An Improved Approximation Algorithm for Steiner Tree

Anil Kumar
CS13M004

Indian Institute of Technology Madras

anilcs13m@gmail.com

December 30, 2015

Overview

Introduction

- Steiner Tree

  The Steiner tree problem is one of the most fundamental NP-hard problems [2].

  INPUT:- Given a undirected connected graph G $= (V, E)$ with edges cost c:E$\rightarrow \mathbb{R}_{\geq 0}$ and a subset of nodes S$\subseteq$V, S is so called as the required vertices(terminals).

  OUTPUT:- A tree T_s that cover all the required vertices (S) with minimum cost $c(T_s) := \sum_{e \in T_s} c(e)$, tree T_s might contain some other vertices, called optional vertices(Steiner nodes).

Existing Algorithms

- Torricelli Point:-

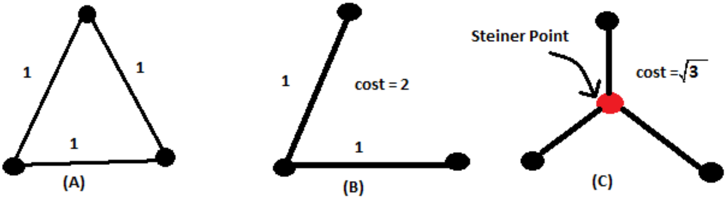

Figure: *(A) equilateral triangle with edge cost 1. (B) MST with cost equal to 2. (C) Steiner Tree with cost* $\sqrt{3}$.

- **Heuristic Algorithm**:-

Heuristic algorithm for construct a Steiner tree [3].

INPUT:- Given a undirected distance graph $G = (V, E)$ and the distance d between the vertices v_i and v_j and set of required vertices (S) as a input.
OUTPUT:- A Steiner Tree T_s and S.

Existing Algorithms continue...

- <u>Heuristic Algorithm:-</u> Give a undirected connected graph G(V,E,d) and a set of Required Points(S).

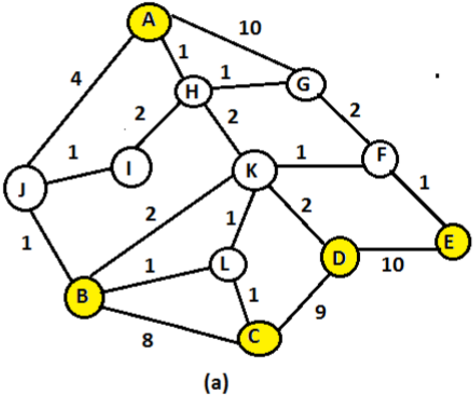

Figure: *(a) A graph G with terminals as A,B,C,D,E.*

- Heuristic Algorithm:- First take the given graph G and the required vertices(S), and construct a complete distance graph G_1=(V_1, E_1).

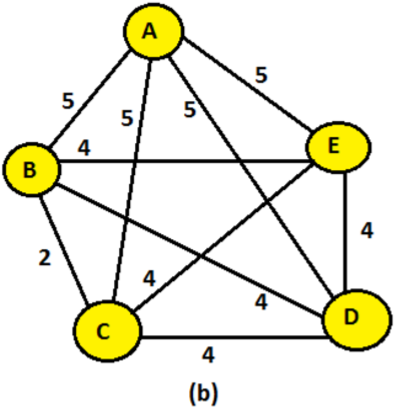

Figure: *(b) A complete graph G_1 by using terminals A,B,C,D,E.*

Existing Algorithms continue...

- Heuristic Algorithm:- From that graph G_1 find a minimum spanning tree, T_1.

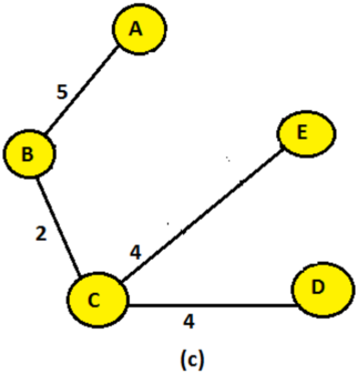

Figure: *(c) MST from complete graph G_1*

Existing Algorithms continue...

- <u>Heuristic Algorithm:-</u> Construct the sub-graph G_s, edges are according to the edges present in the minimum spanning tree from step 2.

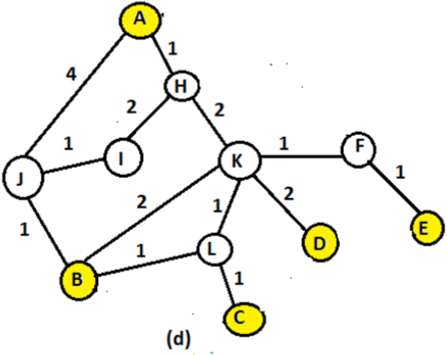

Figure: *(d) A graph G_s.*

Existing Algorithms continue...

