An Improved LP-based Approximation for Steiner Tree*

Jarosław 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 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 $\ln(4)+\varepsilon < 1.39$ 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) $c: E \to \mathbb{Q}^+$, 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 $\frac{96}{95}$ 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,

*Extended abstract.

†Partially supported by MNISW grant number N N206 1723 33, 2007 - 2010.

‡Developed while visiting EPFL.

§Supported by Swiss National Science Foundation within the project "Robust Network Design"

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA

[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.

38, 42], culminating with the famous $1+\frac{\ln(3)}{2}+\varepsilon < 1.55$ approximation algorithm by Robins and Zelikovsky [38] (here $\varepsilon > 0$ is an arbitrary small constant).

All the mentioned improvements are based on the following idea. A *full component* (or, for short, *component*) of a Steiner tree is a maximal subtree whose terminals coincide with its leaves. Note that the edge set of the Steiner tree is partitioned by its components. A *k-restricted* Steiner tree is a Steiner tree whose components contain no more than k terminals (*k-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, $opt := c(Opt)$ and $opt_k := c(Opt_k)$.

THEOREM 1. *[6] Let ρ_k be the k-Steiner ratio, i.e. the supremum of the ratio opt_k/opt. Then $\rho_k = \frac{(r+1)2^r+s}{r2^r+s} \leq 1+\frac{1}{\lfloor \log_2 k \rfloor}$, where r and s are non-negative integers such that $k = 2^r + s$ and $s < 2^r$.*

We remark that, given an optimal k-restricted Steiner tree S^*, its components are optimal Steiner trees connecting the corresponding terminals. For any fixed k, a list $\{Z_1, \ldots, Z_q\}$, $q = O(n^k)$, 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 tree[2] on terminals R'. Unfortunately, selecting the cheapest subset of $\{Z_1, \ldots, Z_q\}$ spanning the terminals is an **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

$$\min \sum_{e \in E} c(e) z_e \qquad \text{(BCR)}$$

$$\sum_{e \in \delta^+(U)} z_e \geq 1 \qquad \forall U \subseteq V \setminus \{r\} : U \cap R \neq \emptyset$$

$$z_e \geq 0 \qquad \forall e \in E.$$

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-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.

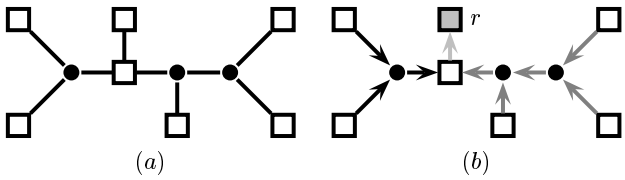

Figure 1: (a) A 4-restricted Steiner tree S, where rectangles denote terminals and circles represent Steiner nodes. (b) Edges of S are directed towards a root r. The resulting directed components are depicted with different colors.

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

[2]We recall that, given k terminals, the dynamic-programming algorithm by Dreyfus and Wagner [11] computes an optimal Steiner tree among them in $O(3^k n + 2^k n^2 + n^3)$ worst-case time. A faster parameterized algorithm can be found in [33].

of one (*sink*) terminal u_i in it, and direct all the edges of C towards u_i. 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 $sink(C_j)$ be the sink terminal of C_j, and $sources(C_j) := V(C_j) \cap R \setminus \{sink(C_j)\}$ 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:

$$\begin{aligned} \min \sum_j c(C_j) x_j \qquad & (k\text{-DCR}) \\ \sum_{C_j \in \delta_k^+(U)} x_j \quad &\geq \quad 1 \quad \forall U \subseteq R \setminus \{r\},\ R \neq \emptyset \\ x_j \quad &\geq \quad 0 \quad \forall j = 1, \ldots, h. \end{aligned}$$

