An Improved LP-based Approximation for Steiner Tree

Article *in* Proceedings of the Annual ACM Symposium on Theory of Computing · December 2010

DOI: 10.1145/1806689.1806769 · Source: OAI

Jaroslaw Byrka
University of Wroclaw

Fabrizio Grandoni
University of Lugano

Thomas Rothvoß
University of Washington Seattle

An Improved LP-based Approximation for Steiner Tree

Jarosław Byrka* Fabrizio Grandoni† Thomas Rothvoß‡ Laura Sanità§

November 5, 2009

Abstract

The *Steiner tree* problem is one of the most fundamental $\mathbf{NP}$-hard problems: given a weighted undirected graph and a subset of terminal nodes, find a minimum weight 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 the 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. This might have consequences for a number of related problems. Consider for example the *prize-collecting Steiner tree* problem, where one is allowed to connect a subset of the terminals, but has to pay a prize for each disconnected terminal. For long time the best-known approximation for this problem was 2 [Goemans,Williamson-SICOMP'95], and it was recently slightly improved to 1.992 [Archer et al.-FOCS'09]. An easy consequence of our integrality gap bound is a 1.94-approximation for this problem.

*Institute of Mathematics, EPFL, Lausanne, Switzerland, jaroslaw.byrka@epfl.ch

†Dipartimento di Informatica, Sistemi e Produzione, Università di Roma Tor Vergata, Via del Politecnico 1, 0013 Roma, Italy, grandoni@disp.uniroma2.it. Developed while visiting EPFL.

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

§Institute of Mathematics, EPFL, Lausanne, Switzerland, laura.sanita@epfl.ch. Supported by Swiss National Science Foundation within the project "Robust Network Design"

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 T spanning the terminals, of minimum cost $c(T) := \sum_{e \in T} c(e)$. Steiner tree is one of the classical and, probably, most fundamental problems in Computer Science and Operations Research, with great theoretical and practical relevance. The problem appears 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. [26]).

The Steiner tree problem appears already in the list of **NP**-hard problems in the book by Garey and Johnson [18]. In fact, it is **NP**-hard to find solutions of cost less than $\frac{96}{95}$ times the optimal cost [4, 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 is a 2-approximation for the Steiner tree problem [19, 37]. (This is known as the *minimum spanning tree heuristic*). A sequence of improved approximation algorithms appeared in the literature [28, 31, 34, 38], culminating with the famous $1 + \frac{\ln(3)}{2} + \varepsilon < 1.55$ approximation algorithm by Robins and Zelikovsky [34] (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 [5] shows that, in order to obtain a good approximation factor, it is sufficient to restrict our attention to k-restricted Steiner trees.

Theorem 1. *[5] Let ρ_k be the k-Steiner ratio, that is the best possible upper bound on the ratio of the cost of an optimal k-restricted Steiner tree compared to the cost of an optimal (unrestricted) Steiner tree. Then $\rho_k \leq 1 + \frac{1}{\lfloor \log_2 k \rfloor}$.*

We remark that, given an optimal k-restricted Steiner tree T, its components are optimal Steiner trees connecting the corresponding terminals. For any constant k, a list $\{U_1, U_2, \ldots, U_q\}$ of all potential k-components can be computed in polynomial time by considering all the subset R' of at most k terminals, and computing an optimal Steiner tree on terminals R'[2]. Unfortunately, selecting the cheapest subset of $\{U_1, U_2, \ldots, U_q\}$ spanning the terminals is an **NP**-hard problem already for $k \geq 4$ [17]. For this reason, [34] and previous papers rather select a subset of the U_i's with a local-search approach. The idea is to start with a minimum-cost terminal spanning tree (which is formed by 2-components), and iteratively improve it. At each step, one considers each U_i, and checks *how much* adding U_i to the current solution (and removing redundant edges) *improves* the solution itself. The algorithm each time selects the U_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

[1] We recall that 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.

[2] For $k = O(1)$, such an optimal Steiner tree can be computed in polynomial time using, say, the algorithm by Dreyfus and Wagner [12]

any further improvement after [34]. For this reason, it makes sense to search for alternative methods. One standard approach is to exploit a proper LP-relaxation. A natural formulation for the problem is the *undirected cut formulation* (see [20, 37]), 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 [20] or iterative rounding [27]. However, there is a simple example of a graph (namely, a cycle) showing an integrality gap of 2 already in the spanning tree case, i.e., when $R = V$ (see, e.g., [37]).

A more promising and well-studied LP is the so called *bi-directed cut relaxation* [7, 13, 32]. Let us fix an arbitrary terminal r as *root*. Replace each edge uv by two directed edges (u, v) and (v, u) of the same cost and define $\delta(S) = \{(u, v) \in E \mid u \in S,\ v \notin S\}$ as the set of edges, leaving S. The mentioned relaxation is

$$\begin{aligned} \min \sum_{e \in E} c(e) z_e & \qquad \text{(BCR)} \\ \sum_{e \in \delta(S)} z_e \geq 1 & \qquad \forall S \subseteq V \setminus \{r\} : S \cap R \neq \emptyset \\ z_e \geq 0 & \qquad \forall e \in E. \end{aligned}$$

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 [13] showed that the integrality gap of BCR is 1 in the spanning tree case.

Theorem 2. *[13] For $R = V$, the polyhedron of BCR is integral.*

The best-known lower bound on the integrality gap of BCR is $8/7$ [30, 37]. The best-known upper bound is 2, though BCR is believed to have a smaller integrality gap than the undirected cut relaxation [32]. The authors in [7] 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 [32].

Finding a better-than-2 LP-relaxation for the (k-restricted) Steiner tree problem is a long-standing open problem [7, 32]. 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. Good LPs are also important in the design of (practically) efficient and accurate heuristics.

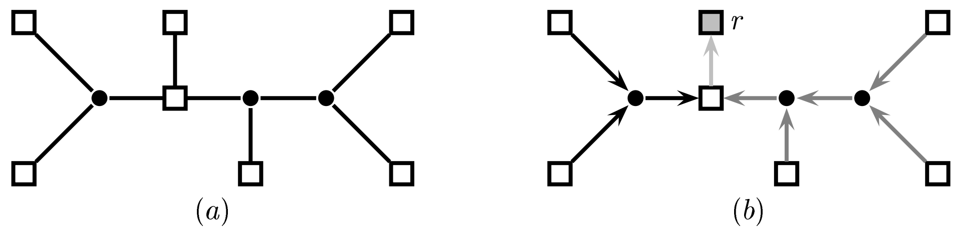
1.1 Our Results and Techniques

We next state the main result of this paper.

