BTP Report
Steiner Trees and Steiner Forests

Supervisors:
Prof. Naveen Garg
Prof. Amit Kumar

Submitted By:
Ankit Garg 2007CS10155
Arpit Goel 2007CS10162
Sahil Singla 2007CS10181

Contents

1 Introduction

The Steiner Tree problem is one of the classical NP-hard problems. Here we are given an undirected graph G with non-negative costs on edges and a subset of vertices, called terminals. The problem is to find a minimum cost subgraph connecting the terminals. The remaining vertices are known as Steiner vertices. Note that an optimal solution must be a tree, with no Steiner vertex as a leaf. We can assume that the edge costs obey triangle inequality, by solving the problem on the metric closure of G.

2 Approximation Algorithms for Steiner Tree

MST on the terminals is within twice the optimal for the metric Steiner Tree problem. This is gives an easy 2-approximation algorithm for the problem. Almost all the further algorithms use the idea of full components. A full component is a Steiner Tree on a subset of terminals with all those terminals as leaves. Robins and Zelikovsky [1] gave a combinatorial 1.55-approximation algorithm. Very recently, Byrka et al. [2] reduced the factor to 1.39 using an iterative randomized rounding approach. They even prove a bound of 1.55 on the Integrality Gap of the Directed Cut Relaxation. In [1], they use the concept of the gain of a component with respect to a terminal Steiner Tree, and also of the loss of a component.
$gain_T(K) = cost(T) - cost(T[K])$, where $T[K]$ is the minimum cost subgraph of $T \cup K$ containing K and spanning all the terminals. $loss(K)$ is defined as the minimum subgraph in which each Steiner vertex is connected to some terminal. They add compoenents based on the $gain/loss$ values, while commiting only the loss of a component. This way we don't loose that much even if we make a bad choice. In [2], Byrka et al. prove that the solution given by Robins' algorithm is infact within 1.55 of the fractional optimal of k-DCR, for any k.

Borchers and Du [3] showed that for achieving a good approximation factor, it is sufficient to restrict our attention to k-restricted Steiner trees. A k-restricted Steiner tree is one in which all the full components have k or fewer terminals. If OPT_k is the optimal k-restricted Steiner tree, and OPT is the optimal Steiner tree, then in [3], they show that $OPT_k/OPT \leq 1 + 1/\log_2 k$.

2.1 Iterative Randomized Rounding Algorithm

Let $C_1, C_2, ..., C_h$ be the directed k-components. Then the following is the k-DCR relaxation for the k-restricted Steiner Tree problem

$$
\begin{aligned}
\min \textstyle\sum_j c(C_j)x_j & \\
\textstyle\sum_{C_j \in \delta_k^+(U)} x_j \geq 1 & \qquad \forall U \subseteq R\backslash r, R \neq \phi \\
x_j \geq 0 & \qquad \forall j = 1, ..., h.
\end{aligned}
$$

The algorithm is very simple. We solve the LP. Let x be the optimal solution. We sample a component C_j with probability proportional to x_j, contract it, and then do this iteratively. An important of analysis is the following lemma

Bridge Lemma[2] : Let T be a terminal spanning tree and let x be a feasible solution to the k-DCR. Then $c(T) \leq \sum_j x_j gain_T(C_j)$.

The lemma helps in bounding the decrease in the cost of the terminal spanning tree in consecutive iterations, which leads to a simpler analysis of the 1.5 factor for a slightly different version of the algorithm, in which we stop after a certain number of iterations and output the terminal spanning tree along with the contracted components. The analysis of 1.39 factor for the original algorithm uses some clever randomization ideas. It uses the idea of a random bridge of a component with respect to a terminal spanning tree to simulate the random sampling of the components by an almost uniform sampling of the edges of a terminal spanning tree. This uniform sampling further induces deletion of edges from the optimal Steiner Tree such that each edge is deleted after an expected $ln(4)$ steps and also that the remaining tree along with the contracted components connnects the terminals. The sum of this series of Steiner trees is then an upper bound on the sum of fractional optimal LP values over iterations, and thus gives an $ln(4)$ factor.

2.2 Relation between BCR and DCR

To obtain an approximation of $1.39(1+1/\log_2 k)$, we need to solve the k-DCR LP which has $O(n^k)$ variables. Thus it is costly to obtain an approximation of $1.39 + \epsilon$. It should be interesting to see if these techniques can be applied somehow to the Bidirected Cut Relaxation, though it seems difficult as Polzin in [4], proves that for any k, k-DCR is strictly stronger than BCR. An approach to splitting flows of a BCR solution motivated solving the Steiner Tree Problem on an approximate tree, in which the root is a terminal, the root along with the Steiner vertices and the remaining terminals are connected to the leaves of this tree. There were connections between this problem and the Group Steiner Problem on trees, and also between the 1-level tree case and the Facility Location Problem. These connections need to be explored further.

