Open Journal of Social Sciences, 2016, 4, 303-308
Published Online March 2016 in SciRes. http://www.scirp.org/journal/jss
http://dx.doi.org/10.4236/jss.2016.43037

A Neighborhood Expansion Tabu Search Algorithm Based On Genetic Factors

Dan Wang, Haitao Xiong, Deying Fang

School of Computer and Information Engineering, Beijing Technology and Business University,
Beijing, China
Email: wangdanzi@126.com

Received 5 March 2016; accepted 28 March 2016; published 31 March 2016

Abstract

We provide an improved algorithm called "a neighborhood expansion tabu search algorithm based on genetic factors" (NETS) to solve traveling salesman problem. The algorithm keeps the traditional tabu algorithm's neighborhood, ensure the algorithm's strong climbing ability and go to the local optimization. At the same time, introduce the genetic algorithm's genetic factor (crossover factor and variation factor) to develop new search space for bounded domain. It can avoid the defects of the alternate search. The results show that this optimization algorithm has improved a lot in the target of "target value", "convergence" and compared with traditional tabu algorithm and genetic algorithm.

Keywords

Traveling Salesman Problem, Genetic Algorithm, Tabu Algorithm, Genetic Factors

1. Introduction

Traveling salesman problem (TSP) is one of the most classic problems in traffic path optimization. It is a minimum path cost of road planning problem when a single traveler start from a point, through all the requirements points only once, finally return to the origin.

TSP has a lot of applications in reality, such as computer networking, electronic maps, traffic induction, electrical wiring, VLSI layout, ATM packet-switched networks [1]. Thus, it's often regarded as a classic case due to its important theoretical and practical value. The algorithm to solve TSP contains a lot, And we can divided them into the traditional algorithm and intelligent algorithm. Traditional algorithms including dynamic programming, branch threshold method and so on. This kind of traditional algorithm idea is simple and it has a good effect in the small size TSP. But as the problem size is bigger, the difficulty of solving TSP growing exponentially. So we'll use heuristic algorithm. Heuristic algorithms can be divided into stereotypes and modified algorithm. Stereotype algorithms include nearest neighbors, greedy method, and insertion method. These methods can be generated an approximate optimal solution from distance matrix. They are simple and efficient, but

How to cite this paper: Wang, D., Xiong, H.T. and Fang, D.Y. (2016) A Neighborhood Expansion Tabu Search Algorithm Based On Genetic Factors. *Open Journal of Social Sciences*, **4**, 303-308. http://dx.doi.org/10.4236/jss.2016.43037

its solution quality is poor relatively. Modified algorithms include local search algorithms and intelligent optimization algorithms. These methods can improve a viable path to find a better solution continuously. Guo's algorithm [2] IN-Over operators perform simply and converges fast, but it's easy to fall into local optimum later. Artificial neural network algorithm [3] has a strong adaptive ability and fault tolerance, but it depends on mechanism greatly. Simulated annealing algorithm [4] can achieve parallel operation, find out the global optimal solution for large scale combinatorial optimization problem, but it's sensitive to initial parameters. Tabu search algorithm [5] is of better convergence and higher retrieval efficiency. But it is dependent on the initial solution and the neighborhood space intensively. Ant colony algorithm [6] initial path can be arbitrary and it has strong concurrency and robustness, but the scope of the pheromone concentration limits the convergence of the algorithm drastically. At present, TSP focus on the combination of a variety of optimization mechanism and adjacent domain search structure design of hybrid heuristic algorithm [7],which expand the applicable domain of algorithm and improve the effect of global optimization [8]. Our paper, using genetic factor expand the neighborhood of tabu algorithm, find a high efficiency and strong convergence of TSP planning scheme.

2. Algorithm Design Ideas

2.1. Tabu Search Algorithm

Tabu search [9], created by Fred W. Glover in 1986, is a metaheuristic search method employing local search methods used for mathematical optimization. It imitates human memory, starts from an initial feasible solution, chooses a series of specific search direction (neighborhood space) as a temptation, and uses tabu list storage the area just has searched to avoid detour search. At the same time, it forgives some good condition of the area of the tabu list to ensure the diversity of the search, Thus it makes the specific objective function value moving constantly. So it has strong local search ability.

