International Journal of Academic Research (ISSN: 2075-4124)
May 2015, pp: 347-354

Solving the Minimum Steiner Tree by Using Firefly Algorithm

Leila Fathi[1], Faramarz Sadeghi*[2], Amin Golabpour[3], Kobra Etminani[4]

[1] Department of computer engineering, Kerman Branch, Islamic Azad University, Kerman, Iran,

[2] Department of computer engineering, Kerman Branch, Islamic Azad University, Kerman, Iran,

[3] Departments of Medical Informatics, School of Medicine, Mashhad University of Medical Sciences, Mashhad, Iran

[4] Departments of Medical Informatics, School of Medicine, Mashhad University of Medical Sciences, Mashhad, Iran

Abstract:

The issue of finding minimum Steiner tree in a valuable graph is finding a tree by least cost on graph which involves special loop naming terminal. This issue is out of NP-Complete issues, therefore, several approximate algorithms as genetic algorithm, ant colony, learning automata &etc has reported. In this paper an algorithm base on firefly algorithm is suggested for solving the issue of minimum Steiner tree. The results of tests show that suggested algorithm compare to reported methods as genetic algorithm & ant colony enjoys more proficiency.

Keywords: Steiner tree, NP-Complete, Firefly algorithm.

Introduction:

The issue of finding minimum Steiner tree in a valuable graph is out of NP-COMPLETE issues [1,2] , that is finding a tree by least cost which involves special loops naming terminal the issue has many types in one type many dot are presented on two dimensional paper as entrance & its aim is producing a tree on Euclidean paper which has the least cost & include all the dots[3].

The Steiner tree issue on graph is defined as follow. Consider the graph G(V,E) what v shows the complex of loops& $E \subseteq V*V$ shows the complex of graphs crests. A cost c(i,j) is attributed to the each crest on graph. One subordination of loops, naming T, is introduced as the complex of terminals. The arm of Steiner issue is finding a tree on graph which includes all the terminals &has the least cost. This tree is called minimum Steiner tree. If the outcome tree involves just the terminals; therefore, it is called minimum extender tree & there are several polynomial algorithms as prim algorithm to solve it.

There for, by using the loops of non-terminals the cost of tree decreases. These loops are called Steiner ‘s dots .it has shown a Steiner tree on sample graph in Figure 1.the terminals loops & tree’s crests of Steiner tree has drawn boldly .[4,5]

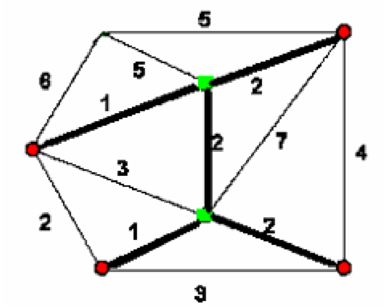

Figure 1 Steiner tree [4]

2) Firefly algorithm:

This algorithm has presented in Cambridge university by xin-she yang in 2007 [6].

In nature, the firefly by using chemical mechanism can spread the saved energy as light which is saved through food & other methods that some topics of firefly's operation are unknown, &can spread special pattern. These patterns are unique for different species of firefly or for hunts attraction, a pattern couses to escape the aggressor person .

The attractive point is that firefly produce attraction blight. Now, attractive can be a base optimization model.

There are some artificial firefly &they have fitness that it is possible to rank them. The better or more attractive firefly attracts more people but the amount of attraction is less in distance dimension.

2.1 Different ways of firefly algorithm.

- Producing first responses &evaluating them.
- For each firefly as i
    - For each firefly as j
- If I_i is less than I_j (I is better than j)
    - According to attractive formula, the $x_i^{'}$ is produced.
    - $\boxed{x_i{}' = x_i + \beta\left(x_j - x_i\right) + \alpha\varepsilon_i}$ $\beta = \beta_0 e^{-\gamma r^m}$
- The evaluation of new firefly
- Determining the best found response.
- Repeat from second step if the ending condition s fulfilled.