3 BCR for quasi-bipartite graphs

Although there has been little progress on using BCR in the approximation algorithms for general graphs, BCR has been extensively studied for quasi-bipartite graphs. These are the class of graphs which do not have Steiner-Steiner edges. The drop in difficulty for the problem in case of such graphs (although it still remains NP-hard) is summed up by the following lemma :

Lemma[5] : Let MST denote the minimum spanning tree on terminals. If $MST(R \cup v) < MST$ $\forall$ Steiner v, then MST is the optimal Steiner tree. Infact, $MST = OPT_f(BCR)$.
This is not true for general graphs. In [5], they use the idea of L_1-embedding to give a relaxation whose dual is equivalent to the BCR. The relaxation aims at maximizing an objective function $\gamma(z)$ for an embedding such that the edge lengths are only shortened. The objective function is a lower bound on the cost of an optimal Steiner tree in a λ-simplex (sum of coordinates is λ). They give a primal dual algorithm which either returns a valid embedding γ such that $MST = \lambda(z)$ or returns a Steiner vertex v such that $MST(R \cup v) < MST$.
Using the lemma, we can prove that the local search gives a 3/2-approximation algorithm for quasi-bipartite graphs, and using the work in [5], proves a 3/2 bound on the integrality gap of BCR for quasi-bipartite graphs. Also they introduce a filtering technique in which they reduce the cost of required-required edges by a factor of 4/3 and then run the local search. This gives a 4/3-approximation algorithm for the Steiner tree for quasi-bipartite graphs and also places a bound of 4/3 on the integrality gap of BCR for such instances. Skutella's example [8] gives a lower bound of 8/7.
Chakrabarty et al. [6] prove that DCR is equivalent to BCR for the case of quasi-bipartite graphs.

3.1 Bound of 1.28 for BCR for the case of quasi-bipartite graphs, [7]

Chakrabarty et al. [7] give a short proof of the upper bound of 1.55 on the integrality gap of DCR. They also give a upper bound of 1.28 for the case of quasi-bipartite graphs, which with a slighlty better analysis can be improved to 1.23. This also implies an upper bound of 1.23 on the integrality gap of BCR for quasi-bipartite instances (DCR and BCR are equivalent for such instances). DCR can also be written as follows :

$$min \sum_K C_K x_K :$$

$$\forall \phi \neq S \subseteq R : \sum_{K:K\cap S\neq\phi} x_K(|K \cap S| - 1) \leq |S| - 1$$
$$\sum_K x_K(|K| - 1) = |R| - 1$$
$$\forall K : x_K \geq 0 \tag{1}$$

Where $K \subseteq R$ and R is the set of required vertices. Define $F(K)$ as the minimum cost full component on terminal set K. Let $Loss(K)$ be the loss of the full component $F(K)$ ie. a minimum cost subset of edges of $F(K)$ which connect every steiner vertex to some terminal. Let $loss(K)$ be the cost of edges in $Loss(K)$. Define $LC(K)$ as the loss contracted full component of K, ie. the $F(K) - Loss(K)$ The algorithm given in [7] is:

Algorithm RLC

1. Let T_1 be a minimum spanning tree of the induced graph $G[R]$
2. $x \leftarrow \; Solve(1)$
3. for $1 \leq i \leq t$ do
4. Sample with replacement a single component K_i as follows: with probability x_K/M it is the full component K, with probability $1 - \sum_K x_K/M$ it is the empty set
5. $T_{i+1} \leftarrow MST(T_i \cup LC(K_i))$
6. end for
7. Output any Steiner Tree in $ALG := T_{i+1} \cup_{i=1}^{t} Loss(K_i)$

Using the bridge lemma, the authors in [7] prove:

$$E[c(ALG)] \leq lp^*(1/2 + 3/2e^{-t/M} + t/2M) \tag{2}$$

Where $E[c(ALG)]$ is the expected value of the cost of the answer of RLC algorithms, lp^* is the optimal fractional value of the $LP(1)$, $M \geq \sum_K x_K$ such that $t = Mln3$ is integral. Substituting the values we get a $1+(ln3)/2 \sim 1.55$ approximation.
In addition, they have shown for quasi bipartite graphs the following holds:

$$E[c(ALG)] \leq lp^*(1 + e^{-t/M}) + loss^*(t/M - 1 - e^{-t/M}) \tag{3}$$

Now observing that the value minimizes at $t/M = \alpha$ where $\alpha = 1 + e^{-\alpha}$, they obtain an $\alpha \sim 1.28$ approximation for the quasi bipartite case.