Tabu search algorithm only search in a designated adjacent neighborhood space. Speed, therefore, is a big advantage, but it is easily trapped into local optimization. So it is an important step to master the neighborhood space for obtaining the target. Moreover, the traditional scatter search strategy of tabu search will jump to the sub-optimal solution's search space when current optimal solution keep certain search times, which waste a lot of time and reduce the scatter search ability. Tabu search has only one initial solution in the search process. So it has a strong dependence to the initial solution. A good initial solution might result a good convergence, a poor initial solution may be bad conversely. Below shows the pseudo code of tabu search:

```
Begin
    Set the parameters; $i = 1$;
    Initialize a path at random
    clear up the tabu list
    while condition is not satisfied Do
        Calculate the so_far_bestpath's neighborhood;
        Bestpath; so_far_bestpath;
        update tabu list;
        $i = i + 1$;
End
```

2.2. Genetic Algorithm

Genetic algorithm [10], a powerful tool to solve the arbitrary function optimization problem, is developed by Holland, who is an American scholar, and his colleagues. Genetic algorithm is a random search algorithm, using the reference of natural selection and genetics theory in biosphere. Genetic algorithm simulated the natural genetic process through selection, mutation and crossover. At each iteration, a set of candidate solutions is retained, and we'll generate better candidate solutions group with genetic operators and repeat until convergence occurs.

Genetic algorithm, belong to group search, is easy to parallel processing. And heuristic search is not exhaustive blindly, but enhance the global search ability. But the probability of crossover and mutation operator in genetic algorithm is small, it lead to the limitation of search space and weakness of climbing ability. When the number of groups growing, the target effect worse and worse, time consuming longer and longer. All these can't satisfy the demand of reality. Below shows the pseudo code of genetic algorithm:

```
Begin
    Set the parameters; i = 1;
    Initialize the population (pop);
        while condition is not satisfied Do
            Calculate the length of each solution;
            shortPath;
            pop = selection (pop);
            pop = crossover (pop);
            pop = mutation (pop);
            i = i + 1
End
```

2.3. A neighborhood Expansion Tabu Search Algorithm Based on Genetic Factors

The application scope is restricted although GA and TS have their own successful application in the reality [11]. So this paper put forward an improving algorithm called "neighborhood expansion tabu search algorithm based on genetic factors" (NETS). It will introduce the crossover and mutation factors of GA into TS to construct a new neighborhood, which makes the decentralized strategy and focus strategy of TS operate simultaneously.

NETS will conduct an optimization of standard neighborhood (*i.e.*, a mountain climbing) first, only the selected neighborhood will carry out crossover and mutation. It can guide the extended neighborhood toward an optimum solution. Then NETS will conduct another optimization of final neighborhood (including optimized and extended neighborhood). Where there is a climbing, there is a search. It's a decentralized strategy that recorded the optimal value and improved efficiency. Below shows the pseudo code of process of NETS optimization neighborhood:

```
Begin
    Calculate the so_far_bestpath's neighborhood;
    Sort and Prefer the so_far_bestpath's neighborhood;
    pop = prefer neighborhood;
    pop1 = crossover (pop);
    pop2 = mutation (pop);
    neighborhood = pop + pop1 + pop2;
End
```

We introduce the crossover and mutation factors of GA to extend the neighborhood. And Partially Mapped Crossover (PMC) [12] [13] will cross an optimal neighborhood and a randomly generated neighborhood. We generate a pair of paths in the two cross position randomly, and define the area between the two nodes as mapping area. Then exchange the two mapping area with each path. Assuming the path of the cross position is as follows:

When exchange the mapping area between A and B, we can get:

Obviously, there are same node in A' and B', for example, 3 has appeared two times in A', At this point, we can exchange A' and B's node outside the mapping area according to the location of the mapping area node, we can get:

In addition to crossover, 3 methods are used to variation: 2-opt operation, shift operation and exchange operation [14].

1) 2-opt operation

Select a node as a starting point, then find another nearest node and reverse the nodes between the two points. Shown in **Figure 2**, node 3 is the nearest point of node 4, the nodes between them will reverse.

