$: The maximal mutation probability

$P_{m\_\min}$: The minimal mutation probability

$Iter_{\max}$: The maximal generation

$Iter$: The current generation

f_{avg} : The average fitness of the species

f' : The fitness of one of the parent

f : The fitness of the one for mutation

In addition, in order to further avoid the bad influence of greedy algorithm in the process of the local optimization, this paper adds the 2-opt algorithm.

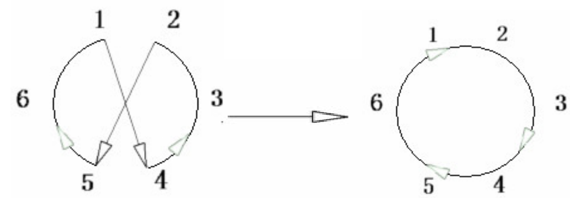


Fig. 1 The 2-opt algorithm

B. The greedy algorithm based on one-way and the base point

The improved greedy algorithm starts from the base point and then searches in accordance with the rules.

The standard of selecting the base point: Choosing the largest city (or the smallest) among the abscissa of the cities. Search rules: Followed by the size of the order in accordance with the abscissa to search the nearest city and to put it into the tour while ensuring the new length of the tour is the shortest. Through this way, it can greatly reduce the number of cross-routes, which were just because of the choice of the random initial points. And it can reduce the length of the tour, and reduce the computation greatly .It is the same as the ordinate. In this paper we choose the point, whose abscissa is the smallest, as the base point, and then find two points which are nearer to the base point, and constitute a triangle with the three points.

Then follow the search rules, satisfy the following formula, make the‘d’ is the minimal [5]:

$$d = \begin{cases} d(f_j, f_{k+1}) + d(f_j + f_k) - d(f_k, f_{k+1}) \\ d(f_j, f_{k+1}) + d(f_j + f_k) \end{cases} \qquad (5)$$

J: The current city to insert

K: The city in front of J in the tour

K-1: The city behind J in the tour

$d(f_j + f_k)$: *The distance between City J and City K*

For a more intuitive description of the advantages of the proposed algorithm, the following figures show us the comparison.

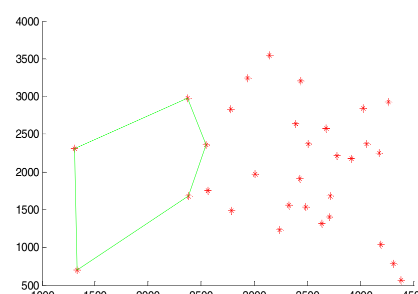


Fig.2 The route with five cities

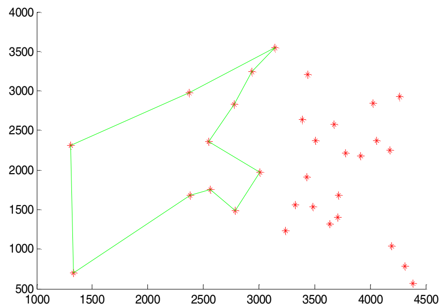


Fig.3 The route with eleven cities

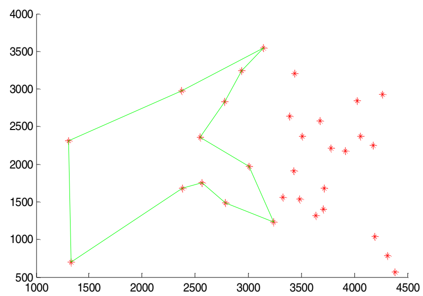


Fig.4 The route with twelve cities

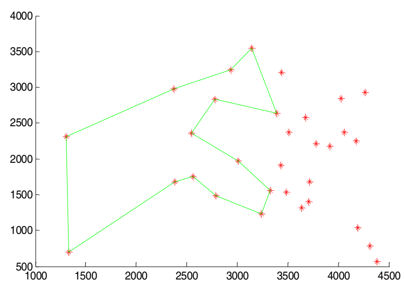


Fig.5 The route with fourteen cities

Fig.2~Fig.5 show the evolution of the greedy algorithm based on one-way and the base point. Fig.6~Fig.9 show the evolution of the traditional greedy algorithm