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

Our algorithm is based on the following *directed-component cut relaxation* for the k-restricted Steiner tree problem. Consider a k-restricted Steiner tree T with the edges directed towards the

Figure 1 In (a) we have a 4-restricted Steiner tree, where rectangles denote terminals and circles represent Steiner nodes. In (b) edges are directed towards a root r. The resulting directed components are depicted with different colors.



chosen root terminal r (see Figure 1). Consider the (undirected) k-components U_i introduced before. Make a copy C of each U_i for each choice of one (*sink*) terminal u_i in it, and direct all the edges of C towards u_i. Let $C_1, C_2, \ldots, C_h$ be the resulting *directed k-components*. Observe that h is polynomially bounded for any fixed k. We denote by $c(C_j)$ the cost of C_j, by $sink(C_j)$ the corresponding sink terminal and by $sources(C_j) := V(C_j) \cap R \setminus \{sink(C_j)\}$ the other (leaf) terminals (*sources* of C_j). We say that a component C_j *crosses* $S \subseteq V$ if it has at least one source in S and the sink outside. By $\delta'(S)$ we denote the set of components crossing S. Our LP-relaxation is then

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

$$\sum_{C_j \in \delta'(S)} x_j \geq 1 \quad \forall S \subseteq V \setminus \{r\} : R \cap S \neq \emptyset$$

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

Observe that if x is an optimum integral solution, then the union of all components C_j with $x_j = 1$ yields the cheapest k-restricted Steiner tree. Hence k-DCR is in fact a relaxation for the k-restricted Steiner tree problem. The LP above can be solved in polynomial time (see Section 2).

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 the value of x_j in the optimal fractional solution, and contract C_j into its sink node $sink(C_j)$. We reoptimize our LP w.r.t. the new instance and iterate. After a suitable number of iterations, the algorithm halts. With a simple analysis we can show that the cost of a minimum-cost terminal spanning tree plus the cost of the sampled components is (in expectation) at most 3/2 times the cost of the optimal k-restricted Steiner tree. With a refined analysis, we improve this bound to $\ln(4)$. A $\ln(4)(1 + \frac{1}{\lfloor \log_2 k \rfloor}) < 1.39$ approximation for Steiner tree immediately follows from Theorem 1 by choosing k large enough. This bound can be further improved for the special case of quasi-bipartite graphs, where all components form stars.

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 might turn out to be useful in 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 [13]. In our opinion, our analysis is simpler (or at least more intuitive) than the one in [34].

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 [7, 32] (for the k-restricted case). A more technical analysis, based on an adaptation of the analysis of Robins and Zelikovsky exploiting our Bridge Lemma, leads to the following improved result (see Section 4).

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 turn out to be useful to deal with variants and generalizations of the starting 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. We now provide one such application as an example. In the *prize-collecting Steiner tree* problem, we are given the same input as in the Steiner tree problem, plus a *prize* $p(v)$ for each terminal $v \in R$. The goal is finding a tree T spanning a subset $R' \subseteq R$ of terminals, which minimizes the sum of the cost $c(T)$ of the tree and the total prize $p(R \setminus R') := \sum_{v \in R \setminus R'} p(v)$ of the terminals not spanned by that tree. For a long time the best-known approximation for this problem was 2 [20]. Only very recently it has be improved to 1.992 via a rather complex algorithm and analysis [1]. An easy consequence of Theorem 4 is the following result (see Section 4).

Theorem 5. *There is a 1.94 approximation algorithm for the prize-collecting Steiner tree problem.*

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 Virtual Private Network [14, 15, 23], Single-Sink Rent-or-Buy [16, 24], Connected Facility Location [16, 35] and Single-Sink Buy-At-Bulk [22, 24, 36].

Both, the previously cited primal-dual and iterative rounding approximation techniques apply to a more general class of problems. In particular, the iterative rounding introduced by Jain [27] provides a 2-approximation for Generalized Steiner Network, and primal-dual algorithms developed by Goemans and Williamson [20] gave the same approximation factor for a large class of constrained forest problems.

Regarding LP relaxations for the Steiner tree problem and integrality gaps, upper bounds better than 2 are only known for special graph classes. In fact, BCR has an integrality gap smaller than 2 on quasi-bipartite graphs, where edges between Steiner nodes are forbidden. For such graphs Rajagopalan and Vazirani [32] (see also Rizzi [33]) gave an upper bound of $3/2$ on the gap which was recently improved to $4/3$ by Chakrabarty, Devanur and Vazirani [7]. Still, for this class of graphs the lower bound of $8/7$ holds [30, 37]. Könemann, Pritchard and Tan [30] showed that for a different LP formulation, which is stronger than the bi-directed cut relaxation, 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 [2] if the nodes are points in a fixed-dimensional Euclidean space, and using the algorithm of Borradaile, Kenyon-Mathieu and Klein [6] for planar graphs.

2 A Directed-Component Cut Relaxation

In this section we discuss some properties of our k-DCR relaxation which will be useful in the following.

We let Opt and Opt_k denote an optimum Steiner tree and an optimum k-restricted Steiner tree, respectively. Moreover, $opt := c(Opt)$ and $opt_k := c(Opt_k)$. The optimal fractional solution to k-DCR is denoted by Opt_k^f, and opt_k^f is its cost. For a given component C, $R(C) := R \cap C$.

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 [21, 29], since we can solve the separation problem in polynomial time.

Lemma 6. *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', containing initially only R. Then for each j add private copies of Steiner nodes of C_j and edges with capacity x_j, which are directed towards $sink(C_j)$. Let $G' = (V', E')$ be this graph with capacity function $w : E' \to \mathbb{Q}_+$. Let e_j denote the edge of capacity $w(e_j) = x_j$ in G' that is ingoing to the sink of C_j. The crucial observation is that for a terminal $s \in R$, w.l.o.g. any s-r MinCut $U \subseteq V'$ consists just of edges e_j. Then for $S := U \cap R$

$$\sum_{C_j \in \delta'(S)} x_j = \sum_{C_j \in \delta'(S)} w(e_j) = \sum_{e \in \delta(U)} w(e)$$

and a violated inequality can be found with $|R|$ many MinCut applications in G'. $\square$

Let T^0 be a minimum-cost terminal spanning tree, i.e., T^0 spans R, but does not contain any further Steiner node. It is a well-known fact that $c(T^0) \leq 2 \cdot opt$ (again see, e.g., [37]). Using the same standard proof, this bound also holds w.r.t. our LP relaxation:

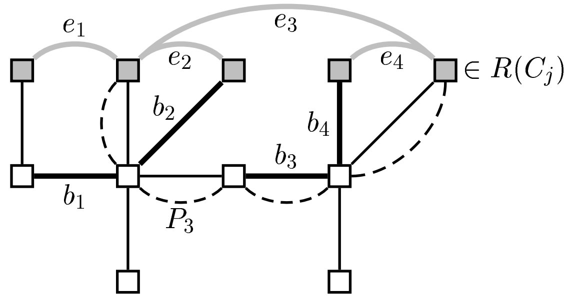
Lemma 7. *For any k, $c(T^0) \leq 2 \cdot opt_k^f$.*

Proof. For each component C_j obtain a TSP tour on $R(C_j)$ of cost at most $2c(C_j)$. Direct the edges of that tour towards $sink(C_j)$. This gives a fractional solution to k-DCR of cost $2 \cdot opt_k^f$ 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 [13], the claim follows. $\square$

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.