2) shift operation

Find a star point and a length, extract these nodes, and then put the segment after the rest nearest node. As shown below, the node is 3 and the length is 3, node 4 is the rest nearest node, the extracted segment is put after node 4.

3) exchange operation

Select one node, exchange the position of two segments before and after the node. As shown below, select node 9, change the two segments before and after node 9.

The neighborhood got great extension when introduce the crossover and mutation operator. NETS has strong dispersion and improve the efficiency of the decentralized search. **Figure 1** shows the process of the NETS.

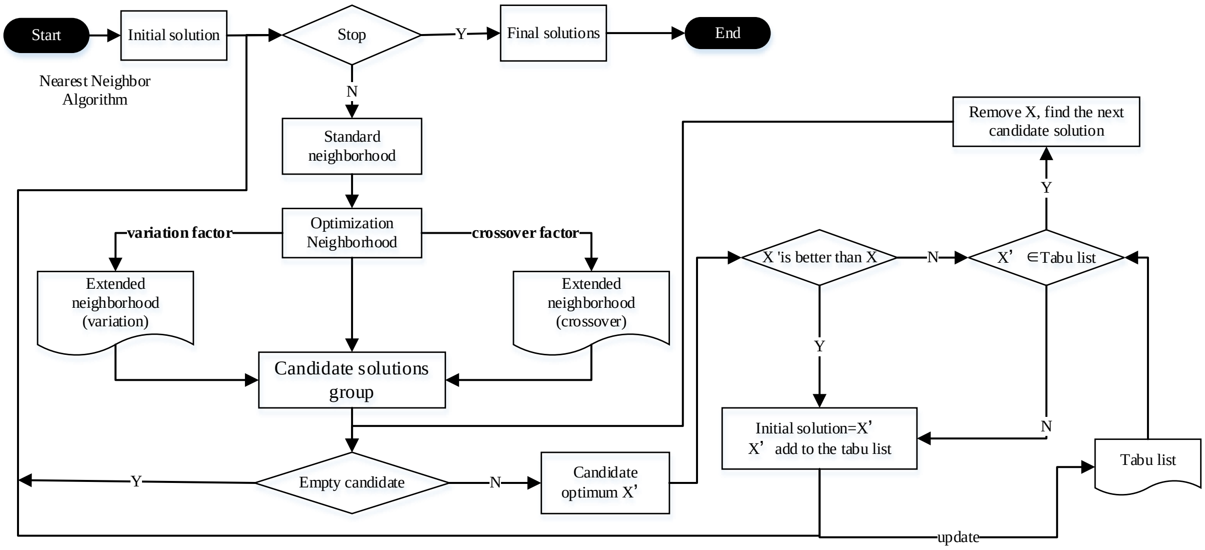

Figure 1. NETS execution process for TSP.

3. Experiments and Analysis

3.1. Experimental Environments

The algorithm described in this paper was coded in Matlab2012b, and all experiments were run on Windows 7 operating system with 6GB memory and Intel Core (TM) i3 CPU M380.

3.2. Experimental Data

The experimental data of this paper is come from standard symmetric TSP test problem sets of TSPLIB (Reinelt, 1991) (available at http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/). The data scale is 14, 16, 22, 29, 48 and 52. We compared the traditional TS, GA and NETS in target value, convergence and time consuming.

3.3. Performance Testing

3.3.1. Target Value

There is no exact solution for that TSP belongs to NP problem. We carry out 20 times experiments for each algorithm. The results are shown in **Table 1**. "Best" represents the best target value and "Avg" represents the average of 20 runs.

Experimental results show that when the number of nodes is small, the best target value of NETS is the same as traditional TS and GA, but the average of NETS is better than TS and GA even if the difference is less than 1%. With the increase of the number of nodes, the target value of NETS is superior to TS and GA significantly. Especially, when the data sets reach to 48 and 52, it regards as failure for GA because the target value deviates too much from our expectations. And the average value of NETS is increased by 6% and 8.6% than TS respectively. Thus, NETS has a great improvement in the target value of the solution compared with TS and GA.

3.3.2. Convergence Speed

The search process of NETS, TS and GA is almost the same in data set "Burma14". Thus, we'll display the search process of the optimal solution in one graph of remaining six sets of data with different algorithms (GA is valid only in the four set of data).