- Heuristic Algorithm:- Find minimum spanning tree from that sub-graph G_s.

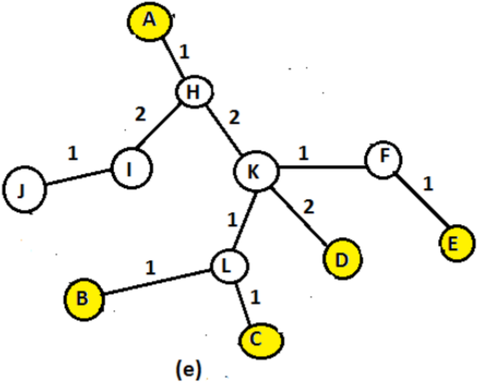

Figure: *(e) MST T_s of graph G_s.*

- Heuristic Algorithm:- Now delete all the unnecessary edges which are not connecting the required vertices (S), final tree is so called as the Steiner tree.

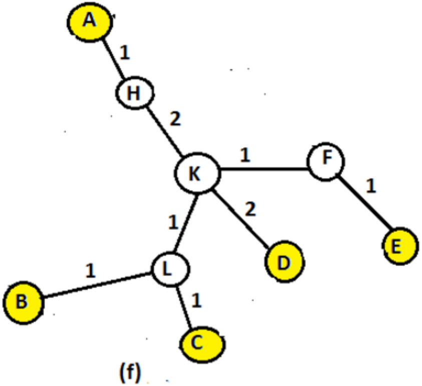

Figure: *(f) Steiner tree of graph G*

Running Time for Heuristic Algorithm

- Running Times for Each Steps:- Running time taken by each steps are as.
  - Step 1 could be done in $O(|S||V|^2)$ time.
  - Step 2 could be done in $O(|S|^2)$ time.
  - Step 3 could be done in $O(|V|)$ time.
  - Step 4 could be done in $O(|V|^2)$ time.
  - Step 5 could be done in $O(|V|)$ time.

  Here in these steps only first step is taking more running time then the other steps. So, we can say that overall running time complexity for this algorithm will be $O(|S||V|^2)$.

Analysis of Approximation Ratio

- 2-Approximation:-

$$\text{cost(T)} \leq 2.\text{cost(OPT)}$$

- Tight Approximation Ratio:-

$$\text{cost(T)} \leq 2.\text{OPT}\left(1 - \frac{1}{|S|}\right)$$

Some more tight bound

$$D_H \leq 2.D_{MIN}\left(1 - \frac{1}{l}\right)$$

where, T is Steiner tree, and OPT is optimal Steiner tree, S is the terminals, l is the leaves in tree, D_H is the distance of Steiner tree, D_{MIN} distance of minimum Steiner tree.

Proposed Heuristic Algorithm

- preliminary:- What are the requirement for our new heuristic algorithm [1].

    - A optimal path.

    - All-pairs shortest path problem.

    - Dijktra's algorithm.

Proposed Heuristic Algorithm...

- Heuristic Algorithm:- Previous algorithm is taking 5 steps with running time $O(|S||V|^2)$ for all graph.

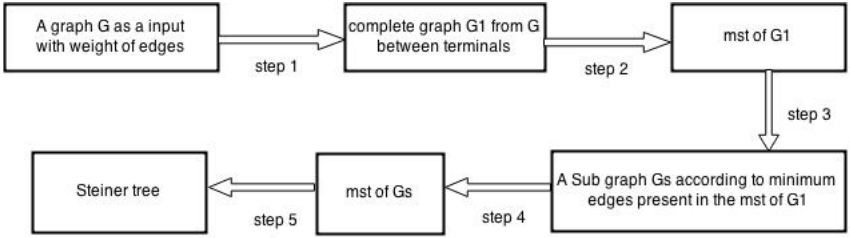

Figure: *All 5 steps of Heuristic algorithm*

- Proposed Heuristic Algorithm:- Our proposed algorithm with a better running time.

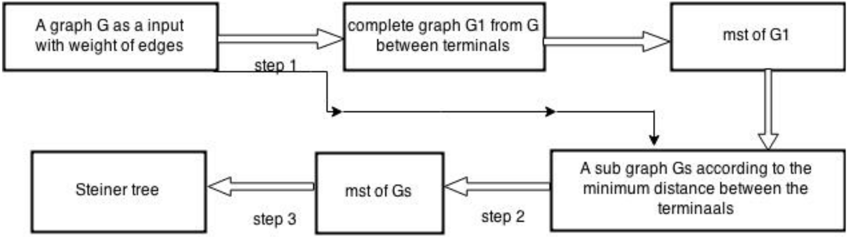

Figure: *All 3 steps of Our heuristic algorithm*

- New Heuristic Algorithm:- Give a undirected connected graph G(V,E,d) and a set of Require Points(S).