Consider any component C_j in Opt_k^f. Let $k_j := |R(C_j)|$. The *bridges* $Br_T(C_j)$ of T w.r.t. C_j are the $k_j - 1$ many edges in T that are not contained in a minimum spanning tree (using only edges of T) after the contraction of C_j. Let us abbreviate $br_T(C_j) := c(Br_T(C_j))$. We next describe

Figure 2 Tree T with $Br_T(C_j) = \{b_1, \dots, b_4\}$ drawn bold. The upper 5 terminals (gray shaded) are in $R(C_j)$. Edges $e_1, \dots, e_4$ of Y_j are drawn gray. Note that $w(e_i) = c(b_i)$. The dashed path depicts P_3.



a procedure which constructs an undirected graph Y_j on nodes $R(C_j)$, with edge weights $w(e)$, $e \in E(Y_j)$, such that $w(Y_j) = br_T(C_j)$. Let $Br_T(C_j) = \{b_1, b_2, \dots, b_{k_j-1}\}$. For each b_ℓ, let P_ℓ be the path in T containing b_ℓ and no other bridge in $Br_T(C_j)$, with both endpoints u_ℓ and v_ℓ in $R(C_j)$. We add edge $u_\ell v_\ell$ with cost $w(u_\ell v_\ell) := c(b_\ell)$ to Y_j. Note that such a path P_ℓ exists and is unique since, by construction, $C_j \cup (T \setminus Br_T(C_j)) \cup \{b_\ell\}$ contains exactly one cycle: this cycle includes b_ℓ and no other bridge. Removing the edges of the cycle in C_j gives the desired path P_ℓ. See Figure 2 for an illustration. We next discuss some properties of the Y_j's.

Lemma 8. *Graph Y_j is a spanning tree of $R(C_j)$.*

Proof. Removing the bridges splits T into trees $F_1, \dots, F_{k_j}$, each containing exactly one terminal from $R(C_j) = \{r_1, \dots, r_{k_j}\}$. After renumbering we have $r_i \in F_i$. Since T is a tree, the bridges connect the F_i without closing cycles. There is a bridge connecting F_i and F_ℓ if and only if there is an edge between r_i and r_ℓ in Y_j. Hence also Y_j is connected without containing cycles. ☐

Lemma 9. *Given Y_j and Y_ℓ, $j \neq \ell$, if both graphs contain an edge uv, then the weight $w(uv)$ of uv is the same in those two graphs.*

Proof. There is a unique simple path P_{uv} between u and v in T. Hence this must be the path associated to uv in the construction of *both* Y_j and Y_ℓ. For Y_j and Y_ℓ one has exactly one bridge on P_{uv}. Hence the heaviest edge on P_{uv} defines the weight $w(uv)$. ☐

The following lemma is the heart of our analysis.

Lemma 10. [Bridge Lemma] *Let T be a terminal spanning tree and $(x_j)_j$ be a k-DCR solution. Then $c(T) \leq \sum_j x_j \cdot br_T(C_j)$.*

Proof. We define a directed graph $G_R = (R, E_R)$ with nodeset R, by adding the trees Y_j directed towards $sink(C_j)$ for all components C_j, with a cost function w as defined above. Here, Lemma 9 gives that this definition is consistent. By construction, $w(uv)$ for $uv \in E_R$ is the cost of a bridge along the path between u and v in T. In particular, $w(uv)$ is the largest weight of any edge of the unique cycle in $T \cup \{uv\}$. As a consequence (see e.g. [11]), $w(F) \geq c(T)$ for every spanning tree F of G_R.

Next, for each component C_j we install capacity x_j in a cumulative way on the edges corre-

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

(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 a component C^t (w.r.t. x^t) and contract it.

(2) Compute a terminal spanning tree T^μ in the remaining instance

(3) Output $T^\mu \cup \bigcup_{t=1}^{\mu} C^t$.

sponding to Y_j in G_R and term the emerging capacity reservation $y : E_R \to \mathbb{Q}^+$. The directed tree Y_j supports at least the same flow as component C_j. Hence y is a feasible solution to BCR w.r.t. G_R. But in the absence of Steiner nodes, BCR is an integral polyhedron (see Theorem 2), hence its integrality gap is 1 and there is a spanning tree F in G_R that is not more costly than the fractional solution y. By definition $br_T(C_j) = w(Y_j)$, hence

$$\sum_j x_j br_T(C_j) = \sum_j x_j w(Y_j) = \sum_{e \in E_R} w(e) y_e \geq w(F) \geq c(T).$$

□

3 Improved Approximation for Steiner Tree

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 given in Appendix A.

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 inherites all the edges incident to C^t (in case of parallel edges, we only keep the cheapest one).

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 (and hence of cost zero), and add the constraint $\sum_i x_i = \Sigma$. 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.

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

Lemma 11. *One has* $E[c(T^\mu)] \leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^\mu \cdot opt_k^f$.

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 $t = 1, \ldots, \mu$ and let $C_1, \ldots, C_{h'}$ denote the components for k-DCR in this iteration. By definition, the reduction in the cost of T^t w.r.t. T^{t-1} is $br_{T^{t-1}}(C^t)$. Therefore:

$$E[c(T^t)] \leq c(T^{t-1}) - E[br_{T^{t-1}}(C^t)] = c(T^{t-1}) - \frac{1}{\Sigma}\sum_j x_j^t \cdot br_{T^{t-1}}(C_j) \overset{\text{Lemma 10}}{\leq} \left(1 - \frac{1}{\Sigma}\right) \cdot c(T^{t-1}).$$

Recall that initially $c(T_0) \le 2 \cdot opt_k^f$ by Lemma 7. The claim follows by repeatedly applying the above bound. □

In the statement of the Bridge Lemma 10 the tree T is a terminal spanning tree, that is, the lemma applies for trees that do not contain any Steiner node. For arbitrary Steiner trees T we can show a slightly weaker result.

