An Improved LP-based Approximation for Steiner Tree*

Jaroslaw Byrka†
Institute of Mathematics, EPFL
Lausanne, Switzerland
jaroslaw.byrka@epfl.ch

Fabrizio Grandoni‡
University of Tor Vergata
Rome, Italy
grandoni@disp.uniroma2.it

Thomas Rothvoß
Institute of Mathematics, EPFL
Lausanne, Switzerland
thomas.rothvoss@epfl.ch

Laura Sanità§
Institute of Mathematics, EPFL
Lausanne, Switzerland
laura.sanita@epfl.ch

ABSTRACT

The Steiner tree problem is one of the most fundamental NP-hard problems: given a weighted undirected graph and a subset of terminal nodes, find a minimum-cost tree spanning the terminals. In a sequence of papers, the approximation ratio for this problem was improved from 2 to the current best 1.55 [Robins,Zelikovsky-SIDMA’05]. All these algorithms are purely combinatorial. A long-standing open problem is whether there is an LP-relaxation for Steiner tree with integrality gap smaller than 2 [Vazirani,Rajagopalan-SODA’99].

In this paper we improve the approximation factor for Steiner tree, developing an LP-based approximation algorithm. Our algorithm is based on a, seemingly novel, iterative randomized rounding technique. We consider a directed-component cut relaxation for the k-restricted Steiner tree problem. We sample one of these components with probability proportional to the value of the associated variable in the remaining graph. In a sequence of improved approximation ratios for this problem was improved from 2 to the current best 1.55 [Robins,Zelikovsky-SIDMA’05]. All these algorithms are purely combinatorial. A long-standing open problem is whether there is an LP-relaxation for Steiner tree with integrality gap smaller than 2 [Vazirani,Rajagopalan-SODA’99].

In this paper we improve the approximation factor for Steiner tree, developing an LP-based approximation algorithm. Our algorithm is based on a, seemingly novel, iterative randomized rounding technique. We consider a directed-component cut relaxation for the k-restricted Steiner tree problem. We sample one of these components with probability proportional to the value of the associated variable in the optimal fractional solution and contract it. We iterate this process for a proper number of times and finally output the sampled components together with a minimum-cost terminal spanning tree in the remaining graph. Our algorithm delivers a solution of cost at most \( \ln(4) \) times the cost of an optimal k-restricted Steiner tree. This directly implies a \( \ln(4) + \epsilon < 1.39 \) approximation for Steiner tree.

As a byproduct of our analysis, we show that the integrality gap of our LP is at most 1.55, hence answering to the mentioned open question. This might have consequences for a number of related problems.

Categories and Subject Descriptors

F.2.2 [Computations on discrete structures]: Non-numerical Algorithms and Problems

General Terms

Algorithms, Theory

Keywords

Network design, approximation algorithms, linear programming relaxations, randomized algorithms

1. INTRODUCTION

Given an undirected \( n \)-node graph \( G = (V,E) \), with edge costs (or weights) \( c : E \rightarrow \mathbb{Q}^+ \), and a subset of nodes \( R \subseteq V \) (terminals), the Steiner tree problem asks for a tree \( S \) spanning the terminals, of minimum cost \( c(S) := \sum_{e \in S} c(e) \). Note that \( S \) might contain some other nodes, besides the terminals (Steiner nodes). Steiner tree is one of the classical and, probably, most fundamental problems in Computer Science and Operations Research, with great theoretical and practical relevance. This problem emerges in a number of contexts, such as the design of VLSI, optical and wireless communication systems, as well as transportation and distribution networks (see, e.g., [27]).

The Steiner tree problem appears already in the list of NP-hard problems in the book by Garey and Johnson [19]. In fact, it is NP-hard to find solutions of cost less than \( 2^\text{poly}(n) \) times the optimal cost [5, 10]. Hence, the best one can hope for is an approximation algorithm with a small but constant approximation guarantee. Without loss of generality, we can replace the weighted graph given as input by its metric closure\(^1\). It is well-known that a minimum-cost terminal spanning tree \( T \) is a 2-approximation for the Steiner tree problem [20, 41]. A terminal spanning tree is a Steiner tree without Steiner nodes: such a tree always exists in the metric closure of the graph. A sequence of improved approximation algorithms appeared in the literature [29, 35, 55].

\(^1\)The **metric closure** of a weighted graph is a complete weighted graph on the same node set, with weights given by shortest path distances with respect to original weights.
ordering its linear relaxation, 2-approximation algorithms can be obtained either using primal-dual schemes \[22\] or iter-cut separating the set of terminals. Each constraint forces an undirected cut formulation LP-relaxation (see, e.g., \[21\] for a list of LP-relaxations for methods. One standard approach is to exploit a proper refinement after \[38\]. This motivated our search for alternative approaches. The idea is to start with a minimum-cost terminals \(R\) partitioned by its components. A Steiner tree is a maximal subtree whose terminals coincide with its leaves. Note that the edge set of the Steiner tree is the set of edges leaving \(U\) and \(v(\cdot \cup \{u\})\) of at most \(k\) terminals \((k\text{-component})\). The following result by Borchers and Du \[6\] shows that, in order to obtain a good approximation factor, it is sufficient to restrict our attention to \(k\)-restricted Steiner trees. We let \(Opt\) and \(Opt_k\) denote an optimal Steiner tree and an optimal \(k\)-restricted Steiner tree, respectively. Moreover, \(opt := c(\text{Opt})\) and \(opt_k := c(\text{Opt}_k)\).

**Theorem 1.** \[6\] Let \(\rho_k\) be the \(k\)-Steiner ratio, i.e., the supremum of the ratio \(\frac{\text{opt}}{\text{opt}_k}\). Then \(\rho_k = \frac{(k+1)2^k + s}{2^k + s} \leq 1 + \frac{1}{\log_2 k}\), where \(r\) and \(s\) are non-negative integers such that \(k = 2^r + s < 2^s\).