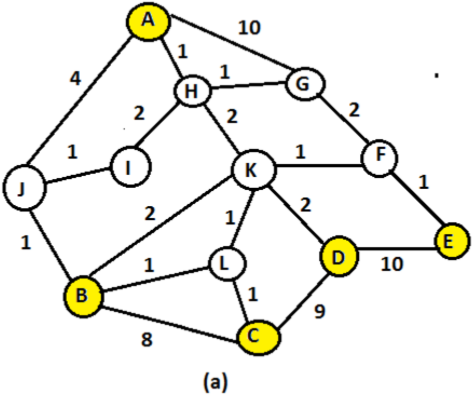

Figure: *(a) A graph G with terminals as A,B,C,D,E.*

- Step 1:- Construct a subgraph G_s, from that graph G. This subgraph having the shortest path between each and every pairs of terminals. This construction is done by running the dijktra's algorithm by considering each terminal as a source.

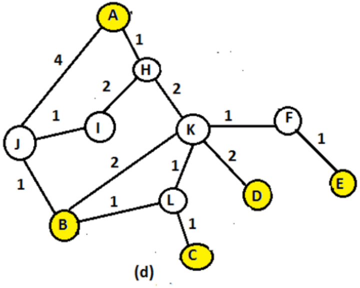

Figure: *(a) A subgraph G_s.*

Proposed Heuristic Algorithm...

- Step 2:- Find minimum spanning tree from that sub-graph G_s.

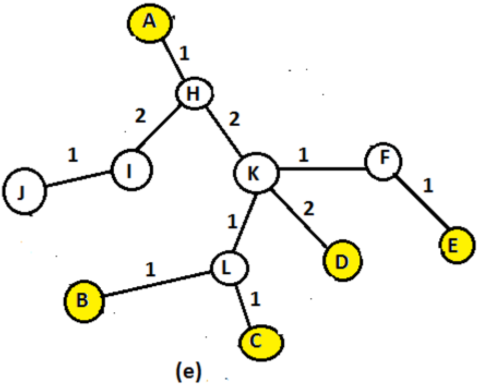

Figure: *(e) MST T_s of graph G_s.*

Proposed Heuristic Algorithm...

- Step 3:- Now delete all the unnecessary edges which are not connecting the required vertices (S), final tree is so called as the Steiner tree.

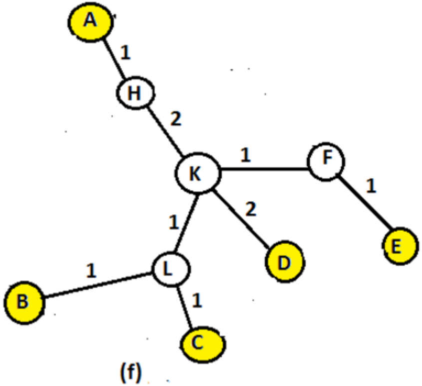

Figure: *(f) Steiner tree of graph G*

Theorem and Lemma

Theorem

The dijktra's algorithm finds the optimal path between the vertices of the graph. Furthermore it discovers path in the increasing order of weight [5].

Lemma

A path $(v \rightsquigarrow u)$ is called optimal iff no other path $(v \rightsquigarrow' u)$ in between (v, u) is called optimal.

$$||v \rightsquigarrow' u|| \leq ||v \rightsquigarrow u||$$

Lemma

In a graph $G = (V, E)$ such that $S \subseteq V$, where S is the terminals of the graph G, then we can stated that $O(|S||V|log|V|) \leq O(|V|^2 log|V|)$.

Running Time of Proposed Heuristic Algorithm

- Running Times for Each Steps:- Running time taken by each steps are as.
  - Step 1 could be done in $O(|S||V|log|V| + |E||S|)$ time.
  - Step 2 could be done in $O(|E| + |V|log|V|)$ time.
  - Step 3 could be done in $O(|V|)$ time.

Here in these steps only first step is taking more running time then the other steps. So can say the overall running time complexity for this algorithm will be $O(|S||V|log|V| + |E||S|)$.

Running Time Conclusion

- Total running time of this algorithm $O(|S||V|log|V| + |E||S|)$.
- If order of edges in the graph $O(E) \leq |V|log|V|$, than running time of our heuristic will be $O(|S||V|log|V|)$.
- Even for edges of order $O(E) < |V|^2$, this is better the previous heuristic algorithm.
- For $O(E) = |V|^2$, running time will be same as the previous one but we are solving this in 3 steps only.