4 An LP similar to the subtour elimination for Steiner Tree problem, [9]

We keep a variable y_v for each Steiner vertex and set it to 1 if that vertex is selected in the optimal Steiner Tree. Once we have the selceted vertices, the optimal Steiner tree is just a minimum spanning tree on the required and the selected Steiner vertices. We can write the following LP :

$$\begin{aligned} min \sum_{e} c_e x_e \\ x(E(S,S)) \leq \sum_{v \in S} y_v - 1 \forall S : S \cap R \neq \phi \\ x(E(S,S)) \leq y(S-k) \forall \phi \neq S \subseteq N, and\ k \in S \\ y_v = 1 \forall v \in R \\ 0 \leq y_v \leq 1 \forall v \in N \\ x_e \geq 0 \end{aligned} \tag{4}$$

In [9], they prove that this LP is equivalent to the BCR. It might be that it is easier to work with this LP instead of BCR.

4.1 A proof of integrality of subtour elimination LP for MST problem

While trying to do a primal dual on the gerneralized subtour elimination LP, we found a proof of the integrality of subtour elimination for MST. An attempt at extending this proof to the generalized subtour to obtain similar results could be interesting. Write the subtour elimination LP as :

$$\begin{aligned} max \sum_{e} -c_e x_e \\ x(E(S,S)) \leq |S| - 1 \forall \phi \neq S \subset V \\ x(E(V,V)) = |V| - 1 \\ x_e \geq 0 \end{aligned} \tag{5}$$

Dual is :

$$min \sum_{S} z_S(|S|-1)$$

$$\sum_{S:e\in E(S,S)} z_S \geq -c_e \forall e \in E$$
$$z_S \geq 0 \forall \phi \neq S \subset V \quad (6)$$

Now let T be a minimum spanning tree. Then we produce a dual z such that value of $z = -$ cost of T. This will prove the integrality. Let e be the costliest edge in T. Then take $z_V = -c_e$. Breaking this edge divides T into T_1 and T_2. Then let assign $z_{T_1} = c_e-$cost of costliest edge in T_1 and similarly assign z_{T_2} and so on. Note that each edge is tight and each set S which is assigned a positive value(other than V which obviously has $|V|-1$ edges) has $|S|-1$ edges, and thus value of $z = -$ cost of T. Note that the property of T that if we add any edge e to T , then T must contain an edge e' with $c_{e'} \leq c_e$ implies that z is feasible.

5 Steiner Forest

The lack of an alternative 2-factor algorithm for the Steiner Forest problem is a little astonishing. The added difficulty here is how to decide the composition of the components of the final solution. The following approaches might give an alternative 2-factor algorithm :

1. **Shortest s-t Path:**
   (a) While requrement function not satisfied for all vertices
   (b) Find shortest s-t path, ie. the shortest path connecting a required pair of vertices
   (c) Contract the path found
   (d) Change the connectivity requirement function for those 2 vertices
2. **Local Search:**
   (a) Find the MST for each required set. Call each MST as a component.
   (b) while $\exists\, C_i, C_j$ such that$(cost(MST(C_i \cup C_j)) < cost(C_i) + cost(C_j))$ Replace C_i, C_j by $MST(C_i \cup C_j)$
3. **Using Steiner Tree:**
   (a) $Forest = \phi$
   (b) While Connectivity not satisfied, for each required set, find the Steiner Tree (as near to optimum)
   (c) Of above, find the min cost Steiner Tree, call it S
   (d) $Forest = Forest \cup S$
   (e) Contract S into a new Steiner vertex, goto (b)

5.1 Greedy Heuristics for Steiner Forest Problem

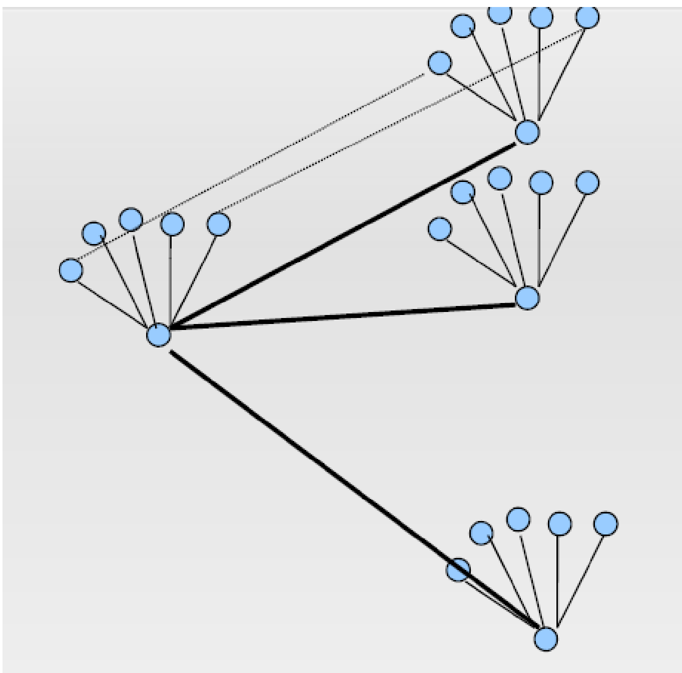
We try the following Algorithm:

1. While there is an (s_i, t_i) pair which is not yet connected

2. Choose the (s_i, t_i) pair which has the shortest distance path between them

3. Zero out this path, ie. make its length as 0 and include the path in the solution

While this algorithm looks promising for the Steiner Tree problem, as the solution we obtain using it is same as the Minimum Spanning Tree solution, which clearly is a 2-Approximation. But it proves to be quite notorious for Steiner Forest. First we provide an example which gives a ratio worse than 4, ie. certainly contradicting the possibility of a 2-Approximation for the problem using the above Greedy Approach. Later we provide a proof of $O(\log(n))$ lower bound on the approximation factor of this Algorithm, where n is the number of nodes in the graph.



In the above figure, the dotted edges are of cost 4, the solid (non bold) edges are of cost 1, and the bold solid edges are of cost 2 each. (s_i, t_i) pairs are the vertices at the ends of dotted edges. Now let each cluster have k such vertices and there be a total of $O(k)$ clusters. Total path length picked by the above algorithm would be $4k^2 * k = 4k^3$. Basically our Algorithm would pick all the dotted edges. Whereas cost picked up by OPT would be $k * k + 2k + k$, ie. all the solid (non bold and bold) edges. It's clear the cost picked by our Algorithm would be $O(4)$ times the OPT in this example.

Proof of $O(\log n)$ Lower Bound
Let us consider a constant degree $\log n$ girth graph $G = (V, E)$, where $|V| = n$. Such a graph does exist as is shown in [19]. Let T be a spanning tree of the graph and E_T denote the set of all edges of that tree. Since each vertex has a constant degree, both $|E| = O(n)$ and $|E - E_T| = O(n)$. The cost of every edge of E_T is set to 1. Next, consider the subgraph $(V, E - E_T)$ and let M be a maximal matching in that subgraph. Clearly, $|M|$ is also $O(n)$. Each matched pair forms the (s_i, t_i) for our example; thus there are $O(n)$ such pairs. The (s_i, t_i) edge cost is set to $(\log n)/2$. The cost of each of the remaining edges of $E - E_T$ is set to 1.

Clearly, T being the spanning tree connects all (s_i, t_i) pairs. Thus the optimal solution is of cost $O(n)$. Let our Greedy Algorithm connect the pairs in order $(s_1, t_1), (s_2, t_2) \cdots (s_n, t_n)$. The graph under consideration has $\log n$ girth and thus initially the direct edge between (s_1, t_1) is the path of minimum cost between them. The algorithm selects that edge in the solution and reduces it to cost zero. At an intermediate stage, suppose we are trying to connect (s_j, t_j). The number of edges in the girth in which (s_j, t_j) belongs is $O(n)$ and in the worst case at most every alternating edge in that girth might have been reduced to cost zero because of the contraction of the pairs $(s_i, t_i), i = 1, 2.., j - 1$. Thus the minimum cost path between s_j and t_j is still the direct edge between them. Hence the total cost of our greedy solution is $O(n) * (\log n)/2$. Thus, the greedy algorithm provides a $O(\log n)$ approximation factor for this instance.

5.2 An alternate LP for Steiner Forest problem?

We tried to write an alternative LP for the Steiner Forest problem as the well known Cut Relxation has an integrality gap of 2. The idea was to write something similar to the BCR or the generalized subtour elimination LP for Steiner tree, so that we could have that the LP is integral for the MST case. However we were unable to do so using polynomial number of variables. We had a variable $z_u v$ for all pairs of terminals u, v, to indicate that v is a root in the Steiner forest (direct each component towards some root). But there was difficulty in encoding the number of edges that must go out of a cut (which must be the number of roots outside that have a descendent inside the set). We were able to write the LP with an exponential number of variables, but it remains to be seen if this can actually be solved in polynomial time. We kept a variable $z_{S v}$ for all sets S and all terminals v to indicate if

v is a root and if S contains a descendent of v.

$$
\begin{aligned}
min \sum_e c_e x_e \\
x(\delta(S)) \geq \sum_{u \notin S} z_{Su} \forall S \\
z_{\{s\}u} = z_{\{t\}u} \forall s, t \in S_i, \forall u \\
z_{Tu} \geq z_{Su} \forall S \subseteq T \\
\sum_u z_{\{s\}u} = 1 \\
z_{Su} \geq 0 \\
0 \leq x_e \leq 1
\end{aligned}
\tag{7}
$$