We remark that, given an optimal \(k\)-restricted Steiner tree \(S^k\), its components are optimal Steiner trees connecting the corresponding terminals. For any fixed \(k\), a list \(\{Z_1, \ldots, Z_r\}\), \(q = O(n^k)\), of all potential \(k\)-components can be computed in polynomial time by considering all subsets \(R'\) of at most \(k\) terminals, and computing an optimal Steiner tree⁵ on terminals \(R'\). Unfortunately, selecting the cheapest subset of \(\{Z_1, \ldots, Z_r\}\) spanning the terminals is an \(\text{NP}\)-hard problem already for \(k \geq 4\) \[18\]. For this reason, \[38\] and previous papers rather select a subset of the \(Z_i\)'s with a local-search approach. The idea is to start with a minimum-cost terminal spanning tree \(T^0\) (which is formed by 2-components), and iteratively improve it. At each step, one considers each \(Z_i\), and checks how much adding \(Z_i\) to the current solution (and removing redundant edges) improves the solution itself. The algorithm each time selects the \(Z_i\) leading to the largest improvement, and halts when no further improvement is possible. Different algorithms (essentially) differ in the way the improvement is evaluated.

Despite the efforts of many researchers in the last 10 years, the approach above did not provide any further improvement after \[38\]. This motivated our search for alternative methods. One standard approach is to exploit a proper LP-relaxation (see, e.g., \[21\] for a list of LP-relaxations for Steiner tree). A natural formulation for the problem is the **undirected cut formulation** (see \[22, 41\]), where we have a variable for each edge of the graph and a constraint for each cut separating the set of terminals. Each constraint forces to pick at least one edge crossing the corresponding cut. Considering its linear relaxation, 2-approximation algorithms can be obtained either using primal-dual schemes \[22\] or iterative rounding \[28\]. However, this relaxation has an integrality gap of 2 already in the spanning tree case, i.e., when \(R = V\) (see example 22.10 in \[41\]).

Another well-studied but more promising LP is the so-called **bidirected cut relaxation** \[8, 12, 36\]. Let us fix an arbitrary terminal \(r\) (root). Replace each edge \(\{u, v\}\) by two directed edges \((u, v)\) and \((v, u)\) of cost \(c(\{u, v\})\). For a given cut \(U \subseteq V\), define \(\delta^+(U) = \{\{u, v\} \in E \mid u \in U, v \notin U\}\) as the set of edges leaving \(U\). The mentioned relaxation is

\[
\min \sum_{e \in E} c(e)z_e \quad \text{(BCR)} \\
\sum_{e \in \delta^+(U)} z_e \geq 1 \quad \forall U \subseteq V \setminus \{r\} : U \cap R \neq \emptyset \\
z_e \geq 0 \quad \forall e \in E.
\]

We can consider the value \(z_e\) as the capacity which we are going to install on the directed edge \(e\). The LP can then be interpreted as computing the minimum-cost capacities that support a flow of 1 from each terminal to the root. In a seminal work, Edmonds \[12\] showed that BCR is integral in the spanning tree case.

**Theorem 2.** \[12\] For \(R = V\), the polyhedron of BCR is integral.

The best-known lower bound on the integrality gap of BCR is \(8/7\) \[31, 41\]. The best-known upper bound is 2, though BCR is believed to have a smaller integrality gap than the undirected cut relaxation \[36\]. The authors in \[8\] report that the structure of the dual to BCR is highly asymmetric, which complicates a primal-dual approach. Moreover, iterative rounding based on picking a single edge cannot yield good approximations, as was pointed out in \[36\].

Finding a better-than-2 LP-relaxation for the \((k\text{-restricted})\) Steiner tree problem is a long-standing open problem \[8, 36\]. We remark that good LP-bounds, besides potentially leading to better approximation algorithms for Steiner tree, might have a much wider impact. This is because Steiner tree appears as a building block in several other problems, and the best approximation algorithms for some of those problems are LP-based. Strong LPs are also important in the design of (practically) efficient and accurate heuristics.

### 1.1 Our Results and Techniques

We next state the main result of this paper (see Section 4.1 for a derandomization of this result).

**Theorem 3.** For any constant \(\varepsilon > 0\), there is a polynomial-time randomized approximation algorithm for the Steiner tree problem with expected approximation ratio \(\ln(4) + \varepsilon\).

Our algorithm is based on the following **directed-component cut relaxation** for the \(k\)-restricted Steiner tree problem (see also \[34\]). Consider a \(k\)-restricted Steiner tree \(S\) with the edges directed towards the chosen root terminal \(r\) (see Figure 1). Consider the (undirected) \(k\)-components \(Z_i\) introduced before. Make a copy \(C\) of each \(Z_i\) for each choice

![Figure 1: (a) A 4-restricted Steiner tree \(S\), where rectangles denote terminals and circles represent Steiner nodes. (b) Edges of \(S\) are directed towards a root \(r\). The resulting directed components are depicted with different colors.](image-url)
of one (sink) terminal $u_1$ in it, and direct all the edges of $C$ towards $u_1$. Let $C_1, \ldots, C_h$ be the resulting directed $k$-components. Observe that $h = O(kn^k)$ is polynomially bounded for any fixed $k$. We denote by $c(C_j)$ the cost of $C_j$. Recall that $c(C_j)$ is the cost of an optimal Steiner tree over $R \cap V(C_j)$, and that this tree can be computed in polynomial time (for constant $k$). We also let $\text{sink}(C_j)$ be the sink terminal of $C_j$, and $\text{sources}(C_j) := V(C_j) \cap R \setminus \{\text{sink}(C_j)\}$ be the other terminals (sources of $C_j$). We say that a component $C_j$ crosses $U \subseteq R$ if $C_j$ has at least one source in $U$ and the sink outside. By $\delta_k(U)$ we denote the set of $k$-components crossing $U$. Our LP-relaxation is then:

$$\min \sum_j c(C_j)x_j \quad \text{(k-DCR)}$$

$$\sum_{j \in \delta_k(U)} x_j \geq 1 \quad \forall U \subseteq R \setminus \{r\}, \ R \neq \emptyset$$

$$x_j \geq 0 \quad \forall j = 1, \ldots, h.$$ 

The LP above can be solved in polynomial time (see Section 2).

We remark that $k$-DCR is a relaxation for the $k$-restricted Steiner tree problem. In fact, consider the optimal integral solution $\text{Opt}_k$. As already observed, we can direct all its edges towards an arbitrary root terminal $r$. At this point, $\text{Opt}_k$ consists of a set of directed $k$-components $C_j$, where each $C_j$ is an optimal Steiner tree over $R \cap V(C_j)$. Setting $x_j = 1$ for those components, and the remaining variables to zero, provides a feasible solution to $k$-DCR of cost exactly $\text{opt}_k$.

