International Journal of Computer Applications Technology and Research
Volume 5– Issue 8, 516 - 518, 2016, ISSN:- 2319–8656

The Presentation of a Genetic Algorithm to Solve Steiner Tree

Elham Naseh
Department of Computer Engineering, Qeshm international Branch, Islamic Azad University
Qeshm ,Iran

*Ali Asghar Safaee
Department of Medical Informatics,
Faculty of Medicine,
Tarbiat Modarres University
Tehran, Iran

Abstract: The problem of Minimal Steiner Tree is a classical and known one (NP-Complete). This problem has been used a lot in navigation of Networks. Since finding Minimal Steiner Tree is NP-Complete, there's no Algorithm for it in multi-nominal time and we should use other Algorithms like Approximate Algorithm or Random Algorithm. This study presents a new genetic Algorithm. In the conclusion, this proposed Algorithm will be evaluated.

Keywords: Navigation, Steiner Tree, NP-Complete, Genetic Algorithm

1. INTRODUCTION

The problem of Minimal Steiner Tree is a classical and known problem (NP-Complete). The directionless graph G(V,E) is given; the change of each Edge crest is non-negative and its vertexes are divided into two parts: A set of required vertexes and Steiner's vertexes .The purpose is to find the least expensive tree in G in a way that it involves all the vertexes of the required set and a subset of Steiner's vertexes .The problem of Steiner tree is defined as the following:

Graph , and a set of vertexes are given. The purpose is to find a tree in G with the least expenses, which includes all the vertexes of R set. Finding a tree with less expenses causes increase in the speed of navigation of computer Networks. [1]

If we consider figure 1 as the entrance graph and R, including colorful vertexes and MST Algorithm on R (without considering the vertex which is drawn hollow) is performed, it gives the tree in figure 2, while the optimum Steiner Tree will be tree number 3.

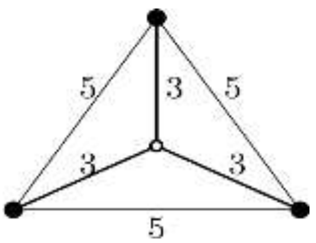


Figure 1: graphic example with three terminal vertexes and Steiner vertex [2]

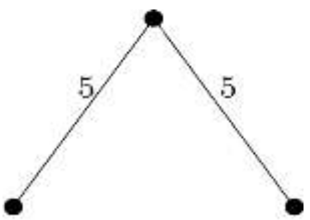


Figure 2, the created tree by Algorithm tree [3]

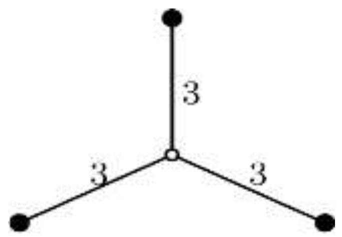


Figure 3: Steiner Tree

2. THE STAGES OF GENETIC ALGORITHM

In this article, genetic Algorithm is used to solve Steiner Tree whose structure is as follows [5]:

2.1 Displaying Gene Solution

The first step is a way to display the solution or Gene. In most applications, a range of bits is used. A range of zero and one (0 and 1) with the length of |V|; i.e. the number of vertexes of graph G. Each bit in the range is equal to one vertex of graph. Whenever it is the ith in the range, it means the ith vertex in graph or answer graph or Steiner Tree doesn't exist. For example, look at example 4. The Corresponding range for figure 5 is 11111110. It means vertexes 1 to 7 exist in answer graph, but vertex 8 doesn't. [6]

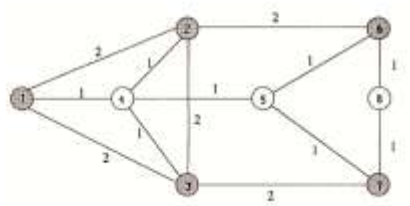


Figure 4: Graph G with 8 vertexes (5 Terminal vertexes and 3 Steiner vertexes) [7]

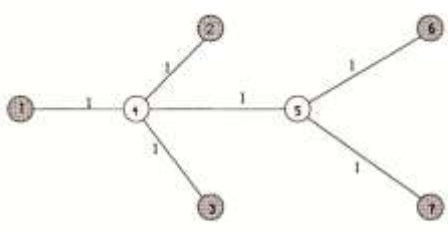


Figure 5: Steiner Tree for graph 4[7]

Not all the genes created by this method are a part of the solution to the problem. The genes in which the bits corresponding with terminal vertexes are zero (0) are not a part of the answer. These genes mustn't remain in the genetic reservoir, and they are omitted.