The structure of firefly algorithm is seen in Figure 2.

```
Firefly Algorithm
   Objective function f(x),    x=(x_1,....x_d)T
   Generate initial population of fireflies
xi(i=1,2,…,n)
   Light intensity Ii at id determined by f(x_i)
   Define light absorption Coefficient γ
   While (t<maxGeneration)
    For i=1:n all n fireflies
      For j=1:n all n fireflies (inner loop)
          if(I_j<I_i) more firefly i towards j; end
if
            vary attractiveness with distance r
via Exp[-γr]
            Evaluate new solutions and update
light intensity
       end for j
     end for i
     rand the fireflies and find the current
global best g*
     end while
     Post process results and visualization
```

Figure 2 Firefly algorithm code [7]

3. Literature review

Garey, garaham & Johnson presented in 1977 that the Steiner issue is NP-Hard. Therefore, the solving algorithms of this issue are divided into absolute algorithm & approximate algorithm.

3.1: absolute algorithm

The first time, melzak suggested an absolute algorithm for solving Euclidean Steiner in 1961. In his method, at first all full Steiner topology is calculated. Then, the lightest topology is chosen. In Steiner topology each Steiner 's dot has grade 3 & each terminal has 1.the Steiner topology is complete if it has included n-2 Steiners dots &n is the number of terminal. Full Steiner topology is designed for four terminals in Figure 3[8].

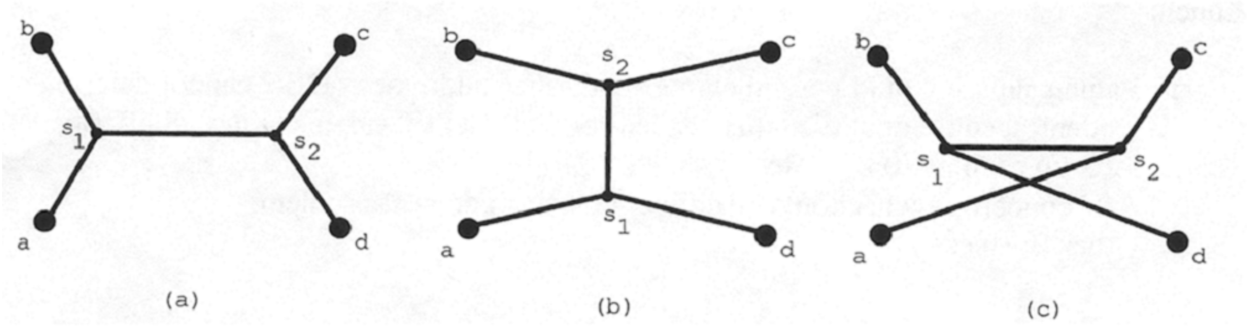

Figure 3 Full Steiner topology for 4 terminals [8]

The number of full Steiner topology in a graph by n terminal is as follow:

$$f(n) = \frac{(2n-4)!}{2^{n-2}(n-2)!}$$

The problem is, the above function grows quicker than exponential function therefore solving the issues which have many terminals impossible.

Winter suggested studying the subordination of full Steiner topology to improve this method. Currently, the best absolute algorithm.

of solving Steiner's , tree issue is presented method by warm & other[9].

This method which is named GeoSteiner can solve issues by more than 2000 terminals. the GeoSteiner method receive help from the idea that a Steiner tree in Euclidean space or horizontal vertical space is crowd of many full Steiner tree that terminals are adjacent full Steiner trees crest .Therefore, firstly we need to produce full Steiner tree then do some examinations for keeping non-necessary full Steiner tree.

3.2: approximate algorithms

Since the absolute algorithms of Steiner issues solving involves performance

Performance time in worst manner; there for most of reported algorithms for solving this issue are approximate algorithms. there are some special cases of this issue which Polynomial solutions has presented for it[10].these cases are used as a base for solving other cases. If the number of terminals, crests & loops are orderly shown by n,m &v , the solvable cases in multinomial time are as follow .

- N=1: in this case there is just one terminal which is the answer.
- N=2: in this case the solution is finding the shortest path between two terminals. This issue is solvable by help of dijkstre method in O(vlogv+m).