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 Opt_k. As already observed, we can direct all its edges towards an arbitrary root terminal r. At this point, 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 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 $sink(C_j)$, 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 $\ln(4)$ (see Section 4). A $\ln(4)\rho_k \leq \ln(4)(1 + \frac{1}{\lfloor \log_2 k \rfloor}) < 1.39$ approximation for Steiner tree immediately follows from Theorem 1 by choosing k large enough[3]. 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 [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 $1 + \frac{\ln(3)}{2} < 1.55$ 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]. Könemann, Pritchard and Tan [31] showed that for a different LP formulation, which is stronger than BCR, the integrality gap is upper-bounded by $\frac{2b+1}{b+1}$, 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 Opt_k^f, and opt_k^f is its cost. For a given (directed or undirected) component C', $R(C') := R \cap V(C')$ is the set of its termi-

[3]Observe that our approach provides approximation factors strictly better than 2 for any $k \geq 6$, since $\rho_6 = 1.4$

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 $u \in sources(C_j)$, and one edge $e_j = (v_j, sink(C_j))$. Set the capacity of each 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 $s \in R \setminus \{r\}$, there is an s-r cut $U' \subseteq V'$ of minimum capacity $\sum_{e \in \delta^+(U')} w(e)$, consisting of edges e_j only. Moreover, given a non-empty subset $R' \subseteq R \setminus \{r\}$, there is a cut U' such that

$$\sum_{e \in \delta^+(U')} w(e) = \sum_{e_j : C_j \in \delta_k^+(R')} w(e_j) = \sum_{C_j \in \delta_k^+(R')} x_j.$$

It then follows that a non-empty subset $R' \subseteq R \setminus \{r\}$ minimizing $\sum_{C_j \in \delta_k^+(R')} x_j$ can be computed via $|R|-1$ minimum-cut computations in G'. The claim follows. □

Note that one can solve k-DCR alternatively by a compact LP formulation which computes a minimum-cost multicommodity flow in G' (setting the cost of e_j to $c(C_j)$). This can be done even in strongly-polynomial time using the framework of Frank and Tardos [17] (see also Theorem 6.6.3 in [24]). Details will appear in the full version of the paper.

Let T^0 be a minimum-cost terminal spanning tree, i.e., T^0 spans R, but does not contain any Steiner node. It is a well-known fact that $c(T^0) \leq 2 \cdot opt$ (see e.g. Theorem 3.3 in [41]). Extending the standard proof, this bound also holds w.r.t. our LP relaxation:

LEMMA 6. *For any k, $c(T^0) \leq 2 \cdot opt_k^f$.*

PROOF. For each component C_j of Opt_k^f, obtain a TSP tour on $R(C_j)$ of cost at most $2c(C_j)$, remove one edge of the tour, and direct the remaining edges towards $sink(C_j)$. This induces a fractional solution to k-DCR of cost at most $2 \cdot opt_k^f$, with the property that only components with 2 terminals and without Steiner nodes are used. This provides a feasible fractional solution to BCR of the same cost. Since BCR without Steiner nodes is integral [12], the claim follows. □

We next prove our Bridge Lemma, which is the heart of our analysis. This lemma relates the cost of any terminal spanning tree to the cost of any fractional solution to k-DCR via the notion of bridges, and its proof is based on Edmonds' Theorem 2.

Before proving the lemma, we need a few intermediate results. Let R' be a subset of k' terminals. Consider a given tree S, with edge weights c, containing the terminals R'. The weight function c associated to S will be clear from the context. Let us collapse the terminals R' into one node, and consider the minimum-cost tree $S' \subseteq S$ in the resulting (possibly, multi-)graph, spanning all nodes of S. Observe that S' will contain all the edges of S but $k'-1$ edges, since collapsing R' decreases the number of nodes in S by $k'-1$.

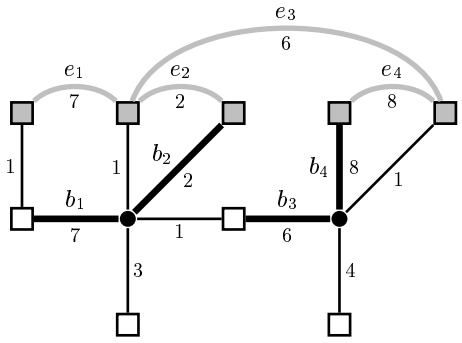

Figure 2: Steiner tree S is drawn in black. Terminals of R' are gray shaded. Bold black edges indicate $Br_S(R') = \{b_1, \ldots, b_4\}$. The corresponding edges $e_1, \ldots, e_4$ of Y' are drawn in gray and labeled with $w(e_i)$. Note that $w(e_i) = c(b_i)$. Observe also that b_3 is the unique bridge on the cycle contained in $S \cup \{e_3\}$.