For a given component Z, we denote as $Loss(Z)$ the minimum-cost subforest of Z with the property that there is a path between each Steiner node in Z and some terminal in $R(Z)$. We let $loss(Z) = c(Loss(Z))$. In the terminology from Section 2, $Loss(Z)$ is the complement of the set of bridges of the subtree Z after contracting $R(Z)$.

Lemma 12. *[28]* $loss(Z) \le \frac{1}{2}c(Z)$.

Proof. Turn Z into a binary tree whose leaves are the terminals by adding dummy Steiner nodes and cost 0 edges. For each Steiner node, choose the less expensive edge out of the edges going to its 2 children. These edges have total cost at most $\frac{1}{2}c(Z)$ and each Steiner node is connected to a terminal using chosen edges. The claim follows. □

Lemma 13. *Let T and T' be the optimal Steiner trees before and after contracting a sampled component. Then $E[c(T')] \le \left(1 - \frac{1}{2\Sigma}\right) \cdot c(T)$.*

Proof. We denote the components of T as $Z_1, \ldots, Z_q$. Let B_i denote the bridges of Z_i w.r.t. $R(Z_i)$ (i.e. B_i is the complement of $Loss(Z_i)$).

We construct a spanning tree Y_i on $R(Z_i)$ like in Section 2 with the properties that (1) for every edge $uv \in Y_i$ ($u, v \in R(Z_i)$) one has exactly one edge $b_{uv} \in B_i$ on the u-v path P_{uv} in Z_i, (2) $w(e) = c(b_e)$ for all $e \in Y_i$ and (3) $b_e \ne b_{e'}$ for $e \ne e'$. Hence $w(Y_i) \ge \frac{1}{2}c(Z_i)$ by Lemma 12.

We obtain a terminal spanning tree $Y := \bigcup_{i=1}^q Y_i$ of cost $w(Y) \ge \frac{1}{2}c(T)$. Consider an edge $uv \in Br_Y(C_j)$ which can be removed from Y after contracting C_j. Then b_{uv} could also be removed from T after buying C_j. In other words $br_T(C_j) \ge br_Y(C_j)$ for every component C_j. If we denote the randomly chosen component by C and the corresponding fractional solution by $(x_j)_j$ one has

$$E[br_T(C)] \ge E[br_Y(C)] = \frac{1}{\Sigma}\sum_j x_j br_Y(C_j) \overset{\text{Lemma 10}}{\ge} \frac{1}{\Sigma}w(Y) \ge \frac{1}{2\Sigma}c(T).$$

The claim follows by the same argument as in Lemma 11. □

Corollary 14. *For every $t = 1, \ldots, \mu$, $E[c(C^t)] \le \frac{1}{\Sigma}\left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot opt_k$.*

Proof. Let $C_1, \ldots, C_{h'}$ be the components in k-DCR at the beginning of step t in the algorithm. By $opt_k^{f,t} := \sum_j x_j^t \cdot c(C_j)$ and opt_k^t we denote the optimal fractional and integral solution at the beginning of iteration t, respectively. Iteratively applying Lemma 13 yields $E[opt_k^t] \le \left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot opt_k$. Then

$$E[c(C^t)] \le \frac{1}{\Sigma}E\left[\sum_j x_j^t \cdot c(C_j)\right] = \frac{1}{\Sigma}E[opt_k^{f,t}] \le \frac{1}{\Sigma}E[opt_k^t] \le \frac{1}{\Sigma}\left(1 - \frac{1}{2\Sigma}\right)^{t-1} \cdot opt_k.$$

□

We can conclude

Theorem 15. *For any $k = O(1)$, there is a polynomial-time expected $3/2$-approximation algorithm for k-restricted Steiner tree.*

Proof. Consider the algorithm above with $\mu = \delta\Sigma$ and $\delta = \ln(4) \approx 1.38$. The cost of the computed solution is $c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)$. The expected approximation ratio satisfies

$$E\left[\frac{c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)}{opt_k}\right] \leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^\mu + \frac{1}{\Sigma}\sum_{t=1}^{\mu}\left(1 - \frac{1}{2\Sigma}\right)^{t-1}$$
$$= 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^{\delta\cdot\Sigma} + 2 - 2 \cdot \left(1 - \frac{1}{2\Sigma}\right)^{\delta\cdot\Sigma} \leq 2e^{-\delta} + 2 - 2 \cdot e^{-\delta/2} = \frac{3}{2}$$

□

Theorem 15 immediately implies an expected $(3/2 + \varepsilon)$-approximation algorithm for the Steiner tree problem. In Appendix A, we give a refined analysis which leads to a $\ln(4) + \varepsilon < 1.39$ approximation.

We remark that the above arguments do not imply the existence of deterministic approximation algorithms with the same approximation factor. Interestingly, despite the fact that our analysis is based on a linear relaxation of the problem, it also does not imply a $3/2$ bound on the integrality gap of the studied LP. We believe that these two issues are closely related. One way to positively resolve both of them could be to prove a fractional version of Lemma 13. Namely, it would suffice to show that not only the integral optimum, but also the fractional optimum solution is getting cheaper in expectation after buying a sampled component.

4 Integrality Gap and Prize-Collecting Steiner Tree

In this section we bound the integrality gap of k-DCR and show how to use it to obtain an improved approximation algorithm for the prize-collecting Steiner tree problem. An easy adaption of the analysis of the argument from the previous section already gives the following.

Theorem 16. *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. Weakening the analysis of Corollary 14, we obtain that, for a proper choice of δ, the solution computed by the algorithm from previous section costs at most

$$E\Big[c(T^\mu) + \sum_{t=1}^{\mu} c(C^t)\Big] \leq 2 \cdot \left(1 - \frac{1}{\Sigma}\right)^\mu opt_k^f + \frac{1}{\Sigma}\sum_{t=1}^{\mu} opt_k^f \leq (2e^{-\delta} + \delta)opt_k^f \overset{\delta=\ln(2)}{\leq} (1 + \ln(2))opt_k^f.$$

□

In order to achieve the better 1.55 bound claimed in Theorem 4, we prove that another algorithm, the algorithm of Robins and Zelikovsky [34], produces solutions bounded with respect to fractional solutions to k-DCR. Our alternative analysis of this algorithm is, to some extent, inspired by an analogous argument of Charikar and Guha [8] in the context of the facility location problem. Our argument is essentially a combination of the analysis in [34] with our new Bridge Lemma 10. For this reason, the proof of Theorem 4 is given in Appendix C.

