ISSN:0975-9646
S N S Kalyan Bharadwaj et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1096-1098

Solving Traveling Salesman Problem Using Hierarchical Clustering and Genetic Algorithm

S N S Kalyan Bharadwaj.B, Krishna Kishore.G, Srinivasa Rao.V

Department of Computer Science and Engineering,
V R Siddhartha Engineering College, Vijayawada, India.

***Abstract*— The Traveling Salesman Problem(TSP) is one of the most intensively studied problem in computational mathematics. To solve this problem a number of algorithms have been developed using genetic algorithms. But these algorithms are not so suitable for solving large-scale TSP. This paper proposes a new solution for TSP using hierarchical clustering and genetic algorithm.**

***Keywords*— Traveling Salesman Problem(TSP),Genetic Algorithm (GA) , Hierarchical Clustering.**

I. INTRODUCTION

Traveling Salesman Problem(TSP) is a NP- Hard problem in Combinatorial Optimization. As early as in 1954, optimal solution to traveling salesman problem with 49 number of cities has been obtained. In 1970's , M. Held and R.M. Karp used minimum spanning tree to solve the TSP with 64 cities. In 1971, M. Bellmore and J. Malone solved TSP using sub-tour elimination .In 1980's, H. Crowder and M.W. Padberg solved the problem with 318 cities using cutting-plane method. In 1991, M. Grötschel and O. Holland, proposed a solution for large scale TSP. D. Applegate , R. Bixby , V. Chvátal and W. Cook proposed solution for TSP using cuts in the years 1998,2001,2004 that solved 13509,15112 ,24978 cities respectively. The solutions worked well up to 5000 cities and can be used up to 33,810 cities.

The solutions for TSP using Genetic Algorithm started in 90's. In 1991, D.Whitley, T.Starkweather and D.Shaner proposed solution for tsp using genetic algorithm in their book " The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination". In 2003, M.Lalena solved TSP using Genetic Algorithms. In 2005, R. Takahashi, Solved the TSP through genetic algorithms with changing crossover operators. In 2007, Ding chao, Cheng ye, He miao proposed a paper for solving another type of TSP called The clustered travelling salesman problem. In 2009, Gohar Vahdati1, Mehdi Yaghoubi Mahdieh Poostchi M. B. Naghibi solved TSP using GA with Heuristic Crossover and mutation operator. In 2009, Jin-Qiu Yang, Jian-Gang Yang, Gen-Lang Chen solved TSP using adaptive clustering[1].In this Paper, the Traveling Salesman Problem is solved using Hierarchical clustering and Genetic algorithm.

II. DEFINITION

In Traveling Salesman Problem(TSP), given a set of cities and the distances between them, we determine the shortest path starting from a given city, passing through all the other cities and returning to the first city. Mathematically, the problem may be stated as follows. Given 'n' cities, and assuming that the distance between city 'i' and city 'j' is $d_{ij,}$ then TSP is given by

$$\min \sum_{i=1}^{n-1} d_{i,\,i+1} + d_{n,1}$$

III. GENETIC ALGORITHM FOR THE TSP

All the existing genetic algorithms proposed for the TSP have the following four steps in common.

A. Encoding

The first step is to choose encoding for the chromosome. In most of the algorithms that solve TSP use permutation encoding. Figure 1, shows a sample chromosome for a problem with 9 cities.

6	9	1	7	4	2	8	3	5

Fig. 1 Chromosome Representation

B. Selection

The selection operator chooses the individual chromosomes to crossover and produce offspring. The selection is made based on the fitness value of the chromosome. Better the individual chromosome's fitness, the higher is the probability of it being selected. Various selection operators used are Roulette-wheel selection, Tournament selection, Rank selection.

C. Crossover

The crossover operator combines two chromosomes (parents) to produce new chromosomes (offspring). The traditional crossover operators produce illegal offspring when representations such as permutation representation are used . So several researchers have made modifications to crossover operators. The commonly used crossover operators for permutation representation are order crossover, cyclic crossover, partially matched crossover.

D. Mutation

Mutation is used to maintain genetic diversity from one generation to next. Generally, mutation is performed at a

smaller rate. It is used to avoid local optima by preventing the chromosomes from becoming more similar to each other. Traditional mutation operators may generate illegal offspring when used on permutation representation. So new mutation operators have been developed. Various mutation operators are insert mutation, swap mutation, scramble mutation.

IV. PROPOSED ALGORITHM

In a genetic algorithm the first step is to create initial population. The population is generated randomly. This strategy works for small number of cities. But as the number of cities increase the fitness values of the chromosomes generated have been low. In our approach, instead of generating the initial population randomly, we create the initial population in 2 steps as explained below.

A. Hierarchical Clustering of Cities

The hierarchical clustering of the data set with 'n' cities is as shown in figure 2. By default ,'n/10' (rounded to nearest decimal) number of clusters are generated.