6 Prize Collecting Steiner Forest Problem

In this problem, we are given an undirected graph $G = (V, E)$ with costs on edges, and penalties associated with pairs of vertices. The goal is to find a forest minimizing the costs of edges used and the sum of penalties occured for pairs not connected in the forest.

$$
\begin{aligned}
min \sum_e c_e x_e + \sum_{i,j} \pi_{ij} z_{ij} \\
x(\delta(S)) + z_{ij} \geq 1 \ \forall i, j \ , \forall S \ separating \ i, j \\
x_e \geq 0 \ \forall e \\
z_{ij} \geq 0 \ \forall i, j
\end{aligned}
\tag{8}
$$

A simple rounding technique gives a 3-approximation as follows. Let (x, z) be the optimal fractional solution. For all pairs i, j s.t. $z_{ij} \geq 1/3$, round the z_{ij}'s to 1, i.e. we will not try to connect these nodes in our solution. Let Q be the set of pairs that remain. Then we have constraints like :

$x(\delta(S)) \geq 2/3 \ \forall i, j \in Q \ , \forall S \ separating \ i, j$

Then $3x/2$ is a feasible solution to the Steiner Forest problem instance in which we are required to connect the pairs Q. And by Goemans-Williamson primal dual procedure for Steiner Forest, we can find a forest connecting the pairs Q in cost $\leq 2 * 3x/2 = 3x$. This gives a 3-approximation as penalties paid for non-connecting vertices are within thrice the z_{ij} values. Jain and Haijaghayi[10] introduced the problem and gave a primal dual 3-aproximation for the problem. They also gave a 2.54 randomized rounding algorithm for the same.

6.1 A 2 factor approximation using Fractional Decomposition technique

Chekuri and Shepherd[11] observerd that the decomposition theorem of Bang-Jenson[12] implied a 2-approximation for the Prize Collecting Steiner Tree problem. Bang-Jenson's result says that given an Eulerian undirected graph, with a root r, and an integer k, we can find k edge-disjoint trees in the graph such that r appears in each tree and any vertex $v \neq r$ appears atleast $min(k, \lambda(r, v)/2)$ times, where $\lambda(r, v)$ denotes the connectivity of r to v. We thought that if we can get a similar looking theorem for the forest case, then it would imply a 2-approxmation for Prize Collecting Steiner Forest problem. In other words, we explored if this result was true : Given an Eulerian undirected graph $G = (V, E)$, and an integer k, we can find k disjoint forests such that for all $u, v \in V$, u, v are connected in atleast $min(k, \lambda(u, v)/2)$ of the forests. But is turned out to be false as shown by the example in figure 1.

In the following graph, $\lambda(u, v) = 4$, and between all other vertices connectivity is 2. If we are asked to find 2 edge-disjoint forests satisfying the conditions, then it can be shown that we will require a spanning tree and a $u - v$ path. But after removing a spanning tree, no $u - v$ path remains. However a fractional decomposition can be done as shown in figure 2.

A related question is of decomposing a graph into trees in accordance with the Bang-Jenson's result, but laminarly. Formally, if the connectivities to the root are $2\lambda_1 \leq 2\lambda_2 \leq \ldots\ldots \leq 2\lambda_k$, then finding λ_k trees such that the vertices with connectivity λ_1 appear in the first λ_1 trees, the ones with connectivity λ_2 appear in the first λ_2 trees and so on. Note that the first λ_1 trees will be spanning trees. But the graph $K_{2,4}$ is a counterexample for this too. Then the question arises whether we can obtain such a fractional

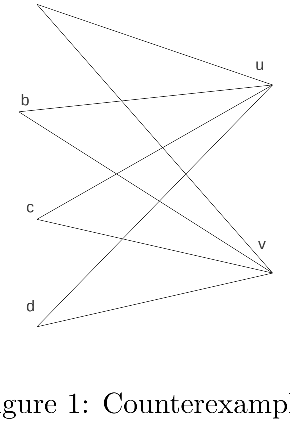


Figure 1: Counterexample

decomposition, because the decomposition for $K_{2,4}$ shown above works for this as well. But we show that such a fractional decompostion is not possible with the help of example in figure 3. It is a multigraph in which each edge is present twice. The connectivites of all the u_i's and v_i's to the root r is 4, and the connectivity of w to r is $2k$. Thus we have to find a fractional decompostion into trees, $\sum_{i=1}^{l} w_i T_i$, with $\sum_i w_i = k$ such that a total weight of 2 are spanning trees and the are $r - w$ paths. Now a spanning tree will have $2k + 1$ edges, and a $r - w$ path will have 3 edges. Thus the total fraction of edges used will be $(2k+1) * 2 + (k-2) * 3 = 7k - 4$, whereas the total fraction of edges available to us is just $6k$. Thus a fractional laminar decomposition is not possible for the graph in figure 3. For the graph G in figure 3(with edges doubled), we haven't been able to find a fractional decomposition into forests such that for all $u, v \in G$, they are connected in a fraction $\lambda(u,v)/2$ of the forests.

