

# Steiner Path for Trees

Shanti Swaroop Moharana  
Dept. of Electrical Engineering  
BITS Pilani University, India

Ankit Joshi  
Dept. of Software Engineering  
SRM University, India

Swapnil Vijay  
Dept. of Computer Science  
VIT University, India

## ABSTRACT

The Steiner minimum tree (SMT) problem is one of the classic nonlinear combinatorial optimization problems for centuries. The Steiner tree problem finds the minimum cost tree connecting a given set of nodes called required nodes in a given undirected weighted graph. In this minimum cost Steiner tree due to the reduction of path-cost, some non-required nodes are also used while finding the SMT, which are called Steiner nodes. The Steiner path problem comes from the Steiner tree problem of finding the minimum cost path connecting a given set of nodes called required nodes in a given undirected weighted graph. The Steiner path problem can be characterized as either a decision or optimization problem. This paper focuses on solving the Steiner path as a decision problem. The Steiner Path decision problem finds whether there is a path connecting a given set of nodes called required nodes in a given binary tree. The decision that whether there exists a Steiner path or not can help in further solving the Steiner path optimization problem. Since the Steiner tree decision problem has wide scales, we should use an efficient algorithm with least complexity.

In this paper we have defined the steiner path decision problem, characterized sufficient conditions for finding the steiner path and presented an algorithm for finding the existence of a steiner path in a 2-tree.

## General Terms

Graph Theory, Theoretical Computer Science, Discrete Mathematics, Algorithms

## Keywords

Steiner Path Decision Problem, Steiner Path, Graph Theory, Algorithms, Sufficient conditions for Steiner path

## 1. INTRODUCTION

Steiner Tree Problem [1, 3] is one of the classical problem in combinatorial optimization, which was named after the Swiss mathematician Jacob Steiner. In real life, people often need to address problems like this, namely given a number of points which are connected to each other and the problem is to find the minimum cost spanning tree containing these points.

In network communications, data always starts from one or more sources, and then sent to multiple destination nodes.

This problem is often described as the multicast routing problem. It seeks the minimum spanning tree (Minimum Spanning Tree-MST) connecting these  $n$  points, but it is not limited to  $n$  points, other points can also be added apart from the  $n$  points, so as to make the total length of the communication lines even shorter. This is the source of Steiner minimum tree problem (abbreviated as SMTP) [2].

This problem of optimizing the network cost of multicast tree is usually referred to Steiner tree problem. For many years, experts and scholars at home and abroad are discussing from different aspects in solving SMT problem. SMT problem has extensive applications in many real-world problems, such as communications network and routing in the urban traffic networks. In such networks with a large amount of nodes and edges, finding the optimum path which connects less number of terminals is desired. Steiner ratio is used as a metric for finding the minimum Steiner path in optimum time. Steiner ratio is the ratio of length of the Steiner minimum tree [3] and the minimum spanning tree.

In 1968 Pollak and Gilbert have proposed the following conjecture [3]: For any finite set of points on the Euclidean plane, the Steiner ratio is not less than  $3/2$ , but they were not able to prove the conjecture by themselves. In 1992, Ding-zhu Du and Guang-ming Huang proved this conjecture [4]. In 1972, Karp and others proved that the minimum Steiner tree problem is a NP-complete problem [5].

In recent years, a series of evolutionary algorithms have been proposed, using the concept of simulated annealing, genetic algorithms and ant colony algorithm [6, 7, 8] to solve the steiner tree problem. These evolutionary algorithms are able to obtain optimal solution for the Steiner tree problem. But this optimal solution is found through a long iterative process which has a longer running time and hence has lower efficiency.

Heuristic algorithms are not able to get the optimal Steiner tree in most cases, but they are able to find the quasi-Steiner trees of optimal cost in a relatively short period of time, so they have more practical significance in the cost optimization of routing network. Thus, the only way to solve the problem of Steiner minimum tree is to find an effective approximate solution [9, 10].

In 2011, Euclidean bottleneck Steiner path problem [13] was proposed by A. Karim Abu-Affash, Paz Carmi, Matthew J.Katz, and Michael Segal. This algorithm basically concentrates on Euclidean plane where points are placed arbitrarily and a path is obtained by joining a set of given points. EBST problem is considered as classic NP complete. However, EBSP can be solved in polynomial time according to the paper. The complexity of this algorithm is  $O(n \log_2 n)$ .