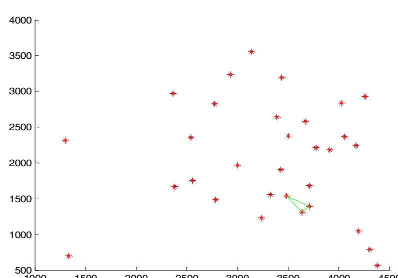


Fig.6 The route with three cities

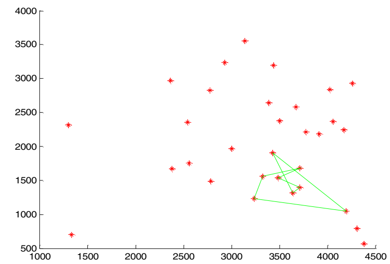


Fig.7 The route with eight cities

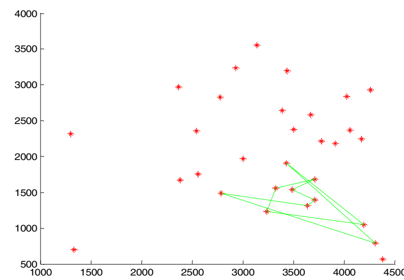


Fig.8 The route with ten cities

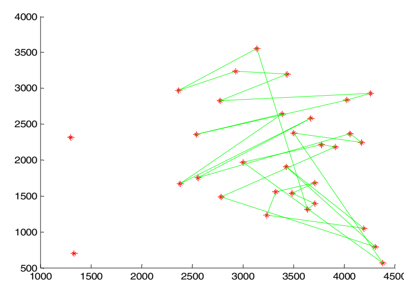


Fig. 9 The route with twenty-nine cities

Through the comparison, greedy algorithm based on base point is very effective. In addition, greedy algorithm based on the base point, can generate better gene fragments during the search, which will contribute to accelerate the speed of search, and will save the search time.

C. The process of our proposed hybrid improved algorithm for solving TSP can be described as follows.

The process of the proposed IGGA for solving TSP can be described as follows:

Step1. Initialization of parameters: Set the current number of iteration *iter*=1; set the maximum number of iteration *itmax=500*; set the number of population as *Nn*, and the number of selected population as *Nq;* set $P_{c\_\max}=0.9, P_{c\_\min}=0.8; P_{m\_\max}=0.8, P_{m\_\min}=0.4$

Step2. Generate initial population. Through the improved greedy algorithm based on the base point and one-way, it gives a solution, as well as the city route α. Then use the 2-opt algorithm to optimize the route of cross –route, just because greedy algorithm is easy to fall into the local optimum.Repeat *Nn* times. Then we get the population with Nn individuals.

Step3. Choose the better population.In order to reduce the computation, choose the excellent population from all. And then we get the new population with *Nq* individuals.

Step4. The introduction of new population .In order to avoid falling into the local optimum, we need introduce new population. The new population, which is with *Nq* individuals, are chosen from randomly generated population with Nn individual. Up to now, we get the population we want, which is with *Nq**2 individuals.