The following fractional decomposition for the Prize Collecting Steiner Forest problem will also lead to a 2-approximation. Consider a solution to the

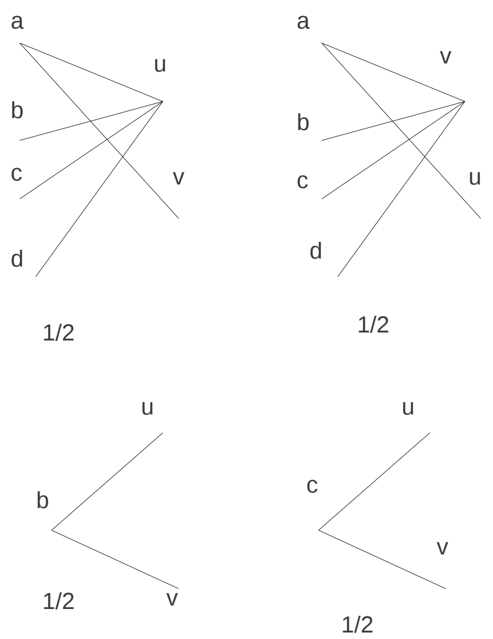


Figure 2: Fractional decomposition into forests

LP relaxation of Prize Collecting Steiner Forest. Here, z_{ij} essentially denotes fractional connectivity between vertices i and j. Double up the fractional value on each edge e to $2x_e$. The modified graph has cost at most twice that incurred by the edges in the LP solution. In this graph, the connectivity between i and j will be $2z_{ij}$. Now, for this particular pair, the penalty incurred by the LP solution is $\pi_{ij}(1-z_{ij})$. In order to bound the approximation ratio by 2, it is sufficient to keep the penalty for this pair in our solution to at most $2\pi_{ij}(1-z_{ij})$. We need to find a weighted packing of a family $\mathcal{F}$ forests $F_1, F_2, F_3, \ldots F_k$ in the graph such that vertex pair (i,j) are connected for a total weight of $2z_{ij} - 1$. Note that we need not connect pairs i,j for which $z_{i,j} \leq 1/2$.

We show this for a very special case when in the optimal fractional solution the z_{ij} values are either 1 or $\lambda, \lambda \geq 1/2$. In this case also we assume that there is just one 1 island. That is the there is a set Q s.t. $z_{ij} = 1$ for all $i,j \in Q$, and $z_{ij} = \lambda$ for all other pairs i,j. Then consider a vertex r in Q as the root and then apply the Bang-Jenson's decomposition theorem to the graph obtained by giving edge e fractional value of $2x_e$. Then we can find a weighted decomposition into trees, $\sum_{i=1}^{l} w_i T_i$, $\sum_i w_i = 1$ s.t. each vertex in Q appears in all of the trees, and the other vertices appear in λ fraction of the trees. Now consider a pair i,j. If one of i,j is in Q, then certainly they are together in $2\lambda_{ij} - 1$ of the trees. If i,j both are not in Q, then they

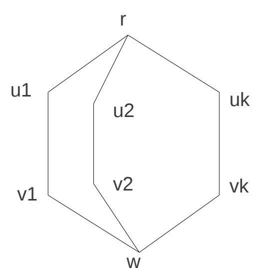


Figure 3: Counterexample for fractioanl laminar decomposition

must be together in atleast $2\lambda - 1$ of the trees, since they both are in λ of the trees, and the total weight of the trees is 1.

6.2 Primal Dual approach giving 2-approximation ?

The problem with the primal dual approach of Jain and Haijaghayi is that they charge the cost of connecting pairs to the duals of unconnected pairs also. A possible approach might be to decrease the potential given to a $s_i - t_i$ pair which is not connected in Jain's final solution such that s_i just occurs as part of an inactive leaf. Then distribute the potential to the connected components containing s_i or t_i. Another approach might be to distribute the potential π_{ij} into π_i and π_j, and then run the Goemans-Williamson procedure for Prize Collecting Steiner Tree. The question is that is there some way to distribute the potentials that gives a 2-approximate solution.

7 Steiner Network Problem with Single Source (SNPSS)

Given an undirected weighted complete graph G with infinite copies of each edge, and a source s, the objective is to find a min cost subgraph such that $\lambda(v, s) \geq r(v, s)$ for all vertices v.