We combine our LP with a (to the best of our knowledge) novel iterative randomized rounding technique. We solve the LP, sample one component $C_j$ with probability proportional to its value $x_j$ in the optimal fractional solution $x$, contract $C_j$ into its sink node $\text{sink}(C_j)$, and reoptimize the LP. We iterate this process for a suitable number of times. With a simple analysis we can show that a minimum-cost terminal spanning tree on the remaining terminals plus the sampled components cost (in expectation) at most $3/2$ times the cost of the optimal $k$-restricted Steiner tree (see Section 3). With a refined analysis, we improve this bound to $1 + \ln(4) + \frac{1}{\ln(3)} < 1.39$ approximation for Steiner tree immediately follows from Theorem 1 by choosing $k$ large enough. This bound can be further improved for special graph classes, as for example for quasi-bipartite graphs, where non-terminal nodes are not adjacent (details will be given in the journal version of the paper).

We remark that our algorithm combines features of randomized rounding (where typically variables are rounded randomly, but simultaneously) and iterative rounding (where variables are rounded iteratively, but deterministically). We believe that our iterative randomized rounding technique will also find other applications, and is henceforth of independent interest.

The key insight in our analysis is to quantify the expected reduction of the cost of the optimal terminal spanning tree and optimal Steiner tree in each iteration. To show this, we exploit a Bridge Lemma, relating the cost of terminal spanning trees with the cost of fractional solutions to $k$-DCR. The proof of the lemma is based on Edmonds’ Theorem 2.

\[ \text{opt}_k \leq 3 \text{opt}_k \]

[12]. In our opinion, our analysis is simpler (or at least more intuitive) than the one in [38].

As an easy consequence of our analysis, we obtain that the integrality gap of $k$-DCR is at most $1 + \ln(2) < 1.694$, hence answering to the mentioned open problem in [8, 36] (for the $k$-restricted case). A more technical analysis, based on an adaptation of the analysis of Robins and Zelikovsky which exploits our Bridge Lemma, leads to the following improved result (see Section 5).

**Theorem 4.** For any constant $k$, there is a polynomial-time algorithm which computes a solution for $k$-restricted Steiner tree of cost at most $1 + \ln(\frac{n}{4}) < 1.55$ times the cost of the optimal fractional solution to $k$-DCR.

As mentioned before, integrality gap results of this type often provide new insights to variants and generalizations of the original problem. We expect that this will be the case with the above theorem as well, since Steiner tree appears as a building block in many other problems.

### 1.2 Related Work

One reason for the importance of Steiner tree is that it appears either as a subproblem or as a special case of many other problems in network design. A (certainly incomplete) list contains Steiner Forest [1, 22], Prize-Collecting Steiner tree [3, 22], Virtual Private Network [13, 14, 25], Single-Sink Rent-or-Buy [15, 16, 26], Connected Facility Location [15, 16, 39] and Single-Sink Buy-At-Bulk [23, 26, 40].

Both the previously cited primal-dual and iterative rounding approximation techniques apply to a more general class of problems. In particular, the iterative rounding technique introduced by Jain [28] provides a 2-approximation for the generalized Steiner network problem, and the primal-dual framework developed by Goemans and Williamson [22] gives the same approximation factor for a large class of constrained forest problems.

Regarding the integrality gap of LP relaxations for the Steiner tree problem, upper bounds better than 2 are known only for special graph classes. For example, BCR has an integrality gap smaller than 2 on quasi-bipartite graphs, where non-terminal nodes induce an independent set. For such graphs Rajagopalan and Vazirani [36] (see also Rizzi [37]) gave an upper bound of 3/2 on the gap. This was recently improved to 4/3 by Chakrabarty, Devanur and Vazirani [8]. Still, for this class of graphs the lower bound of 8/7 holds [31, 41]. Könenmann, Pritchard and Tan [31] showed that for a different LP formulation, which is stronger than BCR, the integrality gap is upper-bounded by $\frac{2b+1}{b}$, where $b$ is the maximum number of Steiner nodes in full components.

Finally, we remark that under additional constraints, Steiner tree admits better approximations. In particular, a PTAS can be obtained by the technique of Arora [4] if the nodes are points in a fixed-dimension Euclidean space, and using the algorithm of Borradaile, Kenyon-Mathieu and Klein [7] for planar graphs.

### 2. A DIRECTED-COMPONENT CUT RELAXATION

We now prove some crucial properties of our $k$-DCR. The optimal fractional solution to $k$-DCR is denoted by $\text{Opt}_k^*$, and $\text{opt}_k^*$ is its cost. For a given (directed or undirected) component $C'$, $R(C') := R \cap V(C')$ is the set of its termi-
nals. Recall that \( k \) is a constant, hence \( k \)-DCR has a polynomial number of variables. Despite the fact that \( k \)-DCR has an exponential number of constraints, it can be solved to optimality using the Ellipsoid method [24, 30], since we can solve the separation problem in polynomial time.

**Lemma 5.** \( k \)-DCR can be solved in polynomial time, for any constant \( k \).

**Proof.** We show how to solve the separation problem in polynomial time. Create a new directed graph \( G' \), on node set \( V' = R \cup \{v_j \mid j = 1, \ldots, h\} \). For every component \( C_j \), insert edges \( (u, v_j) \) for any \( u \in \text{sources}(C_j) \), and one edge \( e_j = (v_j, \text{sink}(C_j)) \). Set the capacity of each edge \( e_j \) to \( w(e_j) := x_j \), and let \( w(e) := \infty \) for the remaining edges. It is not hard to see that, for a terminal \( R \subseteq S \) and consider the minimum-cost tree \( T \) of \( R \). The claim follows.

