The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Genetic Algorithm with Cross Paths Detection for Solving Traveling Salesman Problems

Shi-Jim Yen[1], Shih-Yuan Chiu[1] and Sheng-Ta Hsieh[2]

[1] Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, R.O.C.
[2] Department of Communication Engineering, Oriental Institute of Technology, New Taipei City, Taiwan, R.O.C.

fo013@mail.oit.edu.tw

Abstract: The travelling salesman problem (TSP) is a classic optimization whose goal is to find a shortest route path. In this paper, an improved GA is proposed to solving TSP. For eliminating repeated chromosomes and increasing their diversity, a normalization strategy is proposed. Further, for improving solution searching efficiency, the cross paths detection is also proposed to reduce paths distance. Experiments were conducted on 10 instances of TSPLIB. The results showed that the proposed method exhibits better performance when solving these TSP instances compared to related TSP approaches.

Keywords: cross path detection, genetic algorithm, optimization, traveling salesman problem.

1 INTRODUCTION

The travelling salesman problem (TSP) is a classical NP- hard problem in the field of computer science. In TSP, a set of cities' location will be given and the goal of the problem is try to find a shortest route for visit each city only once.

TSP is used as a benchmark for testing performance of optimization methods. Although there are so many contribution in enhance solution searching ability. Some of them might already be surpassed by novel approaches. In 2007, Shi et al. proposed a particle swarm optimization (PSO) based approach called Discrete PSO [1] for solving TSP. In 2008, Kaur and Murugappan proposed a hybrid genetic algorithm [2]. It combine the nearest neighbor algorithm (NNA) and pure genetic algorithm (GA) for deriving better solutions.

In order to improving optimizer's searching ability and efficiency, this paper presents improved GA to solving TSP. The proposed method is incorporated with the cross path detection to verify and eliminate redundant path for minimizing route solution. The proposed method also involved a normalization strategy to increase diversity of population, which can prevent the solutions from falling into the local optimum and enhance the proposed approach's searching ability.

2 GENETIC ALGORITHM

The basic principles of genetic algorithm (GA) were first introduced by John Holland in 1975[3]. Holland's GA was the first evolutionary computation (EC) paradigm developed and applied. It is for moving from one population of *chromosomes* to a new population by using a kind of Darwin's "natural selection" principle together with the genetics- inspired operators of selection, cross-over, mutation, and inversion [4]. A brief introduction of genetic algorithm will be described.

3 PROPOSED METHOD

Although there are numerous approaches can be applied for solving TSP. Efficiency and robustness of algorithms are still the major issues. In this paper, the nearest neighbor algorithm (NNA) is adopted for initial population. In order to reduce route distance efficiently; the cross paths detection is introduced. Further, the normalization is also involved to keep diversity of population.

3.1 Cross-over

In general, cross-over operation in GA is to generate new offspring. It combines two chromosomes' information which was also generated by chromosomes in a previous generation and evaluated by the cost function. Finally, the better chromosomes will be kept is the population. If a chromosome discovers a new probable solution, its offspring will move closer to it to explore around the region deeply in following cross-over process.

3.2. Mutation

In this paper, mutation is to pick up two dimensions randomly. The chromosome's sequence between the two dimensions will be inversed. For example, there are two randomly selected dimension d_1 and d_2 in a chromosome. The contents (route sequence) between d_1 and $d_{2\,2}$will be re-arrangement.

3.3. Cross Path Detection

The goal of TSP is to find a route as short as possible.

Cross paths in route will serious increase distance in TSP. Although, evolution process of GA will find potential optimal solution. Unfortunately; it still may not be able to avoid such situation. In this paper, a geometry-based method called cross path detection is proposed. For current route, each path will be verified that if it cross to another path. These cross paths will then be eliminated.

Let's use symbol "a-b" denoting the path between city-a and city-b. For example, assuming there are 7 cities in TSP. Assuming the route of these cities is connected according to its serial numbers, and the path between city-1 and city-2 is selected for cross path detection. Thus the path 1-2 need to be compared with 3-4, 4-5, 5-6, 6-7. Note that the path 2-3 and 7-1 need not be compared with 1-2. Due to them will never cross to path 1-2. An example of cross path detection is shown in Fig. 1. In this case, the path 2-5 is cross to path 3-6. Thus, the city-2 will be connected to city-3 and city-5 will be connected to city-6.