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

$$\text{argmax}\Big\{c(B) \mid B \subseteq S,\ S \setminus B \cup \tbinom{R'}{2} \text{ connects } V(S)\Big\}.$$

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* $w : R \times R \to \mathbb{Q}_+$ 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 $i = 1, \ldots, k'-1$,*

- **(a)** *$w(e_i) = c(b_i)$ (hence $w(Y') = br_S(R')$).*
- **(b)** *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 $F_{i'}$ and $F_{i''}$. For each b_i, we add edge $e_i = \{r_{i'}, r_{i''}\}$ to Y'. Clearly Y' is a spanning tree on R'. The path P_i between $r_{i'}$ and $r_{i''}$ 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.

(1) For $t = 1, 2, \ldots, \mu$
  (1a) Compute an optimal fractional solution x^t to k-DCR (w.r.t. the current instance).
  (1b) Sample one component C^t, where $C^t = C_j$ with probability $x_j^t / \sum_i x_i^t$. Contract C^t.
(2) Compute a terminal spanning tree T^μ in the remaining instance.
(3) Output $T^\mu \cup \bigcup_{t=1}^{\mu} C^t$.

Figure 3: A 3/2-approximation algorithm for k-restricted Steiner tree.

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 $y : R \times R \to \mathbb{Q}_+$ as follows: For every j, install capacity x_j in a cumulative manner on Y_j. In other terms, $y(u,v) := \sum_{Y_j \ni (u,v)} x_j$. 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. $w(F) \leq \sum_{e \in R \times R} w(e)y(e)$.

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 $w(F) \geq c(T)$ (see, e.g., Theorem 6.2 in [32]). Altogether

$$\begin{aligned} \sum_j x_j br_T(C_j) &= \sum_j x_j w(Y_j) \\ &= \sum_{e \in R \times R} w(e)y(e) \geq w(F) \geq c(T). \quad \square \end{aligned}$$

3. ITERATIVE RANDOMIZED ROUNDING

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^t be the optimal fractional solution to k-DCR at a generic iteration t. By *sampling* a component C^t, we mean selecting one of the components C_j with probability $x_j^t / \sum_i x_i^t$. *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 $\Sigma^t := \sum_i x_i^t$ 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 μ of iterations is fixed to $\delta\Sigma$, where δ 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_k^t be the optimal k-restricted Steiner tree at the beginning of iteration t, and let opt_k^t be its cost. By $opt_k^{f,t} := \sum_j x_j^t \cdot c(C_j)$ 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 T^μ. The basic idea is showing that the current terminal spanning tree is getting cheaper by a factor $(1 - \frac{1}{\Sigma})$ 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 $\frac{1}{\Sigma}$ fraction of $opt_k^{f,t}$, which is in turn upper bounded by opt_k^t. Hence it is sufficient to bound the cost of opt_k^t (Corollary 12). In order to do that, we show that the cost of the optimal Steiner tree decreases by a factor $(1 - \frac{1}{2\Sigma})$ 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.

LEMMA 9. *One has* $E[c(T^\mu)] \leq \left(1 - \frac{1}{\Sigma}\right)^\mu \cdot 2opt_k^f$.

PROOF. Let T^t (T^0, resp.) be the minimum-cost terminal spanning tree at the end of iteration t (for the original instance, 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:

$$\begin{aligned} E[c(T^t)] &\leq c(T^{t-1}) - E[br_{T^{t-1}}(C^t)] \\ &= c(T^{t-1}) - \frac{1}{\Sigma} \sum_j x_j^t \cdot br_{T^{t-1}}(C_j) \\ &\overset{\text{Bridge Lem 8}}{\leq} \left(1 - \frac{1}{\Sigma}\right) \cdot c(T^{t-1}). \end{aligned}$$

It follows that

$$E[c(T^\mu)] \leq \left(1 - \frac{1}{\Sigma}\right)^\mu \cdot c(T^0) \overset{\text{Lem 6}}{\leq} \left(1 - \frac{1}{\Sigma}\right)^\mu \cdot 2opt_k^f. \quad \square$$

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]).