- N=3: for this issue which is called format issue, different solutions has presented torricell ; suggested to create a triangle on 3 terminals in 1640 .Then a parallelogram triangle create on each side of triangle. The circle around the triangles cut each other in one dot. This dot considers as Steiner &math to the three terminals [8]. This step has shown in Figure 4.

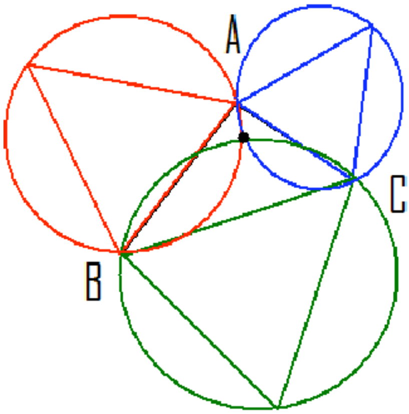

Figure 4 Torricelli method for solving Steiner tree problem [8]

In 1750 sympson suggested in torricelli's method instead of drawing around circles ; it is drown three lines which each of them math the vertices of triangle which is not terminal to the terminal which is not part of triangle is vertices. In the cross place of these lines, the Steiner's dot is need. This solution has shown in Figure 5.

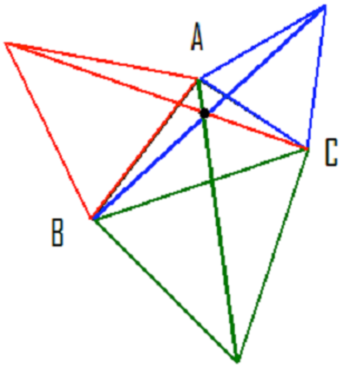

Figure 5 Simpson method [8]

In the above methods when the terminals make triangle which one of the angle is more than 120; therefore, they don't get the answer. In the first the Steiner's dot is out of triangle & in the second method, the three lines don't cut each other in a dot. Hymn presented a general solution for this issue in 1834. He divided the issue into two part[8]. If the terminal's triangle didn't have angle bigger than 120, if would be solved by one of two above method's; Otherwise, the Steiner's dot matches to the bigger side of triangle. If the triangle had not angle, bigger than 120, the Steiner's dot is the cross place of three crests which create 120 angle. This feature is called angle condition. This condition is so significant in solving the cases which n>3.

Different approximate algorithms has reported for solving Steiner's tree in Steiner's environment. One of the simplesr & the earliest of these methods is using minimum extender tree. In these methods no Steiner dot is produced but by using algorithms of making minimum extender as prim & kruskal , the trees are created which support all the terminals. These methods involve the worst time of performance O(n^3) &factor approximation 2.

In order to calculate the approximate methods of solving the Steiner‘s tree, a factor approximation naming Steiner’s rate has introduce which the cost of minimum extender tree. In the all vertices divide into the cost of Steiner tree is defined. Gilbert & Pollak guessed in 1964 what the best available factor approximation is 1.1547.du & Hwang fixed this hypothesis in 1992[11].

Many of the approximate method of solving Steiner‘s tree started by a minimum extender tree &try improve this tree by embodying Steiner’s dot.

Many of the presented discovered methods for solving Euclidean Steiner are based on local search. Local search is a general method of searching which is used in a wide spectrum of combinatorial optimization. A combinatorial optimization issue includes series of solution that each solution of x includes the cost of f(x) and the aim is the solution by the least cost. The solution space in Euclidean Steiner’s issue is not finite since the Steiner’s dot can be situated in all parts of the paper. Therefore, it is possible to gain a finite set of candidate dots for Steiner dots [12].

Another ,method which has research to the good result is using genetic algorithm[13], for solving Steiner’s tree each solution is shown by Steiner’s dot’s coordinate that are saved an vector. In order to make easier the creation of tree, it is assumed that each Steiner’s consecutive double dots on vector match to the nearest Steiner’s dot & there is no need to calculate then connection between dots. The aim is finding Steiner’s dots in bought. Therefore, the order of Steiner’s dots on vector is important. We use genetic operator for generation.