In 2011 a heuristic algorithm for solving Steiner tree problem on urban networks [11] was proposed by Fatemeh Ghadimi and Ali Nourollah. An urban traffic network problem consists of vast no of edges and nodes. Therefore heuristic approach is followed, which solves the problem in low runtime. The exact algorithm leads to a complexity of  $O(n(m + n \log n))$ .

In 2012 a minimum Steiner tree algorithm was proposed by Narendra Maharjan, Abinash Shrestha, Ashish Tamrakar and Sanjeeb Prasad Panday based on behavior of ants [12]. It is a heuristic approach for finding a minimum Steiner tree based on the deviation produced by ant's path which indirectly affects movement of other colonies. The stated algorithm takes very less time in finding a Steiner tree to its minimum length.

However, Steiner path still remains off light here. Any conditions or criteria which lead to Steiner path are not mentioned.

Since 1972, Steiner tree problem (STP) is a NP-Complete problem [5] and hence there exist no optimal solution which can be found in polynomial time. All the algorithms which provide optimal solutions have time complexity in exponential time. Various heuristic approaches are identified which reduces complexity to polynomial time but again these approaches are effective only for certain types of problem.

As we can see all the previous approaches were focused only on Steiner tree except in EBSP [13]. Hence Steiner path still remains an unsolved problem.

Steiner path problem has many applications such as network meeting, which require proper routing to transfer data from one point to another point. It is a problem to find a path between a given set of vertices' in a tree. Steiner path problem allows addition of vertices' that are not part of required points vertices' but are part of given binary tree. Steiner path problem finds the shortest path connecting the required points, allowing the addition of auxiliary/steiner points to the set for the purpose of minimizing the total length of the path.

This paper focuses on Steiner path problem on which no significant work has been done earlier. In this paper a novel algorithm based on recursion is proposed for finding the existence of a Steiner path in a given 2-tree.

## 2. STEINER PATH PROBLEM

In this section, we present a formal definition of Steiner path decision problem. In the next section, we present a polynomial-time algorithm for Steiner path decision problem. In the subsequent sections, we present an implementation using Hash table data structure followed by the run-time analysis of the algorithm.

### 2.1 Problem Definition

Steiner path decision problem finds whether there exist a path which covers all vertices present in R. There can be additional vertices included in path which are not part of R, but the main goal is to find whether there exists a path which consists of all R vertices. Existence of Steiner path is not always necessary; there are many scenarios in which a Steiner path is not possible for a given tree.

Input:

Given a binary tree  $T$  and a set of required terminals  $R$  such that  $R$  is in  $T$ .

Output:

A decision as result either Yes or No.

### 2.2 Prelims

We denote the given tree as  $T$ , set of nodes present in  $T$  as  $N$  and the root of  $T$  as  $r$ .

The root is at level 0, the siblings who are at a distance 1 from  $r$  (i.e. neighborhood of  $r$ ) are at a level 1. In particular any level  $l = i$  contains a set of vertices which are at a distance  $i$  from  $r$ .

$$L_0 = \{r\}$$

$$L_1 = \{Nd(r)\}$$

$$L_2 = \{Nd(Nd(r))\}$$

We arbitrarily root the given tree at any node  $v$  in  $N$  to get subtree  $T_i$ , where  $i = size(N)$ . Then we root the left child node and right child node as two subtrees  $T_j$  and  $T_w$  in case they exist. We follow the above procedure recursively to get all the subtrees of the given tree  $T$ .

### 2.3 Lemma-1

Given a tree  $T$  rooted at  $r$  and contains  $i$  levels.

At only a single level  $j$ ,  $L_j$  of the given tree  $T$  there can be at most two nodes which belongs to  $R$ .

$$L_j \cap R = 2$$

If the above condition is not satisfied then Steiner path does not exist.

### 2.4 Lemma-2

Given a tree  $T$  rooted at  $r$  and contains  $i$  levels. Consider nodes  $u, v, w \in R$  and a single level  $j$  then

$$u, v \in L_j \parallel u, w \in L_j \parallel v, w \in L_j$$

$$\text{such that } L_j \cap R = 2$$

At all other levels  $k = 1..i$ ,  $i \neq j$  of the tree there can be a maximum of one node which belongs to  $R$  and is at a level  $L_k$ .

$$L_k \cap R < 2$$

If all three nodes are at the same level  $L_i$ , then there exists no Steiner path. The proof of the preceding statement is clear because in any tree there can be a unique path containing two nodes. So there can be a maximum of only one nodes at all level  $k$  where  $k = 1..i-j$  such that nodes in  $L_k \in R$  except at the single level  $i$ ,  $L_i$  where there can be a maximum of two nodes which belongs to  $R$