LEMMA 10. *For any Steiner tree S, $br_S(R) \geq \frac{1}{2}c(S)$.*

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 $B \subseteq S$ be the set of marked edges. Observe that $c(B) \geq \frac{1}{2}c(S)$. Furthermore, after contracting R, one can remove B while keeping S connected. From the definition of bridges it follows that $br_S(R) \geq c(B) \geq \frac{1}{2}c(S)$. $\square$

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

LEMMA 11. *Let S be any Steiner tree and x be a feasible solution to k-DCR. 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*

$$E[c(S')] \leq \left(1 - \frac{1}{2\Sigma}\right) \cdot c(S).$$

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

$$S' := S \backslash \{b_i \mid e_i \in Br_Y(C)\}.$$

(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 $e_i = \{u_i, v_i\} \in Y \backslash Br_Y(C)$. The u_i-v_i path in S contains only one potential bridge edge, namely b_i. Hence, if two terminals are connected by $Y \backslash Br_Y(C)$, then they are so in S'. The claim follows since $(Y \backslash Br_Y(C)) \cup C$ is connected due to the definition of bridges. We conclude

$$\begin{aligned} E[c(S')] &= c(S) - E[br_Y(C)] \\ &= c(S) - \frac{1}{\Sigma} \sum_j x_j br_Y(C_j) \\ &\overset{\text{Bridge Lem 8}}{\leq} c(S) - \frac{1}{\Sigma} w(Y) \\ &= c(S) - \frac{1}{\Sigma} br_S(R) \\ &\overset{\text{Lem 10}}{\leq} \left(1 - \frac{1}{2\Sigma}\right) c(S). \quad \square \end{aligned}$$

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 $E[opt_k^{t+1}] \leq \left(1 - \frac{1}{2\Sigma}\right) \cdot opt_k^t$ for any iteration $t = 1, \ldots, \mu - 1$. The claim follows. $\square$

Corollary 12 immediately provides an upper bound on $opt_k^{f,t}$. An upper bound on the expected cost of the sampled components easily follows.

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

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

PROOF. One has

$$\begin{aligned} E[c(C^t)] &= \frac{1}{\Sigma} E\left[\textstyle\sum_j x_j^t \cdot c(C_j)\right] \\ &= \frac{1}{\Sigma} E[opt_k^{f,t}] \\ &\leq \frac{1}{\Sigma} E[opt_k^t] \\ &\overset{\text{Cor 12}}{\leq} \frac{1}{\Sigma} \left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot opt_k. \quad \square \end{aligned}$$

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

$$\begin{aligned} & E\left[\frac{c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)}{opt_k}\right] \\ &\overset{\text{Lem 9 + 13}}{\leq} 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\mu} + \frac{1}{\Sigma} \sum_{t=1}^{\mu} \left(1 - \frac{1}{2\Sigma}\right)^{t-1} \\ &= 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\delta \cdot \Sigma} + 2 - 2 \cdot \left(1 - \frac{1}{2\Sigma}\right)^{\delta \cdot \Sigma} \\ &\leq 2e^{-\delta} + 2 - 2 \cdot e^{-\delta/2} = \frac{3}{2}. \end{aligned}$$

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

(1) For $t = 1, 2, \ldots$
  (1a) Compute an optimal fractional solution x^t to k-DCR (w.r.t. the current instance).
  (1b) Sample one component C^t, where $C^t = C_j$ with probability $x_j^t / \sum_i x_i^t$. Contract C^t.
  (1c) If the instance consists only of the root, return $\bigcup_{i=1}^t C^i$.

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

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 $S^* := Opt_k$ be the optimal k-restricted Steiner tree for the original instance (in particular, $c(S^*) = opt_k$). 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 $Y' \subseteq Y$, then they are as well connected by $\{e \in S^* \mid W(e) \cap Y' \neq \emptyset\}$.