We call the latter edges the bridges of \( S \) w.r.t. \( R' \), and denote them by \( Br_S(R') \)\(^4\). Intuitively, if we imagine to add zero cost dummy edges between the terminals \( R' \), \( Br_S(R') \) is a maximum-cost subset of edges that we could remove from \( S \) and still have a connected spanning subgraph (see Figure 2). In other terms, \( Br_S(R') \) is equal to

\[
\max \left\{ c(B) \mid B \subseteq S, S \setminus B \cup \{R'\} \text{ connects } V(S) \right\}.
\]

Let us abbreviate \( br_S(R') := c(Br_S(R')) \). For a (directed or undirected) component \( C' \), we use \( Br_{S}(C') \) and \( br_S(C') \) as shortcuts for \( Br_S(R(C')) \) and \( Br_S(R(C')) \), respectively.

A key ingredient in the proof of our Bridge Lemma is the construction of a proper weighted terminal spanning tree as described in the following lemma. We define a bridge weight function \( w : R \times R \rightarrow \mathbb{Q}_+ \) as follows: For any terminal pair \( u, v \in R \), the quantity \( w(u, v) \) is the maximum cost of any edge in the unique \( u-v \) path in \( S \).

**Lemma 7.** Let \( R' \subseteq R \) and \( Br_S(R') = \{b_1, \ldots, b_{k'}-1\} \) with \( k' := |R'| \). Then one can construct a spanning tree \( Y' = \{e_1, \ldots, e_{k'}-1\} \) on \( R' \) such that, for \( i = 1, \ldots, k'-1 \),

\[
\text{(a)} \quad w(e_i) = c(b_i) \quad \text{(hence } w(Y') = br_S(R')).
\]

\( b_i \) is the only bridge edge on the cycle in \( S \cup \{e_i\} \).

**Proof.** Observe that \( S \setminus Br_S(R') \) is a forest of trees \( F_1, \ldots, F_{k'} \), where each \( F_i \) contains exactly one terminal \( r_i \in R' \). Each bridge \( b_i \) connects exactly two trees \( F_{l_i} \) and \( F_{l_{i+1}} \). For each \( b_i \), we add edge \( e_i = \{r_{l_i}, r_{l_{i+1}}\} \) to \( Y' \). Clearly \( Y' \) is a spanning tree on \( R' \). The path \( P_i \) between \( r_{l_i} \) and \( r_{l_{i+1}} \) contains \( b_i \) and no other bridge. Hence \( b_i \) is a maximum-cost edge on \( P_i \) and \( w(e_i) = c(b_i) \) (see Figure 2).

The following lemma is the heart of our analysis.

**Lemma 8.** [Bridge Lemma] Let \( T \) be a terminal spanning tree and \( x \) be a \( k \)-DCR solution. Then

\[
c(T) \leq \sum_j x_j \cdot br_T(C_j).
\]

\(^4\)As usual, we break ties according to edge indexes.
Figure 3: A 3/2-approximation algorithm for \(k\)-restricted Steiner tree.

\textbf{Proof.} For every component \(C_j\) we construct a spanning tree \(Y_j\) on \(R(C_j)\) with weight \(w(Y_j) = br_T(C_j)\) according to Lemma 7. Then we direct the edges of \(Y_j\) towards \(sink(C_j)\). We define a directed capacity reservation \(y : R × R → \mathbb{Q}_+\) as follows: For every \(j\), install capacity \(x_j\) in a cumulative manner on \(Y_j\). In other terms, \(y(u, v) := \sum_{j \in R(u, v)} x_j\). The directed tree \(Y_j\) supports at least the same flow as component \(C_j\) with respect to \(R(C_j)\). It then follows that \(y\) supports one unit of flow from each terminal to the root. In other terms, \(y\) is a feasible fractional solution to BCR. By Theorem 2, BCR is integral when no Steiner node is used. As a consequence there is an (integral) terminal spanning tree \(F\) that is not more costly than the fractional solution \(y\), i.e. \(w(F) ≤ \sum e \subseteq R \ w(e) y(e)\).

Recall that \(w(u, v)\) for \(u, v \in R\), is the maximum cost of any edge of the unique cycle in \(T \cup \{u, v\}\). It follows from the classical cycle rule for minimum-cost spanning tree computation that \(w(F) ≥ c(T)\) (see, e.g., Theorem 6.2 in [32]). Altogether

\[
\sum_j x_j br_T(C_j) = \sum_j x_j w(Y_j) = \sum_{e \subseteq R} w(e) y(e) ≥ w(F) ≥ c(T). \quad \square
\]

3. ITERATIVE RANDOMIZED ROUNDELING

In this section we present our approximation algorithm for \(k\)-restricted Steiner tree. To highlight the novel ideas of the approximation technique more than the approximation factor itself, we present a simplified analysis providing a weaker 3/2-approximation factor (which is already an improvement on the previous best 1.55 approximation). The more complex analysis leading to ln(4) is postponed to Section 4.

Our 3/2-approximation algorithm for \(k\)-restricted Steiner tree is described in Figure 3. Let \(x^*\) be the optimal fractional solution to \(k\)-DRC at a generic iteration \(t\). By sampling a component \(C_t\), we mean selecting one of the components \(C_j\) with probability \(x_j^* / \sum x_i^*\). Contracting a component \(C_t\) means collapsing all its terminals into its sink \(sink(C_t)\), which inherits all the edges incident to \(C_t\) (in case of parallel edges, we only keep the cheapest one).

Recall that \(h = O(kn^k)\) is the number of potential \(k\)-components. Observe that the quantity \(\Sigma' := \sum x_i^*\) might vary over the iterations \(t\). In order to simplify the analysis, we apply the above algorithm to a slightly different LP where we add a dummy component \(C_{h+1}\) formed by the root only (hence of cost zero), and add the constraint \(x_{h+1} = \Sigma - \sum_{i=1}^h x_i\). Here \(\Sigma = O(h)\) is an upper bound on the possible sum of the \(x_i\)'s in the original LP. The number \(\mu\) of iterations is fixed to \(\delta \Sigma\), where \(\delta\) is a proper constant to be chosen later. [W.l.o.g., \(\delta \Sigma\) is integral]. It is easy to see that the running time of the algorithm is polynomial.