We can find from the **Figure 2** that NETS reaches the target value quickly and it is the fastest in convergence. Next is the GA algorithm, and the traditional TS algorithm is slowest. With the increase of the number of nodes, the target value of NETS is better than TS and GA in the same number of iterations. When the number of nodes is less than 30, NETS will close to the optimal target value in 10 iterations, GA needs 30 iterations and TS needs

Table 1. Target value comparison on different data sets.

Data sets	NETS		TS		GA	
	Best	Avg	Best	Avg	Best	Avg
Burma14	30.87	30.87	30.87	30.87	30.87	30.87
Ulysses16	73.98	73.98	73.98	73.98	73.98	73.98
Ulysses22	75.3	76.28	75.3	77.38	75.3	76.46
Bays29	9076	9172	9349.45	9491.2	9926.4	10065.3
Dantzig42	686.88	720.56	743.7	802.55	805.1	886.6
Att48	33523	35159	37034	38481.6	-	-
Berlin52	7544	8032	8429.23	8701.27	-	-

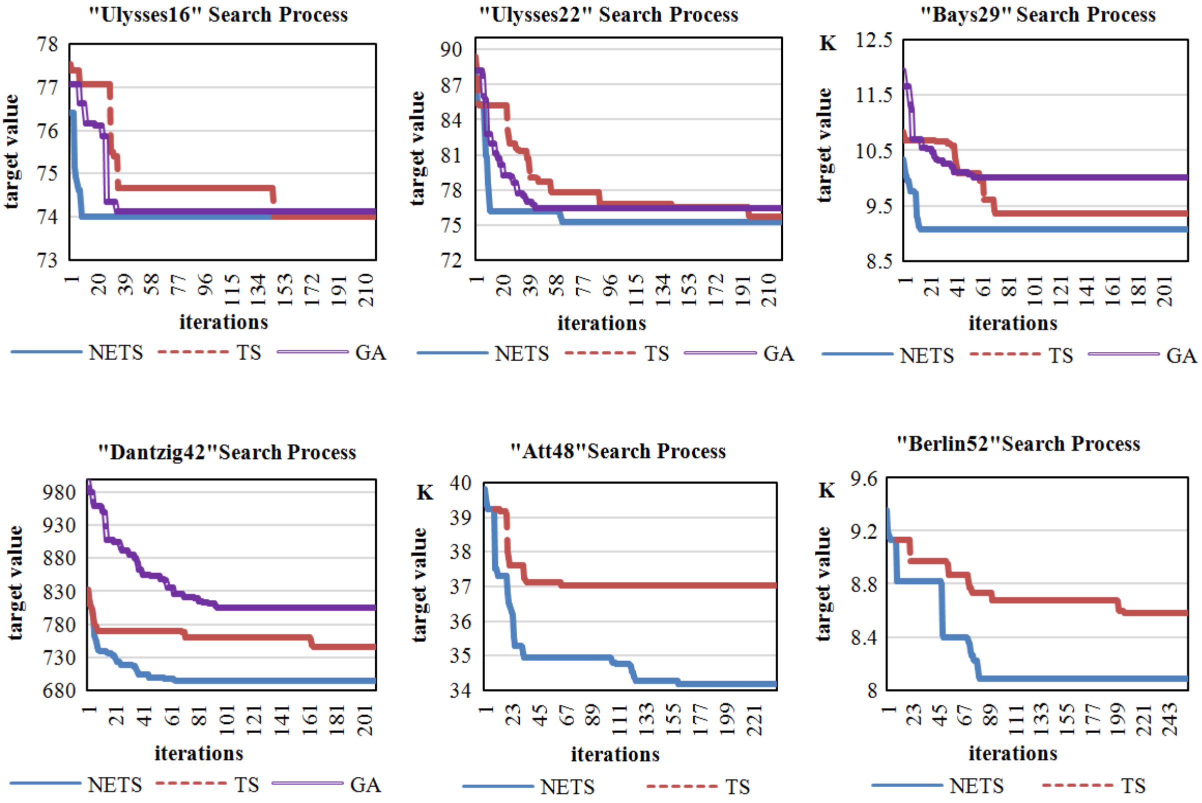

Figure 2. Search process comparison on different data sets.