At each iteration, when component C^t is sampled, we mark a proper *random* subset $\overline{Br}_Y(C^t)$ of edges of Y. This set guarantees that $(Y \backslash \overline{Br}_Y(C^t)) \cup C^t$ is connected (deterministically). The intuitive reason for using $\overline{Br}_Y(C^t)$ 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,

we delete e from S^*. 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 $\overline{Br}_Y(C^t)$. *Delete* an edge e from S^* as soon as all edges in $W(e)$ are marked.

The subgraph S^t is given by the edges of S^* 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 $\overline{Br}_Y(C_j)$ such that every edge in Y is marked with probability at least $\frac{1}{\Sigma}$ 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 $\overline{Br}_Y(C_j) \subseteq Y$ such that if we sample a component C randomly w.r.t. x one has*

(1) $\Pr[e \in \overline{Br}_Y(C)] \geq \frac{1}{\Sigma}$ *for all* $e \in Y$;

(2) $Y \backslash \overline{Br}_Y(C_j) \cup C_j$ *connects* $V(Y)$ *for all* C_j.

PROOF. For a component C_j, let the set of *candidate* bridges $\mathcal{B}_Y(C_j)$ be

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

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

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

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

$$\Pr[e \in \overline{Br}_Y(C)] = \sum_{(B,j): B \in \mathcal{B}_Y(C_j), e \in B} \frac{x_j}{\Sigma} \cdot w_{jB} \geq \frac{1}{\Sigma}.$$

Suppose by contradiction that such a distribution w does not exist. Then the following system of linear inequalities has no solution[5]

$$\begin{aligned} \sum_{B \in \mathcal{B}_Y(C_j)} w_{jB} &\leq 1 \quad \forall j \\ \sum_{(B,j): B \in \mathcal{B}_Y(C_j), e \in B} x_j w_{jB} &\geq 1 \quad \forall e \in Y \\ w &\geq \mathbf{0}. \end{aligned}$$

Farkas' Lemma[6] yields that there is a vector $(y, c) \geq \mathbf{0}$ with

$$(a) \qquad y_j \geq \sum_{e \in B} c_e x_j \quad \forall (B, j) : B \in \mathcal{B}_Y(C_j);$$

$$(b) \qquad \sum_j y_j < \sum_{e \in Y} c_e = c(Y).$$

In particular,

$$y_j \overset{(a)}{\geq} x_j \cdot \max\{c(B) \mid B \in \mathcal{B}_Y(C_j)\} = x_j \cdot br_Y(C_j).$$

[5]We can replace the "=" constraint with "$\leq$" without affecting feasibility since all coefficients of w_{jB} are non-negative.
[6]$\exists x \geq \mathbf{0} : Ax \leq b \,\dot\vee\, \exists z \geq \mathbf{0} : z^T A \geq 0, z^T b < 0$

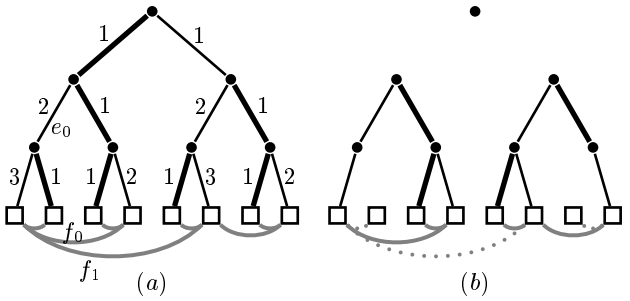

Figure 5: (a) Optimal Steiner tree S^* in black, where bold edges indicate B, and the associated terminal spanning tree Y in gray. Edges e in S^* are labeled with $|W(e)|$. For example $W(e_0) = \{f_0, f_1\}$. (b) Marked edges in Y at a given iteration t are drawn dotted; the non-deleted edges in S^* (i.e. edges of S^t) are drawn in black. Non-marked edges of Y and non-deleted edges of S^* support the same connectivity on R.

