2012 International Conference on System Engineering and Modeling (ICSEM 2012)
IPCSIT vol. 34 (2012) © (2012) IACSIT Press, Singapore

Solving TSP on the Basis of Grid and Genetic Algorithm

Shirin Nozarian[*a], Yasin Dehghan[b], Majid Vafaei Jahan[b]

[a] Young Researchers Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran

[b] Islamic Azad University,Mashhad Branch, Iran.

[*]snozarian@yahoo.com

Abstract. Traveling salesman problem (TSP) is a problem whose solution is applicable in many engineering problems. And its solution is one of the prioritized research activities among the scientists. According to latest statistical facts Lin–Kernighan method is the best method to solve the problem. This algorithm could solve the problem of more than 1,900,000 cities with 0.3 percent error. Thus a new method is offered in this article in order to solve the traveling salesman problem on the basis of categorizing (known as Grid in this article). It can solve the problem by dividing the map into different regions and finding optimum ways in each region via genetic algorithm. The most important problem in such methods is that combination of optimum regions is not optimum necessarily; so it has been tried to obtain a faster and more accurate response by applying a method to combine the regions and their optimizations. The test results on the basic data in represent the priority of this method comparing with some other methods. The tests, which are accomplished in countries having up to 2000 cities, depict the increasing of accuracy in obtained response during an acceptable time.

Keywords: Genetic Algorithm, Traveling Salesman Problem, Mutation Operator, Crossover Operator, Gradation, Grid

1. Introduction

Traveling salesman problem is an important problem in computer field which is considered a lot due to its different applications. It is mentioned as one of the main optimization problems and its solving plays an important role in other problems solutions. In this problem, some cities should be traveled by a salesman in the shortest way. Finding the shortest way between cities or the shortest Hamiltonian cycle in graph is the goal of this problem. It belongs to NP-Complete [1] problems and its absolute solving is practically impossible for maps with several cities. Therefore approximate algorithms are used to solve such a problem.

Genetic algorithm is one of the best approximate-exploring algorithms, which is created three decades ago via receiving an inspiration by natural structure and human evolution [2]. High adaptability and the generalizing feature of genetic algorithm help to execute the traveling salesman problem by a simple structure like array easily. Quick responding, accuracy, and high convergence are some other features of genetic algorithm as well. Genetic algorithm consists of some agents and input parameters that changing each of these cases plays an important role in the speed and accuracy of responses. Considering the traveling salesman problem both speed and accuracy should be improved in order to refer to [3], [4], [5], and [6]. Applying the reinforcement operators and adding them to the algorithm is one of the best methods to development and advancement of genetic algorithms agents which is known as hybrid genetic algorithm [6], [7] and [8].

Hence firstly in part 2, Style of optimization and adding different agents to hybrid genetic algorithm is offered. Results of different conditions in hybrid genetic algorithm and comparing them with previous algorithms are mentioned in part 3. New methods for solving the big problem on the basis of gradation in hybrid genetic algorithm are discussed in part 4 to solve the mentioned problem, and the obtained results are evaluated.

2. Compound Algorithm

One of the reasons for using GA in solving the problem is its rapidity in attaining the final output. This goal is obtained only when algorithms are introduced for large scale problems, too.

One of the ways used to improve GA is adding a function to bring the cities close together and eliminating unwanted distances [9]. The pseudo code of Algorithm is as the following:

Calculate distance gene(city) from next gene(city)
Determine distance how much (more than 1st closest city, 2nd, 3rd, 4th or 5th)
Calculate distance gene(city) from previous gene(city)
Determine distance how much (more than 1st closest city, 2nd, 3rd, 4th or 5th)
Optimize this gene(city)

This function, by improving a chromosome, improves the cities the distance of which with the next city or the previous one is longer than the ones on the map.

The second way to improve the efficiency of GA is using the best function for producing the initial population. All systems and operators of an Algorithm work on the initial population. By producing an initial population consisting of better chromosomes the rapidity and precision will increase. This function acts in a way that it chooses a city and places it in the first city, then another in the second city. It continues until a chromosome is completed. The pseudo code of this function is as follows:

Select city by random and save in NEXT
Put city in gene
Repeat (number of city)
Select from (bound) closest cities to NEXT and put in next gene
If (bound) use cities in gene **then**
Increase bound (step by step) and select from them and put in next gene
Last city in gene put in NEXT

The third way used in GA is mutation function. Mutation function acts in the problem of traveling salesman in a way that two cities are randomly chosen and replaced each other. By confining the space of selection of two cities from among Genes, it is expected that the mutation function acts in a better way. The pseudo code of the new function is as follows:

Create domain by random
Select 2 genes from chromosome (domain length)
Swap 2 genes

This function is placed beside the main mutation function so that the second function is recalled to the same extend of probability as the mutation function is carried out to the account of 50% of the main function, if it is carried out. By adding these functions and improving GA, it may be expected that an output with 100 to 200 cities in an appropriate time is obtained.