4.1 Prize-Collecting Steiner Tree

Consider the following linear relaxation for prize-collecting Steiner tree:

$$\min \sum_j c(C_j)x_j + \sum_{i\in R} p(i) z_i \qquad (k\text{-pDCR})$$

$$\begin{aligned} \sum_{C_j\in\delta'(S)} x_j + z_i \quad &\geq \quad 1 \quad \forall S\subseteq V\setminus\{r\} : i\in S\cap R \\ x_j, z_i \quad &\geq \quad 0 \quad \forall j=1,\dots,h,\ \forall i\in R. \end{aligned}$$

Let *p-opt*$_k^f$ be the value of the optimum solution (x^*, z^*) of k-pDCR, and let $X^* := \sum_j c(C_j)x_j^*$ and $Z^* := \sum_{i\in R} p(i) z_i^*$ denote the contribution given to the optimal solution by components and prizes, respectively. Observe that (x^*, z^*) can be computed in polynomial-time, with an approach analogous to Lemma 6. We also let *p-opt* be the cost of the optimal solution to the problem.

Following an observation in [9], we can use an easy adaptation of Theorem 2.6 in [3] to get the following (see Appendix B for more details).

Lemma 17. *There is a polynomial-time algorithm which computes a solution to the prize-collecting Steiner forest problem of cost at most* $2X^* + Z^*$.

The following lemma applies the techniques from [25] to k-pDCR and exploits Theorem 4.

Lemma 18. *For any* $\beta\in(0,1)$*, there is a polynomial-time algorithm which computes a solution to the prize-collecting Steiner tree problem of cost at most* $\frac{-1.55\ln(1-\beta)}{\beta}X^* + \frac{1}{\beta}Z^*$.

Proof. Consider the following algorithm. First compute an optimum solution to k-pDCR. Then choose a value α uniformly at random in $[0,\beta]$. Now round all the z_i^* variables $\tilde{R}$ with $z_i^*\geq\alpha$ up to one and round the other z_i^*'s down to zero. Solve the residual problem, which is a standard Steiner tree problem, with the algorithm of Robins and Zelikovsky. By the same argument as in [25] in combination with Theorem 4, the expected cost of the computed solution is at most

$$E\Big[1.55\sum_j \frac{1}{1-\alpha} c(C_j)x_j^* + \sum_{i\in\tilde{R}} p(i)\Big] \leq E\Big[\frac{1.55}{1-\alpha}\Big]\sum_j c(C_j)x_j^* + \sum_{i\in R}\Pr[z_i^*\geq\alpha]\cdot p(i)$$
$$\leq \frac{-1.55\ln(1-\beta)}{\beta}X^* + \frac{1}{\beta}Z^*.$$

This procedure can be easily derandomized since the optimal choice of α must be attained at one of the values of the z_i^* (plus 0 and β), which are polynomially many. $\square$

Proof of Theorem 5. Let $\alpha := X^*/$*p-opt*$_k^f$, and let $\beta(\alpha)\in[0,1]$ be the value minimizing the approximation factor $\frac{-1.55\ln(1-\beta)}{\beta}\alpha + \frac{1}{\beta}(1-\alpha)$ of the algorithm from Lemma 18. Consider the algorithm which outputs the best solution among the solutions computed by the algorithms in Lemma 17 and Lemma 18 with $\beta=\beta(\alpha)$. The cost of this solution is

$$\max_{\alpha\in[0,1]}\min\left\{2\alpha + (1-\alpha), \frac{-1.55\ln(1-\beta(\alpha))}{\beta(\alpha)}\alpha + \frac{1}{\beta(\alpha)}(1-\alpha)\right\} \textit{p-opt}_k^f < 1.94\,\textit{p-opt}_k^f.$$

An easy consequence of Theorem 1 is that *p-opt*$_k^f \leq (1+\frac{1}{\lfloor\log_2 k\rfloor})$*p-opt*$_n^f \leq (1+\frac{1}{\lfloor\log_2 k\rfloor})$*p-opt*. The claim follows for k large enough. $\square$

References

[1] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *FOCS*, 2009. To appear.

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

[3] J. Bang-Jensen, A. Frank, and B. Jackson. Preserving and increasing local edge-connectivity in mixed graphs. *SIAM Journal on Discrete Mathematics*, 8(2):155–178, 1995.

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

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

[6] 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.

[7] 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.

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

[9] C. Chekuri and F. B. Shepherd. Approximate integer decomposition for undirected network design problems. Private communication.

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

[11] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial optimization*. John Wiley & Sons Inc., New York, 1998.

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

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

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

[15] 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.

[16] 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.

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

[18] 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.

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

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

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

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

[23] 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.

[24] 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.

[25] M. T. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *SODA*, pages 631–640, 2006.

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

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

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

[29] 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).

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

[31] 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.

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

[33] 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.

[34] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005.

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

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

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

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

Appendix

A Refined Approximation for Steiner Tree

In this section we present a refined analysis leading to a $\ln(4)$-approximation for k-restricted Steiner tree. The algorithm (see Figure 4) is essentially the same as in Section 3, with the difference that we let the algorithm run until all the graph collapses into the root: at the end of the process the union of the contracted components gives the desired Steiner tree.

Recall that, by adding a dummy component, we can assume that $\Sigma := \sum_j x_j^t$ for all t.

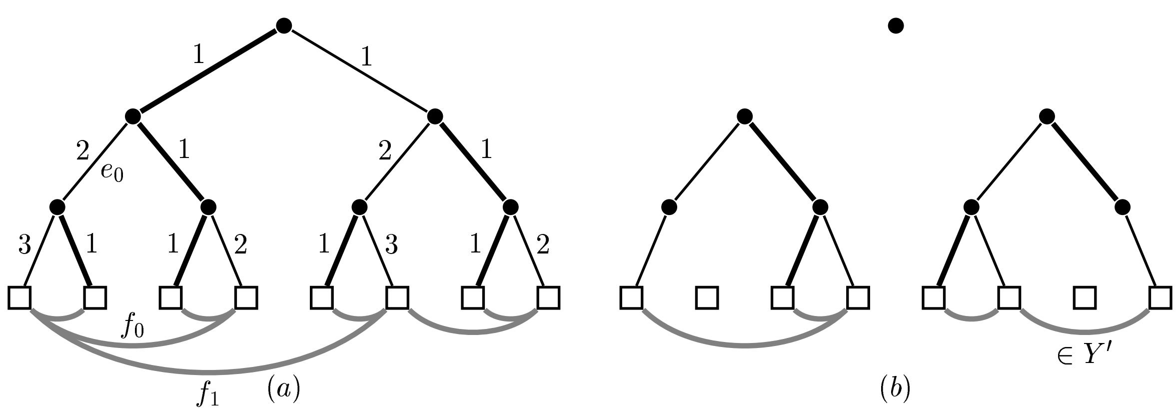
Figure 4 A $\ln(4)$-approximation algorithm for k-restricted Steiner tree.