Finally, the best person of last generation choose as the answer,

4. Solving the issue by firefly’s method.

To solve a firefly issue, firstly we should define artificial firefly in this issue we consider firefly as Steiner‘s loops which is array of binary values and each array element shows If its value is zero, the node Steiner tree component is and if it is equal to one means that the component tree.

4.1 Producing the early population;

50 percent of population has presented from previous exit & 50 percent of population has created randomly. The first 50 percent courses isotropic in algorithm but the second 50 percent is random which courses divergence, in algorithm.

After that the way of producing the early population defined then the measure of population is determined. That how may firefly should be existed in tree sury. The measure of population considers as main factor in algorithm. If the measure of population is small, the small part of answer‘s space will be searched & the answer converge fast &by local optimization. If the measure of population is large, many calculations will be done which are not suitable with the answer; therefore the time of performance last long time. In the algorithm evaluations chapter, different manners of population studied &it is defined. How many people is need for early population.

4.2 The calculation of merit function

The Fitness function evaluation the operation of each member of population in solving the issue &for solving Steiner‘s tree, one function should be defined for measuring gained answer that is artificial firefly. For measuring the fitting extent of each firefly, the sub graph of G is based on the community of terminal vertices &Steiner’s vertices that all the terminal’s vertices are considered and form the Steiner‘s vertices which are equal to the same byte in the firefly & its one. By using prim algorithm calculated the minimum extender tree and sum of the weight is considered as fitting extent.

4.3: Using attraction law:

All the artificial fireflies compared. The firefly by more light courses the attraction of the firefly by less light. The criteria for more light are the optimization of Steiner's tree. The fireflies which have less light attract toward the fireflies which have more light. The following formula is used for attraction.

$$x_i' = x_i + \beta(x_j - x_i) \ ,\beta = \beta_0 e^{-\gamma r^m}$$

In this study the amount of y=1, m=0.2 & β_0=2 has considered orderly. By running algorithm &after each generation the amount of m increased and in the end of algorithm the amount of m is 2. Then, all the gained measures of x_i' are put in another array.

4.4 Using leaping for artificial firefly

The leaping algorithm applied a gained of attraction using, the formula (1) and calculated as the follow:

$$x_i'' = x_i' + \alpha\varepsilon_i \quad (1)$$

ε_i is a random vector which include same distribution and use the number Zero and One. The sub coefficient of ε_i causes leaping in algorithm and the amount of leaping coefficient shouldn't be high &the amount of a is gained through the formula (2).

$$\alpha(t) = \alpha_0\left(e^{-k}\right)^t$$

In this formula k is the extent of algorithm's decrease and its measure equal 0.5 if the calculated measures are less than 0.5, hen change to zero otherwise change to one.

4.5: Replacement

We use generational replacement [14] since this method has the possibility of convergence & divergence by changing its variables. Firstly, the 50% algorithm of parents and 25% algorithm of children produced through attraction & 25% of produced children through leaping method are moved to next generation that Couse divergence . After each generation, the percent of children decreased and the percent of parents increased in order to courses convergence .the amount of parent's percent increasing calculated through effort end errors (experience).

4.6 The condition of ending firefly algorithm

The condition of ending is one of the following

- gaining the 80 percent of optimization
- .producing 100 generations with no changing in merit function. (The amount of 100 has gained by effort and errors).

5. Evaluation:

the suggested algorithm on the graph of β set has examined of or-library this set include 18 graph which the number of loops, crests &the cost of minimum Steiner tree is clear for each graph[15]. The result of the study is seen in the table 1.