2.1 Fonts

Table 2 shows the font sizes and highlighting in a typical manuscript. These font "styles" are contained with this sample manuscript and Section 4 below explains how to use them. Use Times Roman or another standard font to avoid font errors.

3. Investigating the Results of Different Algorithms Putting into Practice

Practical investigation of pure and compound Algorithm indicates an improvement and development in Algorithm; likewise by manipulation of other input parameters including the number of population, percentage of implementation of functions, and the other parameters, an appropriate Algorithm was established to enable us to solve a problem with 200 cities in an acceptable time.

The inputs used in the Algorithm in the form of a text file includes: the number of the cities, the dimensions of the map, geographical latitude and longitude of each city. These maps are set and analyzed according to the maps and the outputs in [10]. In order to calculate the percentage of the correctness of outputs based on the best output, the following formula was used.

$$respomse = \frac{avarage - optimal}{optimal} \times 100 \quad (1)$$

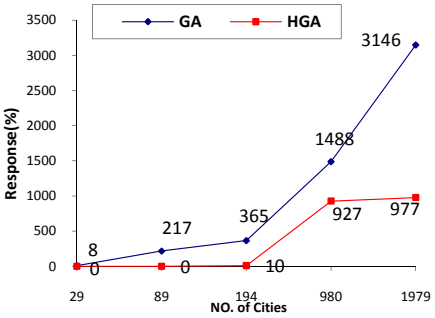
All the experiments have been carried out in Delphi2007 language, and by a computer with Core2 Duo 2.20 GH2 processor under Windows Vista (32 bit). The output analyzed per a parameter is the outcome of 5 times experiments of Algorithm in equal conditions. The best outcome was taken as the output for that parameter.

In pure GA, each chromosome is as long as the number of cities. In order to begin laboring algorithm, an initial population with a definite number of chromosomes is built. Each population creates its following population through synthesizing and mutation.

Table 1: pure and compound GA for investigation

Algorithms	Number of Generations	Number of First Generation	Crossover of each Generation	Hole crossover operation	mutation	neighborhood
Pure GA	200	200	10000	-	70%	-
Hybrid GA	-	400	-	20000	90%	90%

Compound GA is jus like pure Algorithm. Their difference is only that in it the initial population with a definite number of chromosomes are developed and taken as a clan. Then, in this clan the action of synthesizing, mutation, and neighboring of chromosomes is taken. The embryos produced are replaced by the improper chromosomes of the clan.



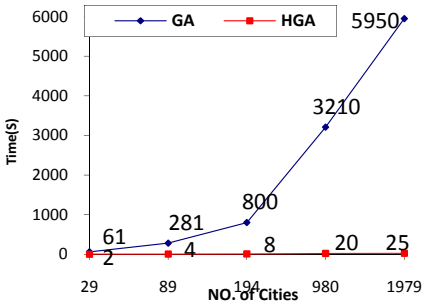


Figure 1: the ultimate output (percent) for the pure and compound GA

Figure 2: The time (second) spent for pure and compound GA

In the diagram Fig.1, the result of analysis of increase in the number of the cities over the percentage of improvement of the ultimate output is indicated. The percentage of improvement is computed with formula (1).

In Diagram Fig.2, the result of increase in the number of cities over the time spent is determined.

The following diagrams illustrate the best track and output of Algorithm for pure and compound Algorithm.

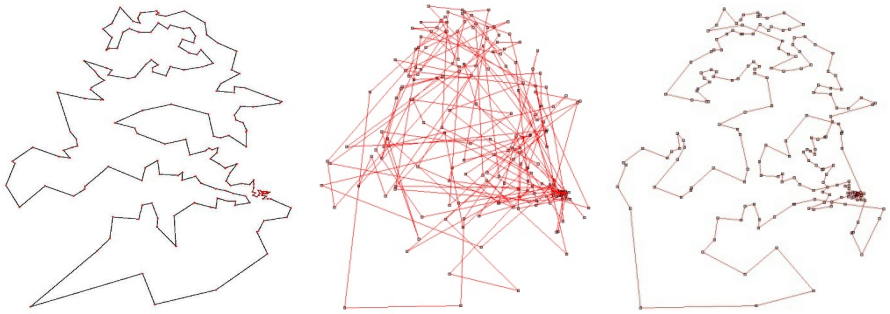


Figure 3: The best track between Qatar and other 194 cities on the map [10]

Figure 4: The pure GA output for the map of Qatar

Figure 5: The compound GA for map of Qatar

An improvement in precision and rapidity of compound GA in attaining the answer for the performed experiments is visible. By adding functions, and deleting futile actions, and creating better initial population

this Algorithm has the capability of 200-400 cities; however, with an increase in the number of cities and extension of the problem, the final answer will be out of efficiency and being the best.