(1) For $t = 1, 2, \ldots$

  (2) Compute an optimal fractional solution x^t to k-DCR (w.r.t. the current instance).
  (3) Sample a component C^t (w.r.t. x^t) and contract it.
  (4) If the instance consists only of the root, return $\bigcup_{i=1}^{t} C^i$.

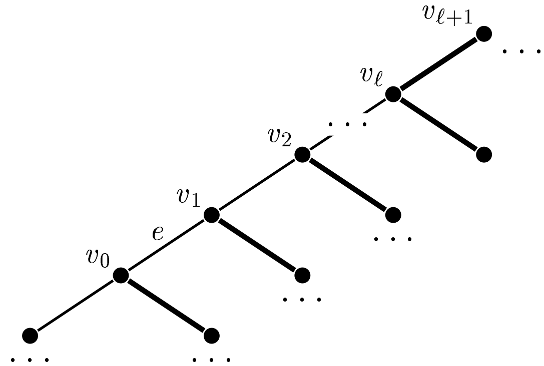
Since the analysis is rather involved, we first outline the basic ideas. Lemma 13 bounds the cost of the optimal k-restricted Steiner tree at each iteration. This is done by (ideally) removing from the original optimal Steiner tree T^* the set of bridges induced by the sampled components. However, differently from the case of terminal spanning trees, the presence of Steiner nodes enlarges the set of edges which can be potentially removed at each sampling step: this is because we can remove a Steiner node v (and the edges incident to v) from the Steiner tree if the presence of v is not needed to connect the terminals.

Figure 5 Tree Z_i in (a). Gates are drawn bold. Y_i is drawn gray. Edges e in Z_i are labeled with $|W(e)|$. For example $W(e_0) = \{f_0, f_1\}$. In (b), not marked edges in Y_i and Z_i are drawn in gray and black, respectively. Both edge sets support the same connectivity on $R(Z_i)$.



In order to capture this phenomenon, we define a random terminal spanning tree Y, that we call *witness tree*. Each edge e of T^* is associated to a set $W(e)$ of edges in Y. As we will see, the expected cardinality of $W(e)$ is small (Lemma 19). Each time we sample a component $C^t = C_j$, we mark a set B of edges of Y. Set B is sampled from a collection of sets $\mathcal{B}_Y(C_j)$ according to a proper

Figure 6 Illustration for $|W(e)| \leq \ell + 1$. Bridge edges are drawn bold



probability distribution w_{jB}. The collection $\mathcal{B}_Y(C_j)$ is given by all the maximal subsets (of at most $k-1$ edges) whose removal does not disconnect the graph $Y \cup C_j$. Note that each $B \in \mathcal{B}_Y(C_j)$ defines a potential set of bridges, and the set $Br_Y(C_j)$ of bridges considered in previous sections is simply the most expensive set in $\mathcal{B}_Y(C_j)$. In turn, when all the edges of $W(e)$ are marked, edge e is marked as well. The overall construction guarantees that the unmarked edges in T^* plus the sampled components induce a feasible Steiner tree, connecting all the terminals (Lemma 20). It is then sufficient to bound the probability that an edge $e \in T^*$ is still present at a given iteration: this immediately gives the desired bound on the expected cost of the optimal Steiner tree in that iteration (Lemma 24).

The witness tree Y is defined as follows. Let $Z_1, \ldots, Z_q$ denote the (undirected) full components of T^*. After introducing dummy nodes and cost 0 edges we may assume that each Z_i is a binary tree where the terminals $R(Z_i)$ form the leaves. For each Steiner node v, we sample uniformly at random one of the 2 edges between v and its children: sampled edges are called *gates*. Similar to Section 2, let Y_i be the graph on node set $R(Z_i)$ which has an edge uv if and only if there is exactly one gate on the u-v path P_{uv} in Z_i (see Figure 5.a). The witness tree is $Y := \bigcup_{i=1}^q Y_i$. It is not hard to see that Y_i is a terminal spanning tree on terminals $R(Z_i)$. Consequently, Y is a terminal spanning tree on all terminals R.

For each edge $e \in Z_i \subseteq T^*$, we define

$$W(e) := \{uv \in Y_i \mid e \in P_{uv}\}.$$

Note that $|W(e)| < k$ deterministically. Moreover, $|W(e)| = 1$ if e is a gate. Indeed, the expected cardinality of $W(e)$ is small also for the remaining edges.

Lemma 19. *For any edge $e \in T^*$ and any $q \in \mathbb{N}$ one has* $\Pr[|W(e)| > q] \leq (\frac{1}{2})^q$.

Proof. Let Z_i be the component of T^* that contains e. Consider the path $v_0, v_1, \ldots$ from e towards the root, and let $(v_\ell, v_{\ell+1})$ be the first gate on this path (if the root is reachable without crossing gates, let $(v_\ell, v_{\ell+1})$ be the gate incident to the root). For each node v_j, $j \geq 1$, there is at most one path P_{uv} with $uv \in Y_i$ that contains e. (See Figure 6). Hence $|W(e)| \leq \ell + 1$. The claim follows since $\Pr[\ell + 1 > q] \leq (\frac{1}{2})^q$. ☐

Recall that we mark $e \in T^*$ only if all the edges in $W(e)$ are marked. Let T^t be all edges from T^* that are unmarked at the beginning of iteration t. Let moreover $X(f) \in \mathbb{N}$ be the random variable denoting the iteration when $f \in Y$ is marked. The next lemma shows that unmarked edges plus sampled components induce a feasible Steiner tree. (See also Figure 5.b).