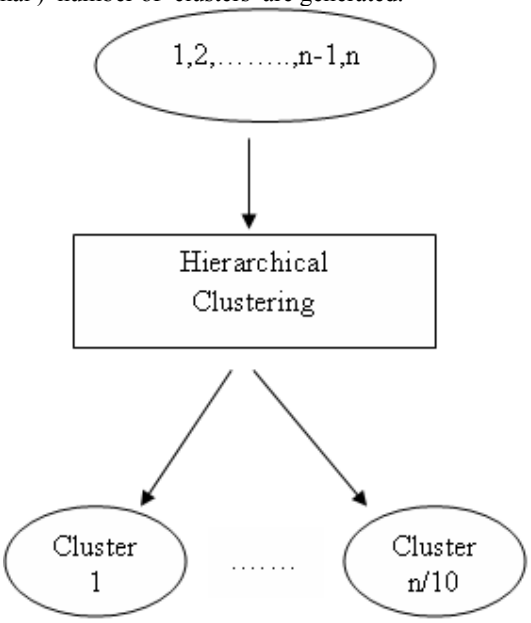

Fig. 2 Hierarchical Clustering of Cities

B. Generating Chromosomes

Suppose that size of the clusters be k1, k2,…., $k_{n/10}$. The sum of the sizes of all the clusters will be equal to 'n', the length of the chromosome to be generated. The chromosome is generated by selecting the first 'k1' genes randomly from cluster 1, the next 'k2' genes randomly from cluster 2, and so on until the selection of last '$k_{n/10}$' genes randomly from cluster 'n/10'. This process is repeated until the entire population of chromosomes is generated. The process of generating chromosome is depicted in figure 3.After generating the initial population using the above procedure, the remaining steps of a simple genetic algorithm can be applied.

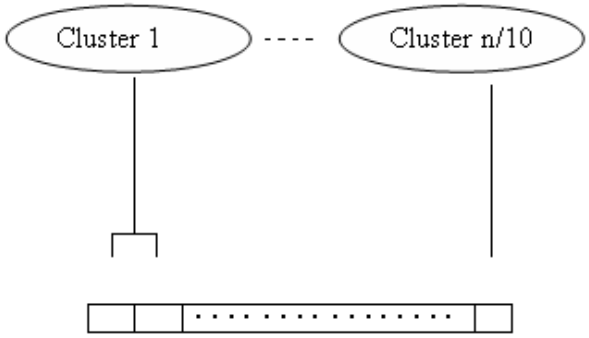

Fig . 3 Generating Chromosome

V. RESULTS AND DISCUSSION

Our algorithm is realized by Java, the hardware is P4 3.06 CPU and 1G memory, and operating system is Windows XP. The parameters chosen for genetic algorithm are permutation representation for encoding, roulette-wheel selection for selection operator, order-crossover for crossover operator, insert mutation for mutation operator. The data files are from the known TSPLIB[7].

TABLE 1

RESULT OF EXPERIMENTS

Data File	Number of cities	Our Algorithm	SGA	Fitness Ratio
att532	532	586350.559	1078037.337	1.838
zi929	929	744229.321	1786326.611	2.401
d1291	1291	454365.752	1367436.564	3.009
pr2392	2392	1053359.214	1.437 E7	13.64

The results show that our algorithm is better than simple genetic algorithm (SGA). As the number of cities increase ,the fitness of best chromosome produced by SGA to our algorithm is increasing which clearly implies that our algorithm is better.

VI. CONCLUSION

Hierarchical Clustering of cities and then generating the initial population using the clustering information is an effective method for solving TSP using genetic algorithm. This idea can be used in other problems solved by genetic algorithm.

REFERENCES

[1] Jin-Qui Yang, Jian-gang Yang, Gen-Lang Chen, "Solving large-scale TSP using adaptive clustering method", 2009 Second International Symposium on Computational Intelligence and Design.

[2] Farshad Farhadnia, "A New Method Based on Genetic Algorithms for Solving Traveling Salesman Problem", 2009 International Conference on Computational Intelligence, Modelling and Simulation.

[3] R. Takahashi, "Solving the traveling salesman problem through genetic algorithms with changing crossover operators," Proceedings of the

Fourth International Conference on Machine Learning and Applications, LosAngeles, CA, USA, pp.319-324, 2005.

[4] D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning", Machine Learning. Addison-Wesley, New York, 1989.

[5] K. Bryant, "Genetic Algorithms and the Traveling Salesman Problem", 2000.

[6] M.Held and R.M.Karp, "The Traveling Salesman Problem and minimum spanning trees: part II", Mathematical Programming 1,6-25

[7] TSPLIB: http://www2.iwr.uni-hiedelberg.de/groups/comopt/software/TSPLIB95/tsp