This problem is a special case of the Steiner Network problem where one source is fixed for all the pairs and also there is no upper bound on the number of copies of each edge. Thus a 2-approximation factor algorithm using Jain's method is possible. But no combinatorial algorithm is known for this problem. The problem has its importance as it can be used in Multicasting and various other Network design problems. Theoretically, a 2-factor combinatorial algorithm for SNPSS will give insight and help in achieving the

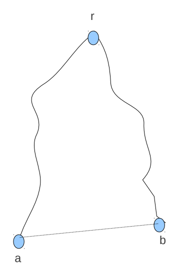


Figure 4: Single source uniform requirement

ultimate goal of designing a 2-factor combinatorial algorithm for the Steiner Network problem.

A natural greedy algorithm for SNPSS is to iteratively calculate the shortest (s, v) path for all vertices v such that their connectivity is augmented and then to select the shortest such path in the solution. This algorithm gives factor $2 - \epsilon$ for the simple case in C_k where each vertex has requirement 2 and all edges have equal weights. Optimal solution will pick a cycle whereas the algorithm selects 2 copies of P_k. As shown below, we have proven a factor 2 by this algorithm for the uniform connectivity requirement case. But we haven't been able to either prove worst case factor-2 or construct an example of factor more than 2 for this Greedy Algorithm. The ideas used and the difficulties in using them to prove a factor 2 have been also discussed below.

A combinatorial algorithm in an un-weighted complete graph exists for SNPSS [13]. This solution is not restricted to single source and works even with different requirements with different $\{s_i, t_i\}$ pairs. Another algorithm by Chou and Frank [14] works for simple graphs i.e. where selecting parallel edges is not allowed.

7.1 Approx factor 2 for Uniform connectivity requirement-k

For uniform connectivity requirement k, this algorithm works very similar to the Prims Algorithm for finding the minimum spanning tree in a weighted graph. It actually just picks k copies of the tree generated by Prims algorithm i.e. the cost of solution is $k * cost(MST)$. Let us call the solution of the greedy algorithm G_r. Now consider the optimal solution OPT to this problem. If we bi-direct the edges of OPT, root s is connected to each vertex v with at least $2 * r(s, v) = 2k$ paths. Now using Nash Williams and Tutte result, any $2k$-edge connected graph consists of k-edge disjoint spanning trees. This gives that $2 * cost(OPT) \geq cost(k\ spanning\ trees) \geq (k\ MST) \geq cost(G_r)$. Hence we will get a 2-approximation algorithm if we prove the following claim.

Claim: The solution of Greedy Algorithm G_r selects k copies of the MST generated by Prims Algorithm starting at root s.
Proof by induction: Label the edges picked by the Prims algorithm in the sequence they were picked.
IH: For the first i iterations of the Greedy Algorithm, there exists a j such that Greedy has picked one or more copies of first j edges of the Prims algorithm.
Base Case: The first edge picked by Prims algorithm and the Greedy Algorithm is same as both tries to connect shortest path vertex to s.

Induction Step:
Proof by contradiction: Suppose in the (i+1)th iteration some edge e(a,b) is picked which is not from the edges 1,2,, j+1 of Prims Algorithm. Case 1: At least one vertex v = a or b is not adjacent to first j+1 edges of Prims Algorithm. Here we get a contradiction as Greedy augments vs connectivity by picking a path that is more expensive then the (j+1)th edge picked by Greedy Algorithm. Case 2: Both a and b are adjacent to first j+1 edges of Prims Algorithm. Suppose this iteration augments as connectivity to x+1. Let r be the first common ancestor of vertices a and b in the tree formed by first j edges. Consider the cycle C formed by insertion of e in the tree T(with some edges repeated) on first j edges. Now e will be the most costly edge in this cycle as otherwise we can insert edge e and remove the other edge to get a tree of cost less than Prims Algorithms tree (MST).

As connecting a to b augments as connectivity, vertex r is connected to s by at least x+1 edge disjoint paths in the subgraph formed by first j edges. Hence consider the first vertex w in T on the path from r to a, which has connectivity less x+1. Now we can connect w to its parent p in T and augment ws connectivity. The cost of edge (w,p) is less than cost(e) which gives a contradiction as (i+1) iteration of the algorithm picks the cheapest edge such that connectivity of some vertex is augmented.