We first outline the analysis of our algorithm. Let \(Opt^t_k\) be the optimal \(k\)-restricted Steiner tree at the beginning of iteration \(t\), and let \(opt^t_k\) be its cost. By \(opt^t_{k, C_t}\) we denote the cost of the optimal fractional solution at the beginning of iteration \(t\). Lemma 9 bounds the expected cost of the final terminal spanning tree \(T^*\). The basic idea is showing that the current terminal spanning tree is getting cheaper by a factor \((1 - 1/\mu)\) at each iteration (in expectation): This is an easy consequence of the Bridge Lemma. Lemma 13 bounds the expected cost of each sampled component \(C_t\). For this component we pay in expectation a \(1/\mu\) fraction of \(opt^t_{k, C}\), which is in turn upper bounded by \(opt_{k, C}\). Hence it is sufficient to bound the cost of \(opt_{k, C}\) (Corollary 12). In order to do that, we show that the cost of the optimal Steiner tree decreases by a factor \((1 - \delta/\mu)\) at each iteration (Lemma 12). Also in this case the proof relies crucially on the Bridge Lemma.

The next lemma bounds the cost of the final terminal spanning tree.

\textbf{Lemma 9.} One has \(E[c(T^*)] ≤ (1 - 1/\mu) \cdot 2opt^t_k\).

\textbf{Proof.} Let \(T^t (T^*, \text{resp.})\) be the minimum-cost terminal spanning tree at the end of iteration \(t\) (for the original instance, \(\text{resp.}\). Consider an arbitrary iteration \(t = 1, \ldots, \mu\). The reduction in the cost of \(T^t\) w.r.t. \(T^{t-1}\) is at least \(br_{T, t-1}(C^t)\). Therefore:

\[
E[c(T^t)] = c(T^{t-1}) - E[br_{T, t-1}(C^t)] \\
= c(T^{t-1}) - \frac{1}{\mu} \sum_j x_j^* \cdot br_{T, t-1}(C_j) \\
\leq Bridge\ Lem\ 8 \left(1 - \frac{1}{\mu}\right) \cdot c(T^{t-1}).
\]

It follows that

\[
E[c(T^t)] ≤ \left(1 - \frac{1}{\mu}\right) \cdot c(T^0) \leq \left(1 - \frac{1}{\mu}\right)^\mu \cdot 2opt_k. \quad \square
\]

It remains to bound the cost of the sampled components. The proof of the following technical lemma is based on standard techniques (see, e.g., [29]).

\textbf{Lemma 10.} For any Steiner tree \(S\), \(br_S(R) ≥ \frac{1}{2} c(S)\).

\textbf{Proof.} Turn \(S\) into a binary tree with leaves \(R\) by adding dummy Steiner nodes and zero cost edges. For each Steiner node of \(S\), mark the most expensive edge out of the edges going to its 2 children. Let \(B ⊆ S\) be the set of marked edges. Observe that \(c(B) ≥ \frac{1}{2} c(S)\). Furthermore, after contracting \(R\), one can remove \(B\) while keeping \(S\) connected. From the definition of bridges it follows that \(br_S(R) ≥ c(B) ≥ \frac{1}{2} c(S)\). \(\square\)

The next lemma and corollary bound the expected decrease of the cost of the optimal Steiner tree after each contraction.

\textbf{Lemma 11.} Let \(S\) be any Steiner tree and \(x\) be a feasible solution to \(k\)-DRC. Sample a component \(C\) randomly w.r.t. \(x\). Then there is a subgraph \(S' \subseteq S\) such that \(S' \cup C\) spans \(R\) and

\[
E[c(S')] ≤ \left(1 - \frac{1}{2\Sigma}\right) \cdot c(S).
\]
PROOF. Let $Br_S(R) = \{b_1, \ldots, b_q\}$. We apply Lemma 7 to $S$ to obtain a terminal spanning tree $Y$ consisting of edges $e_1, \ldots, e_q$ such that $w(e_i) = c(b_i)$ (hence $w(Y) = br_S(R)$) and $b_i$ is the unique bridge on $S \cup \{e_i\}$. Choose $S' := S \{b_i | e_i \in Br_Y(C)\}$. (Here $Br_Y(C)$ is computed w.r.t. weights $w(e_i)$). Observe that $S' \cup C$ spans $R$. In fact, consider any edge $e_i = \{u_i, v_i\} \in Y \setminus Br_Y(C)$. The $u_i$-$v_i$ path in $S$ contains only one potential bridge edge, namely $b_i$. Hence, if two terminals are connected by $Y \setminus Br_Y(C)$, then they are so in $S'$. The claim follows since $(Y \setminus Br_Y(C)) \cup C$ is connected due to the definition of bridges. We conclude

$$E[c(S')] = E[c(S) - E[br_Y(C)]]$$

$$= (c(S) - \frac{1}{\Sigma} \sum_j x_j br_Y(C_j)$$

Bridge Lem 8

$$\leq c(S) - \frac{1}{\Sigma} w(Y)$$

$$= c(S) - \frac{1}{\Sigma} br_S(R)$$

Lem 10

$$\leq \left(1 - \frac{1}{2\Sigma}\right)c(S). \square$$

COROLLARY 12. For every $t = 1, \ldots, \mu$,

$$E[opt_k^t] \leq \left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot opt_k.$$  

Proof. Observe that, in Lemma 11, if the initial Steiner tree $S$ is $k$-restricted, then $S' \cup C$ is $k$-restricted as well. Hence, this lemma implies that $E[\text{opt}^{t+1}_k] \leq (1 - \frac{1}{2\Sigma}) \cdot \text{opt}_k^t$ for any iteration $t = 1, \ldots, \mu - 1$. The claim follows. \square

Corollary 12 immediately provides an upper bound on $\text{opt}_k^{t+1}$. An upper bound on the expected cost of the sampled components easily follows.

LEMMA 13. For every $t = 1, \ldots, \mu$,

$$E[c(C^t)] \leq \frac{1}{\Sigma} \left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot \text{opt}_k.$$  

Proof. One has

$$E[c(C^t)] = \frac{1}{\Sigma} E[\sum_j x_j^t \cdot c(C_j)]$$

$$= \frac{1}{\Sigma} E[\text{opt}_k^{t+1}]$$

$$\leq \frac{1}{\Sigma} E[\text{opt}_k^t]$$

Cor 12

$$\leq \left[1 - \left(1 - \frac{1}{2\Sigma}\right)^{t-1}\right] \cdot \text{opt}_k. \square$$

We now have all the ingredients to show a 3/2-approximation factor for the problem.