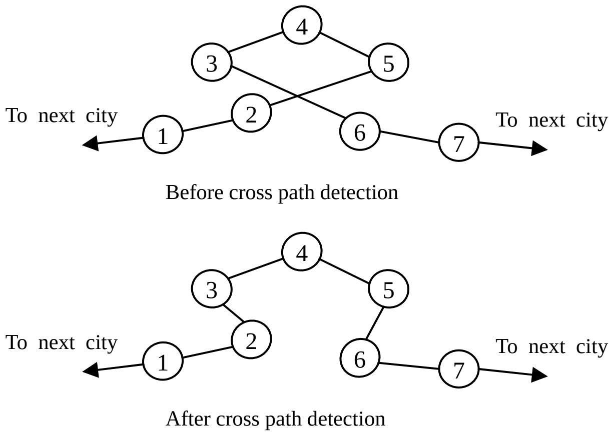

Fig. 1. Example for cross path detection

4 EXPERIMENTS

Ten TSP instances from the TPSLIB [5] were adopted for testing proposed method and compare it to Hybrid Genetic Algorithm [2] and Discrete PSO [1]. The parameters of these methods are set according to their original setting which are listed as following:

- Proposed Method
  The cross-over rate is 1 and the mutation rate is 0.85
- Hybrid Genetic Algorithm
  The cross-over rate is 0.5 and the mutation rate is 0.85
- Discrete PSO
  The inertia weight (*w*) is 1.49 and the accelerate coefficient (c_1 and c_2) are both 0.72.

All the four population-based optimizer were implemented using MATLAB 2010a. The maximal Number of Function Evaluations (FEs) is set to be 50,000 for all approaches. All algorithms were run 30 times and calculated their mean values and standard deviation for the results.

Table 1. Results of Ten TSP Instances

Methods / Instances	Proposed method	Hybrid Genetic Algorithm	Discrete PSO
bayg29	9077 ± 6	9128 ± 70	12416 ± 822
st70	694 ± 4	699 ± 8	1141 ± 90
pr76	112749 ± 1544	112959 ± 1708	176556 ± 12538
rat99	1265 ± 5	1248 ± 14	2117 ± 131
eil101	671 ± 5	675 ± 8	996 ± 42
pr107	44494 ± 77	44683 ± 84	97344 ± 14619
ch130	6539 ± 66	6476 ± 99	11278 ± 792
kroA200	30676 ± 66	31479 ± 461	63030 ± 4124
ts225	132412 ± 1024	132647 ± 1243	209885 ± 10579
a280	2856 ± 23	2893 ± 40	4850 ± 23

Table I presents the mean and standard deviation of 30 runs for proposed method, Hybrid Genetic Algorithm and, Discrete PSO on the ten TSP instances. The best results among the four approaches are shown in bold.

From the results, the proposed method performed with b etter results can be observed. Furthermore, the proposed me thod surpasses all other algorithms in solving instances of b ayg29, st70, pr76, eil101, pr107, kroA200, ts225 and a280. The Hybrid Genetic Algorithm performed better than propo sed method on rat99 and ch130.

5 CONCLUSIONS

In this paper, the proposed method has been presented to solve travelling salesman problem (TSP). The cross path detection was adopted for untie the cross paths and for reducing route distance. Ten TSP instances of TSPLIB were selected for experiments. The experiments results show that the proposed method can find better solutions on these benchmarks than related works.

REFERENCES

[1] X. H. Shi, Y.C. Liang, H. P. Lee, C. Lub and Q. X. Wanga, "Particle swarm optimization-based algorith ms for TSP and generalized TSP," Information Processi ng Letters 103, pp. 169-176, 2007.
[2] D. Kaur and M. M. Murugappan, "Performance En hancement in solving Traveling Salesman Problem usin g Hybrid Genetic Algorithm," Annual Meeting of the North American In Fuzzy Information Processing Socie ty, pp. 1-6, 2008.
[3] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[4] D. E. Gokdberg, Genetic Algorithms in Search, Opt imization & Machine Learning, Addison Wesley, 1989.
[5] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/