4. Genetic Algorithm According to Grid

Regarding fig. 1 and 2, it can be seen that hybrid genetic algorithm is applicable for some problems with few cities. Increasing the number of cities removes the optimization and quick availability.

Clustering and classifying the traveling salesman problem is one of the solutions of big problems. Traveling salesman problem consists of a graph that is composed of some apexes and crests. One or few graph apexes are set in one part after the classification. There are different methods to solve the cluster traveling salesman problem (CTSP) [11]. Classifying a graph may be done according to different criteria. This classification has been done in this article according to cities situations in order to put close cities next to each other and at the same class.

Applying the genetic algorithm may play an important role to increase the speed and accuracy of response in solving a big classified problem. Accomplished researches and obtained results in this field [12], [13], and [14] present this fact that executing the genetic algorithm on a classified map responds more optimized comparing with the same algorithm on an unclassified map.

4.1 Chromosome Performance

One data structure is used to perform each chromosome in clustered problems, in which cities are located in separate classes, and each of them clarifies one class or cluster.

C= [G0(X1, X2), G1(X3, X4, X5) … Gn(Xn-2, Xn-1, Xn)]

In this algorithm, besides the cities in one cluster that are solved by genetic algorithm, the structure of genetic algorithm is applied in order to connect the clusters as well. Each cluster is considered as an apex and the shortest way between clusters is found by genetic algorithm. The structure of cluster chromosomes is as follow:

$$G= (G1, G2, G3… Gn)$$

4.2 Clustering

Clustering the map is one of the most important parts of this algorithm, which is done in different ways. The process of work in this algorithm is as follow: the number of horizontal and vertical Grids is sent to the algorithm as parameters. Close cities are put next to each other and are classified at the same class after clustering the map.

4.3 Genetic Algorithm Agents according to Grid

The hybrid genetic algorithm agents are used as the base of this algorithm. The tree-level structure of this algorithm is one of its features. This structure is executed as follow: firstly, the most optimized ways are found for cities in the cluster by applying the genetic algorithm; then each cluster is considered as a city or a graph apex in second level in the genetic algorithm and the most optimized way is found between the clusters. Proper qualified chromosomes can be created by combining these two responses. In third level, these chromosomes are excluded the cluster structure and are evaluated by genetic algorithm as a simple chromosome in order to find the best response in the shortest possible time. The work process of the algorithm can be summarized as follow:

Initialize population for GRIDs
For each Grid
Initialize population for CITIes in GRIDs
Repeat until child=0
Do crossover, mutation, close city on chromosomes from GRID population
For each Grid
Do crossover, mutation, close city on chromosomes from CITIES in GRID population
Move chrome (CITIES in GRID) to simple chrome (without Grid record)
Repeat until A-child=0
Do crossover, mutation, close city on chromosomes from new population
Select best chromosome by fitness

Output the best individual found

Initially, the first generation of Grids is created according to the cities in each Grid. Then algorithm agents – such as crossover, mutation, and neighborhood – are implemented on the first generation with a definite number of repetitions. During next level, chromosomes are excluded the Grid structure and are saved as simple forms in new chromosomes. The final response is obtained via implementing the algorithm agents on these chromosomes and choosing the best chromosome among them.

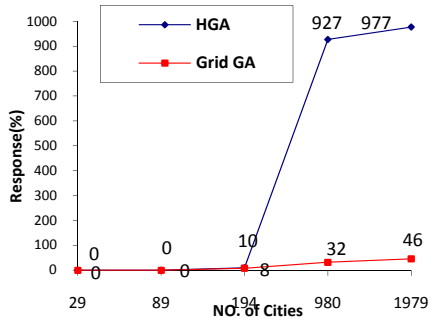
4.4 Results Evaluation

Formula (2) is used to evaluate the output of this algorithm and comparing it with hybrid genetic algorithm in order to calculate the optimization percent. The algorithm is executed and tested by Delphi 2007 in a computer – with Core2 Duo 2.20 GHz – in Windows Vista.