Input Instances E TestData

Name	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert S\rvert$	DC	Opt
e01.stp	2500	3125	5	Ps	111
e02.stp	2500	3125	10	Ps	214
e03.stp	2500	3125	417	Ps	4013
e04.stp	2500	3125	625	PS	5101
e05.stp	2500	3125	1250	Ps	8128
e06.stp	2500	5000	5	Ps	73
e07.stp	2500	5000	10	Pm	145
e08.stp	2500	5000	417	Pm	2640
e09.stp	2500	5000	625	Pm	3604
e10.stp	2500	5000	1250	Pm	5600

Table: Input Instances E [4]

Input Instances E TestData...

Name	$\|V\|$	$\|E\|$	$\|S\|$	DC	Opt
e11.stp	2500	12500	5	Pm	34
e12.stp	2500	12500	10	Pm	67
e13.stp	2500	12500	417	Pm	1280
e14.stp	2500	12500	625	Pm	1732
e15.stp	2500	12500	1250	Ps	2784
e16.stp	2500	62500	5	Ph	15
e17.stp	2500	62500	10	Ph	25
e18.stp	2500	62500	417	NPh	564
e19.stp	2500	62500	625	Pm	758
e20.stp	2500	62500	1250	Ls	1342

Table: Input Instances E [4]

Computational Results on E TestData

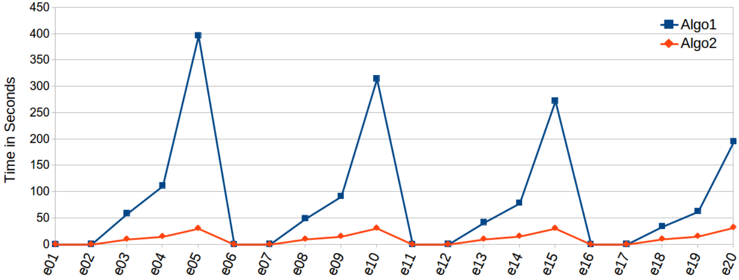

Figure: *c1 computational results in instances E*

Input Instances P4E TestData

Name	$\|V\|$	$\|E\|$	$\|S\|$	DC	Opt
P455	100	4950	5	Ps	1138
P456	100	4950	5	Ps	1228
P457	100	4950	5	Ps	1609
P458	100	4950	5	Ps	1868
P459	100	4950	5	Ps	2345
P460	100	4950	5	Ps	2959
P461	100	4950	5	Ps	4474
P463	200	19900	20	Ps	1510
P465	200	19900	40	Ps	3853
P466	200	19900	100	Ps	6234

Table: Input Instances P4E [4]

Computational Results of P4E TestData

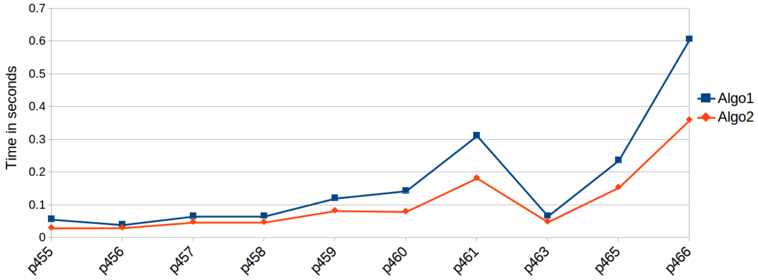

Figure: *c2 computational results in instances P4E*

Conclusion

- Running Times Comparison of Both Algorithms :-
- In figure c1 and c2, $Algo1$ is the previous one and $Algo2$ is the proposed one.
- $Algo2$ is taking the less running time than the $Algo1$ in all the cases of input instance.
- More is the value of OPT more is the difference in the running time.

Here we conclude in all the cases our proposed algorithm is taking less time as comparison of previous one, with a new running time complexity of order $O(|S||V|log|V|)$ and worst case will be $O(|S||V|log|V| + |E||S|)$.

REFERENCES

- Thomas H Cormen, Charles E Leisersen, Ronald L Rivest, Clifford Stein, et al. "Introduction to algorithm, Volume 2." *MIT Press Cambridge,*2001.
- P. Winter, F.K. Hwang, D.S Richards. *The Steiner Tree Problem.* Elsevier, 1992.
- G. Markowsky, L. Kou and L. Berman. *A Fast Algorithm for Steiner Tree.* Acta Informatica, 15: 141-145, 1981.
- A. Martin, S. Voss, and T. Koch. *SteinLib Testdata Library.* online document at http://steinlib.zib.de/download.php, 2001.
- Dijktra's. *A Note on two Problems in with Graphs.* Numerische mathematic, (1): 269-271, 1959.

Thank you