Table 1 Firefly by Firefly algorithm compared with 100 for the initial population standard B

Firefly algorithm execution time in milliseconds	cost of the Steiner tree algorithm obtained by Firefly	cost of the optimal Steiner tree	Number of terminals	Number of edges	number of vertices	Graph
15	82	82	9	63	50	B01
16	83	83	13	63	50	B02
18	138	138	25	63	50	B03
15	59	59	9	100	50	B04
19	61	61	13	100	50	B05
20	122	122	25	100	50	B06
30	111	111	13	94	75	B07
31	104	104	19	94	75	B08
30	220	220	38	94	75	B09
34	86	86	13	150	75	B10
34	88	88	19	150	75	B11
36	174	174	38	150	75	B12
53	165	165	17	125	100	B13
54	235	235	25	125	100	B14
55	318	318	50	125	100	B15
58	127	127	17	200	100	B16
58	131	131	25	200	100	B17
59	218	218	50	200	100	B18

It is used a Linux red hat 9 operating system on the 5 core computer by 2.5 Gigabytes processor and RAM 6G to do the examinations.

6. Conclusion:

In this paper, an algorithm based on firefly algorithm suggested for solving Steiner's tree. The result of examinations showed that the suggested algorithm enjoyed high efficiency. The result of simulation shows the efficiency of suggested algorithm either from the view of the quality of gained answers or convergence speed to the answer compares too many reported algorithms. Furthermore, since the time of running of this method is suitable with the number of graphs loop; therefore, it has suitable convergence speed for crowded graphs (the graph by many crests).

7. References

1. Karp RM. Reducibility among combinatorial problems [Internet]. Springer; 1972 [cited 2014 Oct 29]. Available from: http://link.springer.com/chapter/10.1007/978-1-4684-2001-2_9

2. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms, 3rd Edition.

3rd edition. Cambridge, Mass: The MIT Press; 2009. 1312 p.

3. Robins G, Zelikovsky A. Improved Steiner tree approximation in graphs. Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms [Internet]. Society for Industrial and Applied Mathematics; 2000 [cited 2014 Oct 29]. p. 770–9. Available from: http://dl.acm.org/citation.cfm?id=338638

4. Prömel HJ, Steger A. The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity. Softcover reprint of the original 1st ed. 2002 edition. Braunschweig: Vieweg+Teubner Verlag; 2002. 241 p.

5. Wu BY, Chao K-M. Spanning Trees and Optimization Problems. 1 edition. Boca Raton, FL: Chapman and Hall/CRC; 2004. 200 p.

6. Yang X-S, editor. Cuckoo Search and Firefly Algorithm: Theory and Applications. 2014 edition. New York: Springer; 2013. 360 p.

7. Yang X-S. Firefly algorithms for multimodal optimization. Stochastic algorithms: foundations and applications [Internet]. Springer; 2009 [cited 2014 Oct 29]. p. 169–78. Available from: http://link.springer.com/chapter/10.1007/978-3-642-04944-6_14

8. Herring M. The Euclidean Steiner Tree Problem. Stud Scholarsh [Internet]. 2004 [cited 2014 Oct 29]; Available from: http://ohio5.openrepository.com/ohio5/bitstream/2374.DEN/5095/1/herring.pdf

9. Wang K, Chen J-H. An efficient probabilistic dynamic multicast routing in ATM networks. J Inf Sci Eng. 1999;15(4):485–504.

10. Polzin T. Algorithms for the Steiner problem in networks [Internet]. Universitätsbibliothek; 2003 [cited 2014 Oct 29]. Available from: http://scidok.sulb.uni-saarland.de/volltexte/2004/218/

11. Hougardy S, Prömel HJ. A 1.598 approximation algorithm for the Steiner problem in graphs. Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms [Internet]. Society for Industrial and Applied Mathematics; 1999 [cited 2014 Oct 29]. p. 448–53. Available from: http://dl.acm.org/citation.cfm?id=314500.314599

12. Zachariasen M. Local search for the Steiner tree problem in the Euclidean plane. Eur J Oper Res. 1999;119(2):282–300.

13. Haouari M. A hybrid Lagrangian genetic algorithm for the prize collecting Steiner tree problem.

14. Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. 1 edition. Reading, Mass: Addison-Wesley Professional; 1989. 432 p.

15. Beasley JE. OR-Library: distributing test problems by electronic mail. J Oper Res Soc. 1990;1069–