Then

$$\sum_j x_j \cdot br_Y(C_j) \leq \sum_j y_j \overset{(b)}{<} c(Y),$$

which contradicts the Bridge Lemma 8. □

Next, we define the witness tree Y and the sets $W(e)$ for each $e \in S^*$. Without loss of generality any Steiner node in S^* has degree 3 or more. By adding dummy Steiner nodes and zero cost edges, we can assume that S^* is a (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 two edges to its children. Let B denote the chosen edges. Clearly $\Pr[e \in B] = \frac{1}{2}$ for any $e \in S^*$. Let $P_{uv} \subseteq S^*$ be the unique u-v path in S^*. The witness tree is

$$Y := \{\{u, v\} \in \tbinom{R}{2} \mid |P_{uv} \cap B| = 1\}.$$

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

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

See Figure 5(a) for an illustration. Note that $1 \leq |W(e)| \leq |Y| = |R| - 1$. Observe also 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 $e \in S^*$ at level $k_e \leq |R| - 1$ (edges incident to the root are at level one), one has*

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

PROOF. Consider the path $v_0, v_1, \ldots, v_{k_e}$ 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

similar argument. The latter event happens with probability $1/2^{k_e}$. The claim follows. □

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 $S^t \cup \bigcup_{t'=1}^{t-1} C^{t'}$ 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, $Y' \cup \bigcup_{t'=1}^{t-1} C^{t'}$ 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 *qth 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 $\frac{1}{\Sigma}$ 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 $q \cdot \frac{1}{\Sigma}$ in the first iteration, this distribution has to satisfy the constraint

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

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

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

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 p 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 $i \in \{1, \ldots, q\}$ 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 $p_i = \frac{q}{i\Sigma}$, $p_0 = 1 - \frac{q}{i\Sigma}$, and $p_j = 0$ for all $0 < j \neq i$. Hence

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

Rearranging terms yields

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

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

$$\begin{aligned}
E[D(e)] &= \sum_{q=1}^{k_e} \Pr[|W(e)| = q] \cdot E[D(e) \mid |W(e)| = q] \\
&\overset{\text{Lem 18}}{\leq} \sum_{q=1}^{k_e} \Pr[|W(e)| = q] \cdot H_q \cdot \Sigma \\
&\overset{\text{Lem 16}}{=} \sum_{q=1}^{k_e - 1} \Big(\frac{1}{2}\Big)^q \cdot H_q \cdot \Sigma + \frac{2}{2^{k_e}} \cdot H_{k_e} \cdot \Sigma \\
&\leq \sum_{q \geq 1} \Big(\frac{1}{2}\Big)^q \cdot H_q \cdot \Sigma \\
&= \Sigma \cdot \sum_{q \geq 1} \frac{1}{q} \sum_{i \geq 0} \Big(\frac{1}{2}\Big)^{q+i} \\
&= \Sigma \cdot \sum_{q \geq 1} \frac{1}{q} \Big(\frac{1}{2}\Big)^{q-1} = \ln(4) \cdot \Sigma.
\end{aligned}$$

The expected cost of the approximate solution satisfies

$$\begin{aligned}
E\Big[\sum_{t \geq 1} c(C^t)\Big] &= \sum_{t \geq 1} \frac{1}{\Sigma} E\big[opt_k^{f,t}\big] \\
&\leq \frac{1}{\Sigma} \sum_{t \geq 1} E\left[c(S^t)\right] \\
&= \frac{1}{\Sigma} \sum_{e \in S^*} E[D(e)] \cdot c(e) \leq \ln(4) \cdot opt_k.
\end{aligned}$$

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

LP. (This is because the LP changes during the iterations of the algorithm). However, an easy adaptation of the analysis from previous sections proves the following claim.

THEOREM 21. *There is a polynomial-time algorithm which computes a solution to the k-restricted Steiner tree problem of expected cost at most $1 + \ln(2) < 1.694$ times the cost of the optimal fractional solution to k-DCR.*

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

$$\begin{aligned} & E\Big[c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)\Big] \\ & \quad \overset{\text{Lem 9}}{\leq} \quad 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^\mu opt_k^f + \frac{1}{\Sigma} \sum_{t=1}^{\mu} opt_k^{f,t} \\ & \quad \leq \quad 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^\mu opt_k^f + \frac{\mu}{\Sigma} opt_k^f \\ & \quad \leq \quad (2e^{-\delta} + \delta) opt_k^f = (1 + \ln(2)) opt_k^f. \quad \square \end{aligned}$$

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

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24(3): 440–456, 1995.