### 2.5 Lemma-3

Given a tree  $T$  rooted at  $r$  and the subtrees  $T_i$  where

$$i = 1.. size(N)$$

For any subtree  $i$ ,  $T_i$  having child nodes  $n1, n2..ni$ . The trees rooted at  $n1, n2..n_i$  are the subtrees  $T_{n1}, T_{n2}..T_{ni}$ . There can be only one subtree  $j$ ,  $T_j$  rooted at  $j$  which can contain successor child nodes belonging to  $R$  and is contained in more than one subtrees  $T(jl1)$  and  $T(jr2)$

$$T(jl1) \cap R \neq \emptyset$$

$$T(jr2) \cap R \neq \emptyset$$

If the above condition is not satisfied then there exists no Steiner path.

### 2.6 Lemma-4

Given a tree  $T$  rooted at  $r$  and the subtrees  $T_i$  where

$$i = 1..size(N)$$

For any subtree  $T_j$  having child nodes  $n1, n2..n_i$ . The trees rooted at  $n1, n2..n_i$  are the subtrees  $T_{n_1}, T_{n_2}..T_{n_i}$ . There should be no subtree  $p$  where  $p=1..size(N), p \neq j, T_p$  rooted at node  $p$  containing successor child nodes belonging to  $R$  and is contained in more than one subtrees  $T(p1)$  and  $T(pr2)$  except for a single subtree  $j, T_j$  rooted at  $j$  which can contain successor child nodes belonging to  $R$  and is contained in more than one subtrees  $T(j1)$  and  $T(jr2)$

$$T(j1) \cap R \neq \emptyset$$

$$T(jr2) \cap R \neq \emptyset$$

If the above two conditions are true more than once then there exists no Steiner path.

## 2.7 Observation

Given a tree  $T$  rooted at  $r$ , set of nodes  $N$  and the required set  $R$ . Consider nodes  $u, v, w \in R$

An algorithm for finding the Steiner path is presented in Algorithm1.

```

Steiner-path-find(T)
{
  COMPUTE Least common ancestor(LCA) for each pair of
  nodes which belongs to I
  IF LCA path contains all of the nodes present in R
  Steiner path exist
  ELSE
  Steiner path does not exist
}

```

## 2.8 Sufficient conditions for existence of Steiner path

1. At any level of tree there can be at most two nodes  $v_1, v_2$  which belongs to the required terminal set  $R$
2. There can be at most two nodes  $v_1, v_2$  in  $R$  having the same parent node in the given binary tree. This condition can be true at most once
3. If there are more than two leaf nodes  $v_1, v_2$  which belongs to the required terminal set  $R$
4. Any root/sub-root having child nodes  $v_1, v_2$  which belongs to the set  $R$  cannot be contained in more than one sub-tree. This condition can be true at most once.

## 2.9 Steiner path decision Algorithm

Steiner path decision algorithm for a tree is presented in Algorithm 2. The algorithm uses a single global variable *pathlock*. Depending upon the given input tree  $T$  and as per the requirement other variables are created dynamically.

```

Steiner-path-exist(Sub-tree T){

```

```

  r: Root of T

```

```

  IF r is NULL

```

```

    return 0

```

```

  ENDFIF

```

```

  CREATE subtrees  $T_i$  and variables  $B_i$  for each subtree  $T_i$  of  $T$ 
  rooted at  $r$ 

```

```

  FOR each subtree  $T_i$  of  $T$ 

```

```

     $B_i +=$  Call Steiner-path-exist( $T_i$ )

```

```

  ENDFOR

```

```

  IF pathlock=0 and  $\sum B1+B2+..+Bi > 0$ 

```

```

    pathlock=1

```

```

    return 1

```

```

  ENDFIF

```

```

  IF pathlock=1 and  $\sum B1+B2+..+Bi > 0$ 

```

```

    return 0 \no steiner path exists

```

```

  ENDFIF

```

```

  IF  $r \in R$ 

```

```

    return 1

```

```

  ELSE

```

```

    return  $B_i$ 

```

```

  ENDFIF
}

```

## 2.10 Implementation of Steiner path exists algorithm

The algorithm works on the principle of recursion. For every root/sub-root of a tree the corresponding child nodes are considered as sub-trees of that node. These subtrees are denoted as  $T_i$ , where  $i$  denote number of sub-tree.

For every node  $r$  in  $R$  found in a sub-tree the corresponding  $T_i$  variable is incremented by 1.