THEOREM 14. For any $k = O(1)$, there is a polynomial-time randomized approximation algorithm for $k$-restricted Steiner tree with expected approximation ratio 3/2.

Proof. Consider the algorithm of Figure 3 with $\mu = \delta \Sigma$ and $\delta = \ln(4)$. The cost of the computed solution is $c(T^\mu) + \sum_t c(C^t)$. (1) For $t = 1, 2, \ldots$

(1a) Compute an optimal fractional solution $x^t$ to $\text{k-DCR}$ (w.r.t. the current instance).

(1b) Sample one component $C^t$, where $C^t = C_j$ with probability $x_j^t / \sum x_j^t$. Contract $C^t$.

(1c) If the instance consists only of the root, return $\bigcup_{j=1}^q C_j$.

Figure 4: A $\ln(4) + \varepsilon$-approximation algorithm for $k$-restricted Steiner tree.

$\sum_{t=1}^\mu c(C^t)$. The expected approximation ratio satisfies

$$E \left[ \frac{c(T^\mu) + \sum_{t=1}^\mu c(C^t)}{opt_k} \right]$$

$$\leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\mu} + \frac{1}{\Sigma} \sum_{t=1}^\mu \left(1 - \frac{1}{2\Sigma}\right)^{t-1}$$

$$= 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\delta \Sigma} + 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\delta \Sigma}$$

$$\leq 2e^{-\delta} + 2 \cdot e^{-\delta/2} = \frac{3}{2}. \square$$

In the last inequality we used the fact that $(1 - \frac{1}{\Sigma})^\mu - (1 - \frac{1}{2\Sigma})^\mu$ is an increasing function of $\mu > 1$, and that $\lim_{\mu \to \infty} (1 - \frac{1}{\Sigma})^\mu = \frac{1}{\Sigma}$. 

Theorems 1 and 14 immediately imply an expected $(3/2 + \varepsilon)$-approximation algorithm for the Steiner tree problem.

4. A REFINED APPROXIMATION

In this section, we present a $(\ln(4) + \varepsilon)$-approximation for Steiner tree. The algorithm is the same as in the previous section, with the difference that now we let it run until all the terminals collapse into the root. The set of sampled components provides the desired solution (see Figure 4).

We first give a high-level description of our analysis. Let $S^* := \text{Opt}_k$ be the optimal $k$-restricted Steiner tree for the original instance (in particular, $c(S^*) = \text{opt}_k$). Each time we sample a component $C^t$, we will delete a proper subset of edges from $S^*$. Consider the sequence $S^* = S^1 \supseteq S^2 \supseteq \ldots$ of subgraphs of $S^*$ which are obtained this way. We will guarantee that at any iteration $t$, the edge set $S^t$ plus the previously sampled components yields a subgraph that connects all terminals. Furthermore, we will prove that a fixed edge $e \in S^*$ is deleted after an expected number of at most $\ln(4) \cdot \Sigma$ iterations. This immediately implies the approximation factor of $\ln(4)$.

In order to track which edges can be safely deleted from $S^*$, we will construct an artificial terminal spanning tree $Y$ (the witness tree) and assign a random subset $W(e)$ of edges of $Y$ to each edge $e \in S^*$. The choice of $W(e)$ guarantees (deterministically) that, if a pair of terminals is connected by $Y^{e} \subseteq Y$, then they are as well connected by $\{e \in S^* | W(e) \cap Y^e \neq \emptyset\}$.

At each iteration, when component $C^t$ is sampled, we mark a proper random subset $Br_Y(C^t)$ of edges of $Y$. This set guarantees that $(Y \setminus Br_Y(C^t)) \cup C^t$ is connected (deterministically). The intuitive reason for using $Br_Y(C^t)$ rather than $Br_Y(C^t)$ is that we want to sample each edge of $Y$ more uniformly. When all the edges in $W(e)$ are marked,
In particular, we delete $e$ from $S^\ast$. Summarizing, we consider the following random process:

For $t = 1, 2, \ldots$, sample one component $C^t$ from $x^t$ and mark the edges in $B_{Y}(C^t)$. Delete an edge $e$ from $S^\ast$ as soon as all edges in $W(e)$ are marked.

The subgraph $S^t$ is given by the edges of $S^\ast$ which are not yet deleted at the beginning of iteration $t$.

We next give the details of our analysis. A combination of Farkas’ Lemma together with our Bridge Lemma provides the existence of random sets $B_{Y}(C_j)$ such that every edge in $Y$ is marked with probability at least $\frac{1}{3}$ per iteration.

**Lemma 15.** Let $Y$ be any terminal spanning tree and $x$ be any $k$-DCR solution with $\Sigma = \sum_j x_j$. Then there exist random sets $B_{Y}(C_j)$ such that if we sample a component $C$ randomly w.r.t. $x$ one has

1. $\Pr[e \in B_{Y}(C)] \geq \frac{1}{3}$ for all $e \in Y$;
2. $Y \setminus B_{Y}(C_j) \cup C_j$ connects $V(Y)$ for all $C_j$.

**Proof.** For a component $C_j$, let the set of candidate bridges $B_{Y}(C_j)$ be

$$\{ B \subseteq Y \mid |B| = |R(C_j)| - 1, (Y \setminus B) \cup C_j \text{ connects } V(Y) \}.$$ 

By definition any $B \in B_{Y}(C_j)$ satisfies Property (2). For a proper probability distribution $w$, we let $\Pr[B_{Y}(C_j)] = |B| := w_{jB}$. In particular, $\sum_{B \in B_{Y}(C_j)} w_{jB} = 1$. We will show that there is a $w$ with

$$\sum_{(B,j): B \in B_{Y}(C_j), e \in B} x_j w_{jB} \geq 1$$

for all $e \in Y$. This implies Property (1) since

$$\Pr[e \in B_{Y}(C)] = \sum_{(B,j): B \in B_{Y}(C_j), e \in B} x_j w_{jB} \geq \frac{1}{\Sigma} .$$