[2] N. Alon and J. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., Hoboken, NJ, third edition, 2008.

[3] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *FOCS*, pages 427–436, 2009.

[4] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[5] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32(4):171–176, 1989.

[6] A. Borchers and D. Du. The k-Steiner ratio in graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997.

[7] G. Borradaile, C. Kenyon-Mathieu, and P. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *SODA*, pages 1285–1294, 2007.

[8] D. Chakrabarty, N. R. Devanur, and V. V. Vazirani. New geometry-inspired relaxations and algorithms for the metric Steiner tree problem. In *IPCO*, pages 344–358, 2008.

[9] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM Journal on Computing*, 34:803–824, 2005.

[10] M. Chlebík and J. Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, 2008.

[11] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.

[12] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, B71:233–240, 1967.

[13] F. Eisenbrand and F. Grandoni. An improved approximation algorithm for virtual private network design. In *SODA*, pages 928–932, 2005.

[14] F. Eisenbrand, F. Grandoni, G. Oriolo, and M. Skutella. New approaches for virtual private network designs. *SIAM Journal on Computing* 37(3): 706-721, 2007.

[15] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *SODA*, pages 1174–1183, 2008.

[16] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Connected Facility Location via Random Sampling and Core Detouring. *Journal of Computer and System Sciences*. To appear.

[17] A. Frank and É. Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7:49–65, 1987.

[18] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32:826–834, 1977.

[19] M. R. Garey and D. S. Johnson. *Computers and*

intractability. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[20] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, January 1968.

[21] M. X. Goemans and Y. S. Myung. A catalog of Steiner tree formulations. *Networks*, 23:19–28, 1993.

[22] M. X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[23] F. Grandoni and G. F. Italiano. Improved approximation for single-sink buy-at-bulk. In *ISAAC*, pages 111–120, 2006.

[24] M. Grötschel, L. Lovasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[25] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *STOC*, pages 389–398, 2001.

[26] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost sharing: simpler and better approximation algorithms for network design. *Journal of the ACM*, 54(3):11, 2007.

[27] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner tree problem*. Monograph in Annals of Discrete Mathematics, 53. Elsevier, Amsterdam, 1992.

[28] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. In *FOCS*, pages 448–457, 1998.

[29] M. Karpinski and A. Zelikovsky. New approximation algorithms for the Steiner tree problems. *Journal of Combinatorial Optimization*, 1(1):47–65, 1997.

[30] L. G. Khachiyan. A polynomial algorithm for linear programming. *Soviet Math. Doklady*, 20:191–194, 1979. (Russian original in Doklady Akademiia Nauk SSSR, 244:1093–1096).

[31] J. Könemann, D. Pritchard, and K. Tan. A partition-based relaxation for Steiner trees. *CoRR*, abs/0712.3568, 2007.

[32] B. Korte and J. Vygen. *Combinatorial Optimization - Theory and Algorithms*. Springer-Verlag, Second Edition, 2002.

[33] D. Mölle, S. Richter, and P. Rossmanith. A faster algorithm for the Steiner tree problem. In *STACS 2006*, pages 561–570, 2006.

[34] T. Polzin and S. V. Daneshmand. On Steiner trees and minimum spanning trees in hypergraphs. *Operations Research Letters*, 31:12–20, 2003.

[35] H. J. Prömel and A. Steger. A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. *Journal of Algorithms*, 36:89–101, 2000.

[36] S. Rajagopalan and V. V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *SODA*, pages 742–751, 1999.

[37] R. Rizzi. On Rajagopalan and Vazirani's 3/2-approximation bound for the Iterated 1-Steiner heuristic. *Information Processing Letters*, 86(6):335–338, 2003.

[38] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005. A preliminary version appeared in SODA 2000.

[39] C. Swamy and A. Kumar. Prima-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[40] K. Talwar. The single-sink buy-at-bulk LP has constant integrality gap. In *IPCO*, pages 475–486, 2002.

[41] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[42] A. Z. Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.