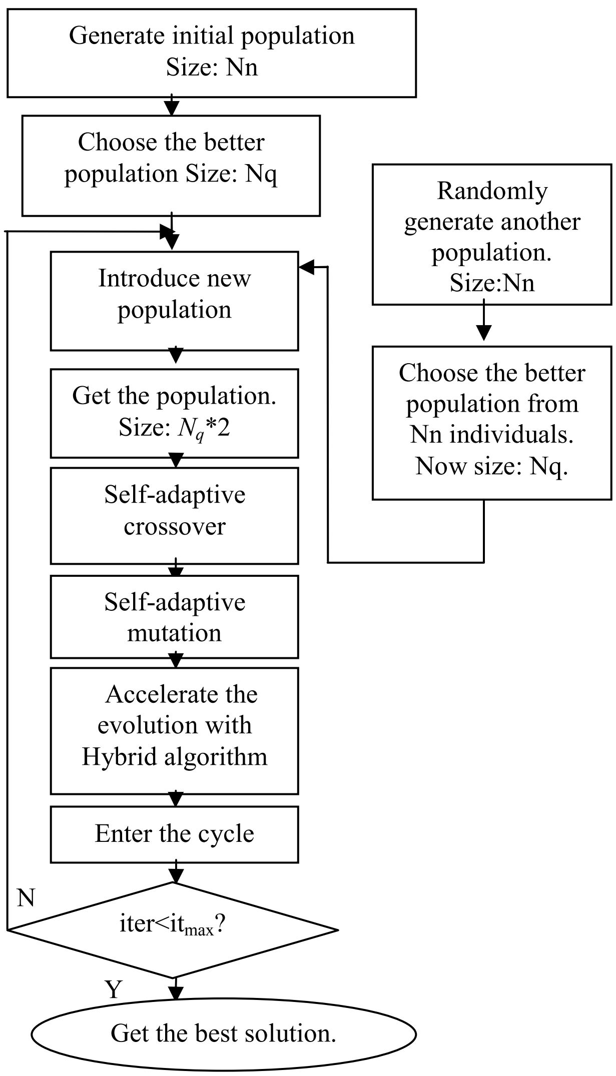


Fig.10 Flow chart of the proposed IGGA

Step5. Self-adaptive crossover. After the crossover process finished, we choose the better ones from parents and their offspring.

Step6. Self-adaptive mutation. After the mutation, we choose the better ones from parents and their offspring.

Step7. Accelerate the evolution with hybrid algorithms. Firstly, we exchange two cities every time, and then choose the better population to substitute for the original. Then select randomly fragment and reverse it.

Step8. Enter the cycle. Repeat step3, until *iter=itmax* or other defined termination condition is satisfied.

Step9. Output the best tour and the shortest length. The above process can also be described with Fig 10.

IV. EXPERIMENTAL RESULTS

A. *Optimal route and the average evolution curve*

To verify the superiority of this algorithm，this paper selected 31 provincial capital cities and municipalities to carry out tests to verify IGGA. Experiments show that the optimal route length obtained by IGGA is 153825443.
The sub-optimal length: 153825443.

The optimal route: 1 15 14 12 13 7 10 9 8 2 4 16 5 6 11 23 19 17 3 18 22 21 20 24 25 26 28 27 30 31 29.

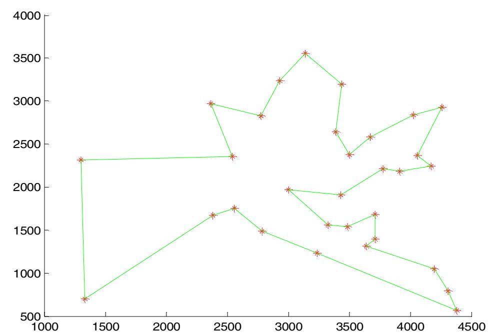


Fig.11 The optimal route by IGGA

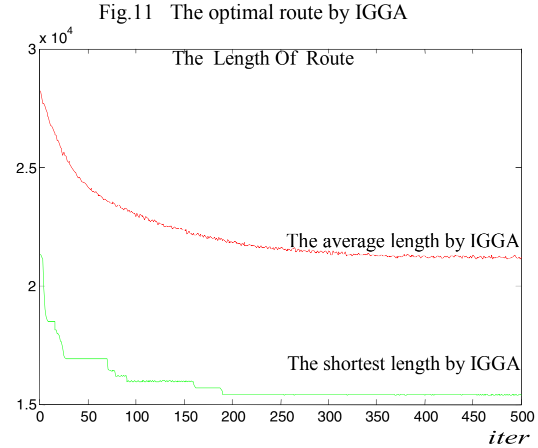
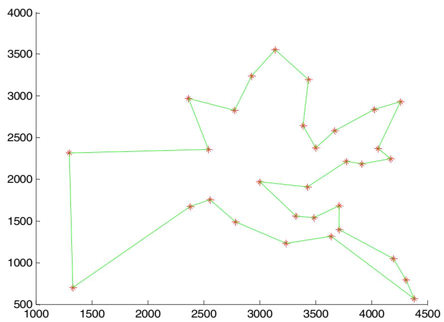


Fig.12 The evolutionary Curve of optimal route by IGGA

B. *The sub-optimal solution of length by IGGA*

The sub-optimal length: 15393.48478.