2.2 The Primary Generation (Population)

The second step is to create a primary population. To create a population, based on the experiments that are done, different techniques can be used. In action, first the population is created randomly and in the second stage, this randomly created population along with a possible answer that is equal to minimum Covered tree is placed for the whole graph; i.e. a whole range of 1 [8]. In the third stage, a detected answer from the remaining of the answer of minimum Covered Tree in the second stage is derived. In this article, 50% of the population is used in the first method and 50% in the third method.

After this stage, the number of population must be determined. It means how many chromosomes must exist in the gene reservoir. The number of population is considered as on important factor in Algorithm efficiency. If the population is too small, a small portion of the answer space will be searched for and the answer will probably be converted into a local optimum, and if the population is too large, a lot of calculations will be required, which is imbalanced in relation to the gained answer; therefore, performance time will be too long. In this article, the number of population is considered 100.

2.3 Fitness Function Calculation

Fitness function evaluates the suitability and performance of every member of the population. To solve the Steiner Tree problem, a function must also be defined to measure the rate of suitability of the gained answers; i.e. chromosomes. To measure the suitability of chromosome, a sub-graph of G is formed based on the combination of Terminal vertexes and Steiner vertexes – the Corresponding bit in the mentioned chromosome is 1. It is supposed that this sub-graph has K components. Minimum Covered Tree for each component is calculated by prim Algorithm and the total weight of all trees is considered as the rate of suitability. If $k \geq 2$, then a big punishment is considered for that chromosome.

2.4 Selection Methods

To choose the parents, we use Ranked Base Selection Method .The reason of this choice is that in the beginning of Algorithm; we prevent hasty convergence of the Algorithm. (The exploration of capability is carried out in Algorithm.) After several generations, the Algorithm must reach convergence. (The number of generations is gained by Trial and Error Software.) To do this, selection Algorithm is switched to Roulette Wheel Algorithm.

2.5 Crossover Combination

For fertilization in Ranked Base Selection, we use Uniform Crossover whose structure is as the following[9]:

In the open operation of Smooth Combination, the number of gene in the baby is chosen according to number of Corresponding gene from both parents. In this method, the number of genes from each parent has an equal chance of presence in the baby's Corresponding gene. In the open operation of Smooth combination, based on a random binary distribution, we can recognize according to which Corresponding gene of which parent, the number of the baby's genes must be chosen. One example is given in figure 6.

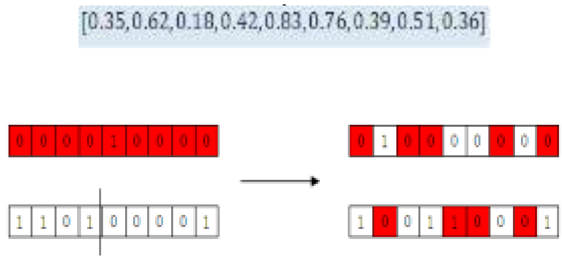


Figure 6: the presentation of an example of Smooth Combination

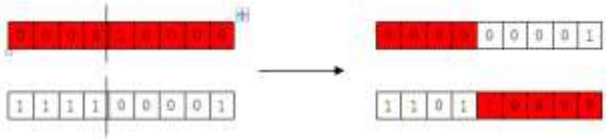
In Roulette Wheel Selection Method, we use one-point crossover whose structure is as the following:

In the operation of one-point Crossover, first a random point in the sequence of the parents' chromosomes is chosen, and then from the selected point, the chromosomes of both parents are sectioned. The second child consists of the first section of the second parent and the second section of the first parent. One example is in figure 7.

Figure 7: the presentation of an example of one-point crossover

2.6 Mutation

In the Mutation Operation with a change in the bit, one gene is randomly selected. Whatever its amount is , it changes. Its structure is shown in figure 8.It's obvious that because of not utilizing the existing information in the population, this operation has complete congruity with the definition of Mutation operation, and it tried to explore the Algorithm[10].



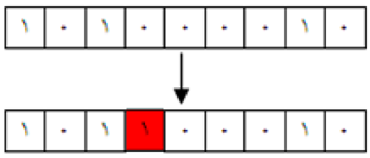


Figure 8- The presentation of an example of Mutation

In the beginning of Algorithm, the percentage of mutation (pm) must be raised so that the rate of exploration of Algorithm will be large. Then after several generations, Pm must be lowered so that it will keep up with the convergence of chromosomes.