60 iterations. When the number of nodes is greater than 30, convergence of the 3 algorithms are all slow down. NETS and GA are close to their optimal target values after 100 iterations, but TS needs 160 iterations and is easy to fall into local optimum. However, the target value of the GA has deviated from the other two algorithms, which is obviously invalid. In conclusion, the convergence speed of NETS is improved greatly compared with the traditional TS and GA.

3.3.3. Time Consuming

NETS need to expand the neighborhood, so there are two selecting process. The first time is to optimize the basic neighborhood. The second time is to optimize the preferred neighborhood after introduce crossover and mutation factors. Thus, the NETS add iteration operation of optimization, crossover and mutation. When the number of node is small, the time consuming is almost the same, but when the number of the nodes increases, NETS will consume time more 20% than TS. This is the next part need to optimize.

4. Summary

In this paper, we design an improving algorithm called “neighborhood expansion tabu search algorithm based on genetic factors” to solve the TSP problem. NETS keep the strong search ability of TS, which guide to a quick and accurate search speed. Then it expends the neighborhood of the initial solution by introducing genetic operators of GA, which make the scattered search and focus search simultaneously. NETS reduce the probability of local optimal search and improve the convergence speed greatly. Compared with the standard TS and GA, with the increase of the number of nodes, the target value of NETS is more superior. And the convergence rate of NETS is faster than other algorithms at the same node. Since the research involved only distance but with no fuel, congestion, so we’ll take more factors into the consideration to achieve maximum benefit with minimal cost next step.

Acknowledgements

This research was supported by the Scientific Research Common Program of Beijing Municipal Commission of Education under Grant KM201310011009, the National Natural Science Foundation of China under Grant 71201004, the Key Program of Beijing Social Science Foundation under Grant 14JGA013, and the Improvement Program of Postgraduate Scientifical Capability of Beijing Technology and Business University.

References

[1] Gao, H.C., Feng, B.Q. and Zhu, L. (2006) Reviews of the Meta-Heuristic Algorithms for TSP. *Control and Decision*, **21**, 241-247.

[2] Li, Y. and Kang, Z. (2000) Guo’s Algorithm and Its Application. *Journal of Wuhan University of Technology*, **22**, 101-104.

[3] Jin, L., Guo, Z.H. and Zheng, C.Y. (2014) Study on Improved Method of Neural Network to Solve TSP. *Computer Simulation*, **31**, 355-358.

[4] Xie, Y. (1998) A Summary on the Simulated Annealing Algorithm. *Application Research of Computers*, **15**, 7-9.

[5] Li, Y.X. and Peng, G. (2010) TSP Problem Based on Tabu Search. Guangxi Computer Society. *Proceedings of the* 2010 *Academic Annual Conference of Guangxi Computer Society*, 5.

[6] Guo, P. and Yan, W.J. (2007) The Review of Ant Colony Algorithm Based on TSP. *Computer Science*, **10**, 181-184+194.

[7] Wang, L. and Zheng, D.Z. (2000) Meta-Heuristic Algorithm: A Review. *Control and Decision*, **3**, 257-262.

[8] Yu, L. and Lu, F. (2014) A Hybrid Algorithm for Traveling Salesman Problem in Road Network. *Acta Geodaetica et Cartographica Sinica*, **43**, 1197-1203.

[9] Glover, F. (1986) Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, **13**, 533-549.

[10] Holland, J.H. (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1992.

[11] Li, D.W., Wang, L. and Wang, M.G. (1998) Genetic Algorithm And Tabu Search: A Hybrid Strategy. *Journal of Systems Engineering*, **3**, 30-36.

[12] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading.

[13] Torr, P.H.S. and Murray, D.W. (1997) The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *International journal of computer vision*, **24**, 271-300. http://dx.doi.org/10.1023/A:1007927408552

[14] Zhang, Z., Qin, H., Zhu, W., *et al.* (2012) The Single Vehicle Routing Problem with Toll-by-Weight Scheme: A Branch-and-Bound Approach. *European Journal of Operational Research*, **220**, 295-304. http://dx.doi.org/10.1016/j.ejor.2012.01.035