The sub-optimal route: 1 15 14 12 13 7 2 10 9 8 4 16 5 6 11 23 19 17 3 18 22 21 20 24 25 26 28 27 30 31 29



Figre.13 The sub-optimal route by IGGA

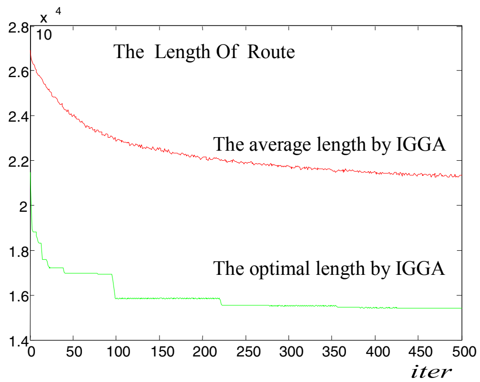


Fig.14 The evolutionary Curve of sub-optimal route by IGGA

C. *Experimental comparison with other algorithms in solving the CHN31*

Table.1 The name and length of other algorithms

Name of the algorithm	**The length of the optimal route**
Standard Genetic Algorithm[6]	15387
Geometric Algorithms[7]	15492
Genetic Algorithm with combinations of the ameliorated OX operator, greed recessive variation, and combined variation[8]	15404
A filled function method[9]	15404
The geometric region-divided method[10]	15404
Man-machine combination optimization method - quality exhaustive method[11]	15499
In the Hopfield neural network methods, together with the constraint condition of the triangle about both sides with in place of one side[11]	16262
A Heuristic greedy method[12]	15409
Simulated annealing algorithm[6]	15398

It is obvious that IGGA can find the best solution than the standard Genetic Algorithm and many other algorithms. IGGA has better convergence, and has a stronger ability of global optimization.

V. Conclusions

This paper proposed an improved greedy genetic algorithm for solving combinatorial optimization problem, and this algorithm can also solve other complex optimization problems effectively.

Acknowledgment

This work was supported by Natural Science Foundation of China under grant #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039, “Beijing NOVA Program” Foundation of China under grant #2007A0017 and “Science Research Training Program” Foundation of Ministry of Education of China. The authors want to show thanks to the anonymous referees for their valuable comments and suggestions.

References

[1] Holland J., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press,Ann Arbor, MI,1975.

[2] Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., et al., *The Traveling Salesman Problem*, John Wiley & Sons, Chichester,1985.

[3] Hichem Talbi, Amer Draa, Mohamed Batouche, “A New Quantum-Inspired Genetic Algorithm for Solving the Travelling Salesman Problem”. *2004 IEEE International Conference on Industrial Technology*, 2004, Vol.3, pp.1192-1197.

[4] Shisanu T, Prabhas C. “Parallel genetic algorithm with parameter adaptation”, *Information Processing Letters*, 2002, Vol.82, No.1, pp.47-54.

[5] Jingjun Zhang, Yuzhen Dong, Ruizhen Gao, Yanmin Shang. “An Improved Genetic Algorithm Based on Fixed Point Theory for Function Optimization”, *2009 International Conference on Computer Engineering and Technology*,2009,Vol.1, pp.481-484.

[6] Yiwen Zhong,Zhengyuan Ning,Yinghua Cai, et al. “An Improved Discrete Particle Swarm Optimization Algorithm”,*Mini-Micro systems*, 2006, Vol.27 , No.10 , pp.1893-1896.

[7] Peide Zhou. “A Geometrical Algorithm for Solving the Travelling Salesman Problem”, J*ournal of Beijing Institute of Technology*, 1995, Vol.15, No.1, pp.97-99.

[8] Chunsong Feng, Junyu Wang, Songsheng Zhou, et al. “Ameliorative Genetic Algorithm of TSP”,*Journal of Wuhan University of Technlogy*,2006, Vol.28, No.4, pp.116-118,130.

[9] Wenxing Zhu, Qingxiang Fu. “A Filled Function Method for the Traveling Salesman Problem”, *Chinese J.Computers*, 2002, Vol.25, No.7, pp.701-707.

[10] Mutian Chen, Hexi Cai. “The Geometric Region-Divided Method for Solving the China TSP Problem”, *Computer Engineering&Science*,1998, Vol.20, No.1, pp.22-27.

[11] Fan Jin, Junbo Fan, Yongdong Tan. *Neural networks and Neural Computers,* Southwest Jiaotong University Press, Chengdu,China, July,1991.

[12] Lideng Pan, Xiaofeng Huang. “A Heuristic Greedy Method for the Traveling Salesman Problem”, *Journal of Beijing University of Chemical Technology*, 1998, Vol.25, No.2, pp.46-51.