7.2 Failed Approach for proving worst case factor 2

We tried using the ideas of uniform connectivity and extend it for the any connectivity requirement r. The idea was to decompose bi- directed OPT into $maxr_v$ trees using Bang- Jenson decomposition. In this tree decomposition, all vertices will be present in at least $r(v,s)$ trees. Hence if we can prove that the Greedy Algorithm gives a solution to SNPSS of cost less

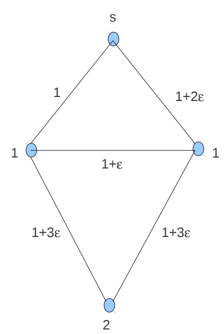


Figure 5: Counterexample

than the minimum way of decomposing a directed graph into trees such that all vertices are present in exactly $r(s, v)$ trees, then we prove a factor 4 approximation for the Greedy Algorithm. This is because the min exact decomposition will be within factor 2 from min decomposition (as Steiner vertices only give a factor 2). Now min- decomposition is of cost less than $2 * OPT$. Thus we will be able to prove a factor 4 for our Greedy Algorithm. Unfortunately there exists a counter example to the above claim as shown in Fig. 5

7.3 Related Problems and their results

1. **Connectivity augmentation:** Find a minimum cost of new edges to be added to a given graph so as to satisfy some prescribed connectivity requirements. As mentioned above, on an initial empty graph, this problem is similar to SNPSS on an un-weighted graph. When starting with a general un-weighted graph, the problem of min-cost augmentation such that the graph becomes k-edge connected was solved by T. Watanabe and A. Nakamura [3]. There is no restriction on the copies of an edge can edge that can be added.

2. **Smallest weight spanning subgraph:** Given a graph $G(V, E)$ with weights on the edges, find the smallest weight spanning subgraph $H(V, E_h)$ that is k edge connected, for any k. A 2- approximation factor algorithm to this problem was given by Khuller and Vishkin [16]. Gabow [17] gave a fast implementation of solving the following problem using weighted matroid intersection. Given a directed graph G with weights on the edges and a fixed root r, find the cheapest directed subgraph H_D that has k edge disjoint paths from root r to each vertex v. Hence in [16], they bi-direct the edges of the undirected graph and then use Gabows algorithm to solve this problem. This give them the required 2- approximation factor.

3. **Min Cost flow problems:** Given a digraph G, capacities $u : E(G) \rightarrow R_+$, edge costs $c : E(G) \rightarrow R$ and a number $b : V(G) \rightarrow R$ with $\sum_{v \in V(G)} b(V) = 0$, find a minimum cost b-flow i.e. a function $f : E(G) \rightarrow R_+$ with $f(e) \leq u(e)$ for all $e \in E(G)$ and $\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v)$ such that $\sum_{e \in E(G)} f(e) * c(e)$ is minimized. Unlike above problems, this problem can be solved in polynomial time and there are many efficient algorithms for solving this problem. Interested readers are referred to Chp-9 of [18].

8 References

1. Improved Steiner Tree Approximation in Graphs : G. Robins, A. Zelikovsky.

2. An Improved LP-based Approximation for Steiner Tree: Jaroslaw Byrka, Fabrizio Grandioni, Thomas Rothvob, Laura Sanita.

3. The k-Steiner Ratio in Graphs : Al Borchers, Ding-Zhu Du

4. On Steiner trees and minimum spanning trees in hypergraphs : Polzin and Daneshmand

5. New-geometry inspired relaxations and algorithms for the metric Steiner tree problem : D. Chakrabarty, N.R. Devanur, V. Vazirani

6. Hypergraphic LP Relaxations for Steiner Trees : Deeparnab Chakrabarty, Jochen Konemann, David Pritchard

7. Integrality Gap of the Hypergraphic Relaxation of Steiner Trees : a short proof of a 1.55 upper bound : Deeparnab Chakrabarty, Jochen Konemann, David Pritchard

8. A Partition-Based Relaxation For Steiner Trees : Jochen Konemann, David Pritchard, Kunlun Tan

9. Catalog of Steiner Tree formulations : Goemans, Young-soo Myung

10. Mohammad Taghi Hajiaghayi, Kamal Jain: The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. SODA 2006: 631-640

11. Mohammad Taghi Hajiaghayi, Kamal Jain: The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. SODA 2006: 631-640

12. Preserving and Increasing Local Edge-Connectivity in Mixed Graphs: Jorgen Bang-Jensen, Andras Frank, Bill Jackson

13. Survivable Communication Networks and the Terminal Capacity Matrix: Chou and Frank, IEEE Transactions on Circuit Theory, May 1970

14. Connectivity considerations in the design of survivable networks: Chou and Frank, IEEE Transactions on Circuit Theory, Nov 1970.

15. Edge connectivity augmentation problems: T. Watanabe and A. Nakamura, Computer and System Science, Vol 35, No 1, 1987

16. Biconnectivity Approximations and Graph Carvings: Samir Khuller and Uzi Vishkin, STOC 1992

17. A matroid approach to finding edge connectivity and packing arborescences: H. N. Gabow STOC 1991.

18. Combinatorial Optimization, Theory and Algorithms: Bernhard Korte and Jens Vygen, Springer 2003

19. L. Sunil Chandran. A high girth graph construction. SIAM J. Discret. Math.,16:366370, March 2003.