Lemma 20. *$T^t \cup \bigcup_{t'=1}^{t-1} C^{t'}$ spans R.*

Proof. Let $Y' := \{f \in Y \mid X(f) \geq t\}$ be the set of not yet marked edges in Y. Then, by definition of bridges, $Y' \cup \bigcup_{t'=1}^{t-1} C^{t'-1}$ spans R. Consider any edge $uv \in Y'$ (say $uv \in Y_i$). Then $uv \in W(e)$ for all $e \in P_{uv} \subseteq Z_i$. Hence no edge on P_{uv} is marked. Therefore u and v are also connected in T^t. The claim follows. □

It remains to define the probability distribution w_{jB}, $\sum_{B \in \mathcal{B}_Y(C_j)} w_{jB} = 1$, with which a set $B \in \mathcal{B}_Y(C_j)$ is marked when C_j is sampled. Let x be the optimal fractional solution in the considered iteration. For our purposes, it is sufficient to consider a (*balanced*) distribution satisfying

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

The next lemma shows that such a distribution always exists.

Lemma 21. *For every solution x to k-DCR, there exists a balanced distribution with respect to x.*

Proof. Suppose by contradiction that such a distribution does not exist. Then the following system of linear inequalities has no solution[3]

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

Farkas' Lemma[4] yields that there is a vector $(y, c) \geq \mathbf{0}$ with

$$(a)\ y_j \geq \sum_{f \in B} c_f x_j \quad \forall (B,j) : B \in \mathcal{B}_Y(C_j) \quad \text{and} \quad (b)\ \sum_j y_j < \sum_{f \in Y} c_f = c(Y).$$

In particular,

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

Then

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

which contradicts the Bridge Lemma 10. □

Balanced distributions have the nice property that each edge in Y is marked with probability at least $1/\Sigma$.

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

Lemma 22. *Each edge $f \in Y$ is marked in each iteration with probability at least $1/\Sigma$, given that it was not marked before.*

Proof. Consider any iteration t, let $Y^t \subseteq Y$ be the set of not yet marked edges ($f \in Y^t$) and let $\mathcal{B}_{Y^t}(C_j)$ be defined w.r.t. the graph G after contraction of $C_1, \ldots, C_{t-1}$. By w^t we denote the balanced distribution according to x^t (see Lemma 21). Then

$$\sum_{(B,j):B\in\mathcal{B}_{Y^t}(C_j),f\in B} \Pr[C_j \text{ is sampled and } B \text{ is chosen}] = \frac{1}{\Sigma} \sum_{(B,j):B\in\mathcal{B}_{Y^t}(C_j),f\in B} x_j^t \cdot w_{jB}^t \overset{w^t \text{ balanced}}{\geq} \frac{1}{\Sigma}.$$

□

We next bound, for a given edge $e \in T^*$, the probability that all the edges in $W(e)$ are marked at a given iteration t (and hence e is marked as well in that iteration). Recall that $X(f)$ is the iteration when $f \in Y$ is marked. We let $X(W) = \max\{X(f) \mid f \in W\}$ denote the iteration when all edges $W \subseteq Y$ are marked.

Lemma 23. *For any iteration $t \in \mathbb{N}$ and any set of edges $W \subseteq Y$*

$$\Pr[X(W) \leq t] \geq \left(1 - e^{-(t-|W|)/\Sigma}\right)^{|W|}.$$

Proof. By Lemma 22, each $f \in W$ is marked with probability at least $1/\Sigma$ in each iteration, given that it was not marked before. Then $\Pr[X(f) \leq t] \geq 1 - (1 - \frac{1}{\Sigma})^t$.

Unfortunately for several edges f, f' the variables $X(f)$ and $X(f')$ are in general dependent. Hence we define a new random variable $X'(f)$ that does not count the previous iterations in which edges in W are marked. In other words one iteration of X' corresponds to one or more iterations of X. In any case $X(W) - |W| \leq X'(W) \leq X(W)$. If we denote $W = \{f_1, \ldots, f_{|W|}\}$, then $\Pr[X'(f_i) > t' \mid X'(\{f_1, \ldots, f_{i-1}\}) \leq t'] \leq (1 - \frac{1}{\Sigma})^{t'}$. Hence

$$\begin{aligned}
\Pr[X(W) \leq t] &\geq \Pr[X'(W) \leq t - |W|] \\
&= \prod_{i=1}^{|W|} \Pr[X'(f_i) \leq t - |W| \mid X'(\{f_1, \ldots, f_{i-1}\}) \leq t - |W|] \\
&\geq \left(1 - \left(1 - \frac{1}{\Sigma}\right)^{t-|W|}\right)^{|W|} \geq \left(1 - e^{-(t-|W|)/\Sigma}\right)^{|W|}.
\end{aligned}$$

□

We now have all the ingredients to bound the expected cost opt_k^t of the optimum Steiner tree at the beginning of a given iteration t.

Lemma 24. $E[opt_k^t] \leq \frac{2}{e^{(t-k)/\Sigma}+1} opt_k$.

Proof. Let $e \in T^*$. Then $e \in T^t$ only if $X(W(e)) \geq t$. Applying Lemma 19 with $W := W(e)$ yields

$$\begin{aligned}\Pr[X(W(e)) \geq t] &= \Big(1 - \sum_{q \geq 1} \Pr[|W(e)| = q] \cdot \Pr[X(W(e)) < t \mid |W(e)| = q]\Big) \\ &\overset{\text{Lemma 19}}{\leq} 1 - \sum_{q \geq 1} \Pr[|W(e)| = q] \cdot (1 - e^{-(t-1-q)/\Sigma})^q \\ &\overset{|W(e)|<k}{\leq} 1 - \sum_{q \geq 1} \left(\frac{1}{2}\right)^q \cdot (1 - e^{-(t-k)/\Sigma})^q = \frac{2}{e^{(t-k)/\Sigma} + 1}.\end{aligned}$$

By linearity of expectation

$$E[opt_k^t] \leq E[c(T^t)] \leq \frac{2}{e^{(t-k)/\Sigma} + 1} opt_k.$$

□

Eventually, we prove Theorem 3.

Proof of Theorem 3. Let apx_{Σ_0} be the cost of the solution which is produced by the algorithm for $\Sigma = \Sigma_0$. By a simple coupling argument $E[apx_{\Sigma_0}]$ is independent from Σ_0 (as long as Σ_0 is at least the number h of components). Hence $E[apx_\Sigma] = \lim_{\Sigma_0 \to \infty} E[apx_{\Sigma_0}]$. The expected ratio between the cost of the solution produced by that algorithm and the cost opt_k of the optimal k-restricted Steiner tree satisfies

$$\begin{aligned}E\left[\frac{\sum_{t \geq 1} c(C^t)}{opt_k}\right] &= \frac{\sum_{t \geq 1} E[\frac{1}{\Sigma} opt_k^{f,t}]}{opt_k} \leq \frac{1}{\Sigma} \sum_{t \geq 1} E\left[\frac{opt_k^t}{opt_k}\right] \\ &\overset{\text{Lemma 24}}{\leq} \frac{1}{\Sigma} \sum_{t \geq 1} \frac{2}{e^{(t-k)/\Sigma} + 1} \xrightarrow{\Sigma \to \infty} \int_0^\infty \frac{2}{e^\delta + 1} d\delta = \ln(4).\end{aligned}$$

The claim follows from Theorem 1. □

B Proof of Lemma 17

A standard theorem of Edmonds states that if a directed graph G has k edge disjoint paths from a root node z to any other node, then there exist k edge-disjoint out-arborescences of root z spanning G. Since then, in a series of papers generalizations of Edmonds theorem have been developed. In particular, we consider the following result in [3].