Suppose by contradiction that such a distribution $w$ does not exist. Then the following system of linear inequalities has no solution\(^5\)

$$\sum_{B \in B_{Y}(C_j)} w_{jB} \leq 1 \quad \forall j$$

$$\sum_{(B,j): B \in B_{Y}(C_j), e \in B} x_j w_{jB} \geq 1 \quad \forall e \in Y$$

$$w \geq 0 .$$

Farkas’ Lemma\(^6\) yields that there is a vector $(y, c) \geq 0$ with

(a) $y_j \geq \sum_{B \in B_{Y}(C_j)} c_B x_j$ \quad $\forall (B,j) : B \in B_{Y}(C_j)$;

(b) $\sum_j y_j \leq \sum_{B \in B_{Y}(C_j)} c_B = c(Y).$

In particular,

$$y_j^{(a)} \geq x_j \cdot \max \{ c_B x_j \mid B \in B_{Y}(C_j) \} = x_j \cdot b_{Y}(C_j).$$

We can replace the “$=$” constraint with “$\leq$” without affecting feasibility since all coefficients of $w_{jB}$ are non-negative.\(^6\)

We let $Pr_0 = 1$. \(\diamond\)

Next, we use the witness tree $Y$ and the sets $W(e)$ for each $e \in S^\ast$. Without loss of generality any Steiner node in $S^\ast$ has degree 3 or more. By adding dummy Steiner nodes and zero cost edges, we can assume that $S^\ast$ is (not necessarily complete) binary tree, rooted at some Steiner node, of height at most $|R| - 1$. For each Steiner node, choose uniformly at random one of the edges to its children. Let $B$ denote the chosen edges. Clearly $\Pr[e \in B] = \frac{1}{2}$ for any $e \in S^\ast$. Let $P_{uv} \subseteq S^\ast$ be the unique u-v path in $S^\ast$. The witness tree is

$$Y := \{ u, v \} \in \{ k \} \mid |P_{uv} \cap B| = 1 .$$

Similarly to arguments in Lemma 7, $Y$ is a spanning tree on $R$. Furthermore, for each edge $e \in S^\ast$, we define

$$W(e) := \{ u, v \} \in Y \mid e \in P_{uv} .$$

See Figure 5(a) for an illustration. Note that $1 \leq |W(e)| \leq |Y| - 1$. Also observe that $|W(e)| = 1$ if $e \in B$. Indeed, the expected cardinality of $W(e)$ is small also for the remaining edges.

**Lemma 16.** For any edge $e \in S^\ast$ at level $k_e \leq |R| - 1$ (edges incident to the root are at level one), one has

$$\Pr[|W(e)| = q] = \begin{cases} \frac{1}{2^q} & \text{if } 1 \leq q \leq k_e; \\ \frac{2}{2^q} & \text{if } q = k_e; \\ 0 & \text{otherwise.} \end{cases}$$

**Proof.** Consider the path $v_0, v_1, \ldots, v_{k_e}, v_e$ from $e$ towards the root. In particular, $e = \{ v_0, v_1 \}$. If $(v_{q-1}, v_q)$ is the first edge from $B$ on this path, then $|W(e)| = q$. This is because, for each node $v_j, j \geq 1$, there is one distinct path $P_{uv}$ with $\{ u, v \} \in Y$ that contains $e$ (see also Figure 5(a)). This event happens with probability $1/2^q$. Otherwise, $|W(e)| = k_e$ by a
For $W \subseteq Y$, let $M(W)$ denote the first iteration when all the edge in $W$ are marked. The choice for the sequence $S' = S_1 \supseteq S_2 \supseteq \ldots$ is given by $S_t = \{ e \in S' \mid |M(W(e))| \geq t \}$. In other words, $S_t$ is the set of edges which are not yet deleted at the beginning of iteration $t$.

**Lemma 17.** The graph $S_t \cup \bigcup_{i=t+1}^{\infty} C_i'$ spans $R$.

**Proof.** Let $Y' \subseteq Y$ be the set of edges which are not yet marked at the beginning of iteration $t$ (see also Figure 5(b)). Then, by definition of bridges, $Y' \cup \bigcup_{i=t+1}^{\infty} C_i'$ spans $R$. Consider any edge $(u, v) \in Y'$. Then $(u, v) \in W(e)$ for all $e \in P_{uv}$. Hence no edge on $P_{uv}$ is deleted. Therefore $u$ and $v$ are also connected in $S_t$. The claim follows.

Recall that $H_q := \sum_{i=1}^{q} \frac{1}{i}$ is the $q$th harmonic number.

**Lemma 18.** Let $W \subseteq Y$. Then the expected number of iterations until all edges in $W$ are marked satisfies

$$E[M(W)] \leq H_{|W|} \cdot \Sigma.$$

**Proof.** Let $m_q$ be the best possible upper bound on the expected number of iterations until all out of a given set $W$ of $q$ edges are marked (over all feasible probability distributions). We will prove that $m_q \leq H_q \cdot \Sigma$ by induction on $q$. For $q = 1$, the only edge in $W$ is marked with probability at least $\frac{1}{2}$ in each iteration, hence $m_1 \leq \Sigma$. Next, let $q > 1$ and consider the first iteration. Consider the probability distribution $p = (p_0, \ldots, p_q)$ where $p_i$ gives the probability that $i$ edges are sampled in the first iteration. Of course, since the expected number of marked edges must be at least $q \cdot \frac{1}{2}$ in the first iteration, this distribution has to satisfy the constraint

$$\sum_{i=0}^{q} p_i = 1; \quad \sum_{i=0}^{q} i \cdot p_i \geq \frac{q}{2}; \quad p_i \geq 0, \forall i = 0, \ldots, q. \quad (1)$$

If we condition on the event that $i \in \{0, \ldots, q \}$ edges are marked in the first iteration, we need in expectation at most $m_{q-i}$ more iterations until the remaining $q-i$ edges are marked. Hence we obtain the bound

$$m_q \leq 1 + \sum_{i=0}^{q} p_i \cdot m_{q-i}. \quad (2)$$

Assume pessimistically that $p$ is the distribution which maximizes the right-hand side of (2) under Constraint (1). The value of $p$ is an optimal fractional solution of a linear program. In particular, we can assume that $p$ is a vertex of the polyhedron induced by (1). Hence all but (at most) two entries of $q$ are zero. Suppose $p_0 = 0$. In this case we would mark deterministically at least one edge. The claim follows since, conditioning on the number $i \in \{0, \ldots, q \}$ of sampled edges, one obtains $m_q \leq 1 + m_{q-i} \leq 1 + H_{q-i} \cdot \Sigma \leq H_q \cdot \Sigma$. Here we use $q \leq |Y| = |R| - 1 \leq \Sigma$. Otherwise, there must be an $i > 0$ such that $p_i = \frac{1}{\Sigma}$, $p_0 = 1 - \frac{1}{\Sigma}$, and $p_j = 0$ for all $0 < j \neq i$. Hence

$$m_q \leq 1 + \frac{q}{2\Sigma} \cdot H_{q-i} \cdot \Sigma + \left(1 - \frac{q}{2\Sigma}\right)m_q.$$

Rearranging terms yields

$$m_q \leq \Sigma \cdot \left(\frac{q}{2} + H_{q-i}\right) \leq \Sigma \cdot H_q,$$

and the assertion follows.