2.7 Replacement

Here, we use generational replacement. In the beginning of the Algorithm, 50% of parents and 50% of children are transferred to the next generation by Generational Replacement, which then causes exploration in the problem. After each generation, the percentage of children will decrease and that of parents will increase so that it will end in convergence. The rate of percentage increase of parents is calculated by Trial and Error Method[11].

3. EVALUATION OF THE PRESENTED ALGORITHM

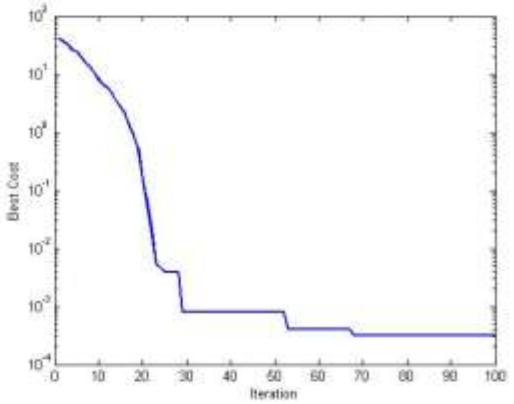


Figure 9: Algorithm with five Steiner nodes and production of 100 generations

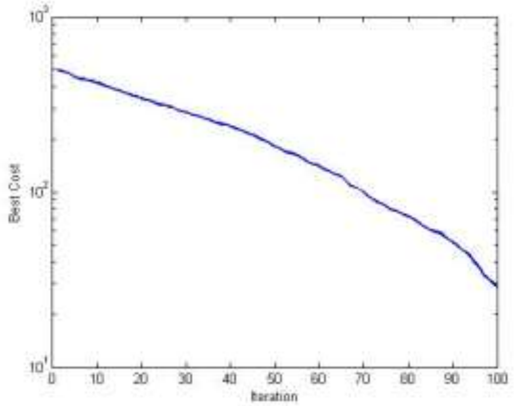


Figure 10: Algorithm with 20 Steiner nodes and production of 100 generations

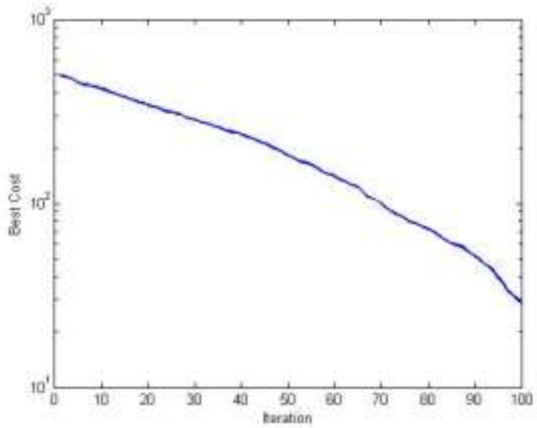


Figure 11: Algorithm with 5 Steiner nodes and production of 500 generations

4. Conclusion

In this study, using Genetic Algorithm, a solution was presented to solve the problem of Steiner Tree, which has reached optimum solution after 500 generations.

5. REFERENCES

[1] Richard M. Karp (1972). "Reducibility Among Combinatorial Problems". In R. E. Miller and J. W. Thatcher (editors).Complexity of Computer Computations. New York.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. (2011),"Introduction to Algorithms", Second Edition. MIT Press and McGraw-Hill.

[3] Vazirani, Vijay V. (2003). Approximation Algorithms. Berlin: Springer.

[4] vafader, Safar (2010) "Approximation Algorithms for the Steiner Tree", Master of Science, Tehran University.

[5] Shirazi, Hussein, Behcet, Shahab (2007) "Introduction to the Analysis and Design of Algorithms" First Edition, Press Malek-ashtar.

[6] P. Berman and V. Ramaiyer, Improved approximations for the Steiner tree problem J. Algorithms 17 . 1994 , 381]408.

[7] Neil Moore, Computational complexity of the problem of tree generation under fine-grained access control policies, information and Computation, Volume 209, Issue 3, March 2011, Pages 548-567.

[8] AZ Zelikovsky An 11/6-approximation algorithm for the network steiner problem , Algorithmica, 1993 – Springer.

[9] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," Acta Informatica no. 15, pp. 141–145, 1981.

[10] M. Doar and I. Leslie, "How bad is naive multicast routing?" IEEE INFOCOM' 1993, pp. 82–89,

[11] D.E. Goldberg ,1988,. Genttic Algorithms in Search, Optimization MachinLearning. Addision Wealey, New York.

[12] L Luyet, S Varone, N Zufferey , "An ant algorithm for the steiner tree problem in graphs" , Springer-Verlag Berlin Heidelberg 2007