Table 2: Genetic algorithm parameters according to Grid for testing

Crossover Operation – Level 3	Number of Chromosomes – Level 3	Neighborhood	Mutation	Total Crossover Operation	Number of the First Generation
50000	400	90%	90%	1000	400

This algorithm looks like the hybrid genetic algorithm, the only difference is that the operation is done inside each cluster by the algorithm, and combining the clusters leads to final response.



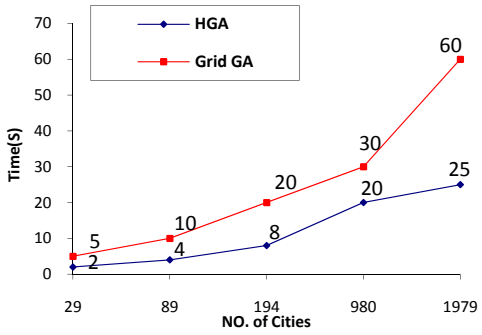


Figure 6: final response (percent) for hybrid genetic algorithm and Grid

Figure 7: Passed time (second) for hybrid genetic algorithm and Grid

It can be seen in fig. 6 that applying the cluster structure is able to propel the algorithm toward optimization. More optimized final response can be expected because of the passed mild steepness by genetic algorithm on the basis of Grid via improving the algorithm.

According to fig. 7, passing time in genetic algorithm on the basis of Grid is more than passing time by hybrid algorithm. It happens because this algorithm is executed in 3 levels and different clusters; nevertheless the passing time is absolutely optimized. In the end of the article, comparing five maps – low, medium, and high number of cities – are depicted for hybrid algorithms and Grid in table 3. It can be easily seen in these pictures that increasing the number of cities does not lead to lack of application in Grid algorithm. Map clustering and districting the map crests are the reasons of quick and accurate solving in Grid algorithm.

5. Results And Future Works

It was tried to offer a new method to solve the big traveling salesman problems in this article. Therefore it was tried to offer a solution which is able to solve big problems by evaluating the previous algorithms, comparing them, and distinguishing their positive and negative aspects. However this algorithm is not able to solve the traveling salesman problem completely in the cases of having more than 1000 cities and performs error in some cases, it can be expected to achieve to the final response by more accurate evaluating and modeling, and also removing the short comes. A method – called Grid GA – was introduced in this article in order to increase the accuracy by applying the genetic algorithm features, categorizing the problem, and changing it into smaller problems. Test outcomes have proven this claim as well.

6. References

[1] M. Hahsler, K. Hornik, “Infrastructure for the traveling salesperson problem”, Journal of Statistical Software, December 2007. ISSN 1548-7660

[2] M. Tomassini, “A survey of genetic algorithms” Annual Reviews of Computational Physics, volume III, pages 87-118. World Scientific, 1995.

[3] M. D. Vose, “The Simple Genetic Algorithm: Foundations and Theory” The MIT Press, Cambridge, MA, 1999.

[4] Paul Charbonneau, “An Introduction to Genetic Algorithms for Numerical Optimization”, National Center for Atmospheric Research, Boulder Colorado, March 2002.

[5] Li-Ying wang, Jie Zhang, Huali, “An Improved Genetic Algorithm for TSP”, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.

[6] Kaur, D. Murugappan, M.M., “Performance Enhancement in solving Traveling Salesman Problem using Hybrid Genetic Algorithm”, 2008

[7] Uma B. Gurav, L.D. Netak, “Hybrid Genetic Algorithm Based Task Scheduling in Heterogeneous Grid”, Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India

[8] Jean-Yves Potvin, “Genetic Algorithms for the Traveling Salesman Problem”, Volume 63, Number 3 / June, 1996, Pages 337-370

[9] Shubhra Sankar Ray, Sanghamitra Bandyopadhyay, Sankar K. Pal, “Genetic operators for combinatorial optimization in TSP and microarray gene ordering”, Volume 26, Number 3 / June, 2007.

[10] Benchmark data at http://www.tsp.gatech.edu/world/countries.html

[11] Laporte G, Palekar U, “Some applications of the clustered traveling salesman problem”, Journal of the Operational Research Society, 2002, 53(9): 972-976.

[12] Lu C Y, Delgado-Frias J G, Lin W. A, “clustering and genetic scheme for large TSP optimization problems”, Cybernetics and Systems, 1998, 29(2): 137-157.

[13] DING Chao, CHENG Ye, HE Miao, “Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs”, Volume 12, Number 4, August 2007, ISSN 1007-0214 15/20 pp459-465.

[14] S. P. Koh, I. B. Aris, C. K. Ho, S. M. Bashi, “Design and Performance Optimization of a Multi-TSP (Traveling Salesman Problem) Algorithm”, AIML Journal, Volume (6), Issue (3), September, 2006.