Eventually, we prove the expected approximation of $\ln(4)$, as claimed in Theorem 3.

**Proof of Theorem 3.** For an edge $e \in S^*$, we define $D(e) = \max\{t \mid e \in S^t\}$ as the iteration in which $e$ is deleted. One has

$$E[D(e)] = \sum_{q=1}^{k_e} \Pr[W(e)] \cdot q \cdot E[D(e) \mid W(e)] = q.$$

**Lemma 18.**

$$\sum_{q=1}^{k_e} \Pr[W(e)] = q \cdot H_q \cdot \Sigma$$

**Lemma 16.**

$$\sum_{q=1}^{k_e-1} \left(\frac{1}{q}\right)^q \cdot H_q \cdot \Sigma \leq \sum_{q=1}^{k_e-1} \left(\frac{1}{q}\right)^q \cdot H_q \cdot \Sigma \leq \sum_{q=1}^{k_e-1} \left(\frac{1}{q}\right)^q = \ln(4) \cdot \Sigma.$$

The expected cost of the approximate solution satisfies

$$E \left[\sum_{i \geq 1} c(C_i')\right] \leq \sum_{i \geq 1} E \left[\text{opt}_{i} f_{i}\right] \leq \sum_{i \geq 1} E \left[c(S_i')\right] \leq \frac{1}{\Sigma} \sum_{e \in S^*} E[D(e)] \cdot c(e) \leq \ln(4) \cdot \text{opt}_h.$$

The claim follows.

### 4.1 Derandomization

After submitting the preliminary version of this paper, we found a way to derandomize our algorithm via the method of limited independence (see, e.g., Alon and Spencer [2]). The basic idea is to partition the sequence of iterations into a (large) constant number of phases. In each phase, we sample a proper number of random components (rather than just one component). The LP is updated only from phase to phase. This sampling is performed in such a way that only $O(\log n)$ random bits are needed. Hence, the algorithm can be derandomized by considering all the (polynomially-many) choices for the random bits. The approximation ratio grows by a factor $(1 + \varepsilon)$. The proof of the following theorem will appear in the journal version of this paper.

**Theorem 19.** For any $k = O(1)$ and any constant $\varepsilon > 0$, there is a polynomial time deterministic $(\ln(4) + \varepsilon)$-approximation algorithm for $k$-restricted Steiner tree.

**Corollary 20.** For any constant $\varepsilon > 0$, there is a polynomial-time deterministic $(\ln(4) + \varepsilon)$-approximation algorithm for Steiner tree.

### 5. Integrality Gap

In this section we bound the integrality gap of $k$-DCR. Note that, despite the fact that our analysis is based on an LP relaxation of the problem, it does not imply a $\ln(4)$ (nor even a 1.5) bound on the integrality gap of the studied
Lemma 8. For this reason, the proof of Theorem 4 is postponed to the full version of the paper.

Proof. Consider the algorithm from Figure 3 with $\mu = \delta \Sigma$ and $\delta = \ln(2)$. Observe that, for any $t \geq 1$, $opt_f^{t,t} \leq opt_f^t$ (contracting components does not increase the cost of the fractional solution). Hence

$$E[c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)]$$

\[
\leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\mu} opt_f^t + \frac{1}{\Sigma} \sum_{t=1}^{\mu} opt_{f,t}^t \\
\leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\mu} opt_f^t + \frac{\mu}{\Sigma} opt_f^t \\
\leq (2e^{-\delta} + \delta) opt_f^t = (1 + \ln(2)) opt_f^t.
\]

In order to achieve the better 1.55 bound on the integrality gap claimed in Theorem 4, we prove that another algorithm, namely the algorithm of Robins and Zelikovsky [38], produces solutions of cost bounded with respect to the optimal fractional solutions to $k$-DCR. Our alternative analysis of this algorithm is, to some extent, inspired by an analogous argument of Charikar and Guha [9] in the context of the facility location problem. Our argument is essentially a combination of the analysis in [38] with our new Bridge Lemma 8. For this reason, the proof of Theorem 4 is postponed to the full version of the paper.

We leave it as an interesting open problem to prove a $\ln(4)$ (or even 1.5) bound on the integrality gap of $k$-DCR (if possible). This might involve the development of a fractional version of Lemma 11.

We conclude the paper with a comparison between BCR and $k$-DCR. It is easy to see that any feasible solution to $k$-DCR can be turned into a feasible solution to BCR of the same cost. In fact, it is sufficient to split each component into the corresponding set of edges.

Interestingly enough, the reverse is not true, as observed in [34]. In other words, for any $k$, $k$-DCR is a relaxation strictly stronger than BCR. In particular, the 1.55 upper bound on the integrality gap of $k$-DCR does not imply the same bound on the integrality gap of BCR. Nevertheless, Skutella's graph [31] implies an 8/7 lower bound also on the integrality gap of our relaxation. It remains as a challenging open problem to show whether the integrality gap of BCR is smaller than 2 or not.

Acknowledgements.

The authors wish to thank C. Chekuri, F. Eisenbrand, M. X. Goemans, J. Könemann, D. Pritchard, F. B. Shepherd, and R. Zenklusen for helpful discussions. The second author is grateful to F. Eisenbrand for supporting his visit at EPFL (during which part of this paper was developed). We thank the anonymous reviewers for many suggestions on how to improve the presentation.

6. REFERENCES