Our algorithm uses hash table data structure whose size is equal to the total no of nodes present in a given tree.  $R$  is the set of required nodes which are represented with Boolean value '1' and other nodes with Boolean value '0' in the hash table. For each sub-root node  $r$ , a check is done in the hash table to find whether it belongs to  $R$  or not.

Each sub-tree  $T_i$  is traversed by recursion till it reaches the leaf node and has dynamic Boolean variable  $B_j$  which is set to 1 if a node  $r$  in  $R$  is found in the current sub-tree.

At any sub-tree if  $B1 > 0$  and  $B2 > 0$  is true then  $pathlock = 1$  which was earlier initialized to 0.

If  $T_1 > 0$  and  $T_2 > 0$  and  $pathlock = 1$  is true then no Steiner path exists which consists of all R nodes.

### 2.11 Run-time analysis of Steiner path exists algorithm

Our algorithm follows dynamic programming approach and hence requires less data structure for storing intermediate data. A single global Boolean variable  $pathlock$  is used to keep track of the nodes which belong to different sub-trees. This Boolean variable requires constant space. The time complexity for solving the Steiner path decision problem is equivalent to a single traversal of tree in our proposed algorithm. The time complexity for traversing a tree having  $n$  nodes is  $O(n)$  i.e. linear time, therefore the proposed algorithm also has  $O(n)$  time complexity.

We have used hash table as the data structure in our algorithm whose size is equal to the number of nodes of the tree. We have used the values 0 and 1 in hash table for representing the required set of terminals. Comparing the current root/sub-root requires constant amount of time  $O(1)$ . So, overall time complexity of the above algorithm is linear time.

### 3. CONCLUSION

In this paper a novel algorithm for solving the Steiner path decision problem in a binary tree is proposed.

Firstly, the characteristic of tree is defined which must be satisfied by any binary tree to have a Steiner path. The conditions defined in these characteristics are only sufficient conditions not necessary conditions.

Secondly, the algorithm performs a single traversal of the tree to find whether there exists a Steiner path or not for a given tree.

The proposed algorithm uses the principle of recursion to get the decision in linear time with a single traversal of tree. It exploits the tree characteristics to solve the Steiner path decision problem in linear time.

### 4. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

### 5. REFERENCES

- [1] Hwang F K, Richard D S. Steiner tree problems. Networks, 1992, 22(1):55-89
- [2] Winter P. Steiner problem in networks: a survey J. Networks 1987, 17:129-167
- [3] Gilbert E N, Pollak H O. Steiner minimal tree. SIAM J. pp. Math, 1968(16):1-29
- [4] Du D Z, Hwang F K. "The Steiner ratiion conjecture of Gilbert" New York: 1992; 7(1):121-135
- [5] R.M. Karp, "Reducibility among combinatorial problems", Complexity of Computer Communications, R.E. Miller and J.W. Thatcher (Eds.), Plenum Press, New York, pp. 85-103, 1972.
- [6] M. Tashakkori, P. Adibi, A. Jahanian, A. Nourollah, "Ant colony solution dynamic Steiner tree problem", Proceedings of 9th Annual Computer Society of Iran Computer Conference, pp. 465-471, 2003
- [7] W.Q. Yang, T.D. Guo, An ant colony optimization algorithm for the minimum Steiner tree problem and its convergence proof. Acta Mathematicae Applicatae Sinica, 2006, Vol. 29, No. 2, pp. 352-361(in Chinese).
- [8] Luc Luyet, Sacha Varone, and Nicolas Zufferey: "An ant algorithm for the Steiner tree problem in graphs." 2007.
- [9] Hui-min JIN, Liang MA, Zhou-mian WANG. "Intelligent Optimization Algorithms for Euclidean Steiner Minimum Tree Problem" J. Computer Engineering. 2006, 32(10):201-203.
- [10] Takahashi H, Matsuyama A. "An approximate solution for the Steiner program in graphs" J. Math Japonica, 1980.24:537-577
- [11] Fatemeh Ghadimi, Ali Nourollah "A Heuristic Algorithm for Solving Steiner Tree Problem on Urban Traffic Network" May 15-17,2012, ICCE2012, Tehran, Iran
- [12] Narendra Maharjan, Abinash Shrestha, Ashish Tamrakar and Sanjeeb Prasad Panday "Solving Minimum Steiner tree based on behavior of ants" 978-1-4673-2590-5/12/ IEEE
- [13] A. Karim Abu-Aash, Paz Carmi, Matthew J. Katz, Michael Segal "The Euclidean Bottleneck Steiner Path Problem" Proceedings of the 27th annual ACM symposium on Computational geometry pp. 440-447