Theorem 25. *[3] Given a directed graph $G = (V, E)$, with rational edge capacities $y_e \in [0, 1]$, $e \in E$, such that, for any node v, $\sum_{uv \in E} y_{uv} = \sum_{vu \in E} y_{vu}$, and a root vertex $r \in V$. Let $r(u)$ be the flow from u to r supported by y. There is a strongly-polynomial-time algorithm which computes a family of arborescences $A_1, \ldots, A_q$ rooted at r, $q = poly(|V|)$, and coefficients $\alpha_1, \ldots, \alpha_q \geq 0$, with $\sum_\ell \alpha_\ell = 1$, such that: (1) $\sum_{A_\ell \ni e} \alpha_\ell \leq y_e$ and (2) $\sum_{A_\ell \ni v} \alpha_\ell \geq \min\{1, r(v)\}$.*

As observed in [9], one can combine the theorem above with the standard LP for prize-collecting Steiner tree to compute a solution to the problem of cost at most $2X^* + Z^*$, where X^* and Z^* denote the connection and prize cost in the optimal fractional solution. Essentially the same idea also works with respect to our modified LP k-pDCR for the same problem.

Proof of Lemma 17. Consider the following algorithm. We first compute an optimum solution (x^*, z^*) to k-pDCR, and replace each undirected edge e in the original graph with two oppositely directed edges e' and e'', each one with capacity $y_{e'} = y_{e''} = \min\{1, \sum_{C_j \ni e} x_j^*\}$. Observe that y is a capacity reservation which supports a flow of $r(i) := 1 - z_i^*$ for each $i \in R$, and the total cost of this reservation is $2X^*$. Let $\{A_\ell, \alpha_\ell\}_\ell$ be the corresponding arborescences and coefficients provided by Theorem 25. Each A_ℓ induces a solution to the prize collecting Steiner tree problem of cost $\sum_{e \in A_\ell} w(e) + \sum_{i \in R, i \notin A_\ell} p(i)$: our algorithm returns the cheapest such solution.

Let us show that the solution returned is cheap enough via a probabilistic argument. Choose one arborescence A_ℓ randomly with probability α_ℓ. Observe that, by (2), the probability of not covering a node $i \in R$ is at most z_i^*. Therefore the total expected penalty cost is Z^*. Note also that (1) implies that the probability of using edge e is at most the capacity of this edge in the bi-directed solution. Thus the expected total connection cost is at most $2X^*$. By linearity of expectation, the expected total cost of A_ℓ is at most $2X^* + Z^*$. $\square$

C 1.55-Integrality Gap for k-DCR

In this section we prove that the integrality gap of k-DCR is at most $1 + \frac{\ln(3)}{2} < 1.55$. This is obtained by considering the algorithm R&Z by Robins and Zelikovsky, and adapting its analysis with the Bridge Lemma in mind.

Algorithm R&Z works as follows. It constructs a sequence $T^0, T^1, \ldots, T^\mu$ of terminal spanning trees, where T^0 is a minimum-cost terminal spanning tree in the original graph. At iteration t we are given a tree T^t and a cost function c_t on the edges of the tree (initially $c_0 \equiv c$). The algorithm considers any candidate component C with at least 2 and at most k terminals. Let $T^t[C]$ denotes the minimum spanning tree of the graph $T^t \cup C$, where the edges $e \in E(C)$ have weight 0 and the edges $f \in E(T^t)$ weight $c_t(f)$. The subset of edges in T^t but not in $T^t[C]$ are denoted by $Br_t(C)$ (bridges of T^t with respect to C), and $br_t(C) = c_t(Br_t(C))$. We let

$$gain_t(C) = br_t(C) - c(C) \qquad \text{and} \qquad sgain_t(C) = gain_t(C) + loss(C).$$

The algorithm selects the component C^{t+1} which maximizes $gain_t(C)/loss(C)$. If this quantity is non-positive, the algorithm halts. Otherwise, it considers the graph $T^t \cup C_{t+1}$, and contracts $Loss(C^{t+1})$. The tree T^{t+1} is a minimum-cost terminal spanning tree in the resulting graph. In case that parallel edges are created this way, the algorithm only keeps the cheapest of such edges. This way we obtain the cost function c_{t+1} on the edges of T^{t+1}.

Lemma 26. *[34] For $t = 1, 2, \ldots, \mu$, $c_t(T^t) = c_{t-1}(T^{t-1}) - sgain_{t-1}(C^t)$.*

Let Apx_k be the approximate solution computed by the algorithm , and $apx_k = c(Apx_k)$.

Lemma 27. *[34] For any $\ell \leq \mu$, $apx_k \leq \sum_{t=1}^{\ell} loss(C^t) + c_\ell(T^\ell)$.*

Recall that x_j is (the value of) the selection variable for component C_j in (an optimal solution to) k-DCR. Let $loss_k^f := \sum_j x_j\, loss(C_j)$.

Corollary 28. $loss_k^f \leq \frac{1}{2} opt_k^f$.

Proof. It follows immediately from Lemma 12 applied to the components C_j of Opt_k^f . □

Corollary 29. $c_\mu(T^\mu) \leq opt_k^f$.

Proof. Using the fact that $gain_\mu(C) = br_\mu(C) - c(C) \leq 0$ for any component C,

$$c_\mu(T^\mu) \overset{Lem.10}{\leq} \sum_j x_j br_\mu(C_j) \leq \sum_j x_j c(C_j) = opt_k^f.$$

□

By Corollary 29, and since $c_t(T^t)$ is a non-increasing function of t, there must be a value of $\ell \leq \mu$ such that:

$$c_{\ell-1}(T^{\ell-1}) > opt_k^f \geq c_\ell(T^\ell). \tag{1}$$

In the following we will bound $\sum_{t=1}^{\ell} loss(C^t) + c_\ell(T^\ell)$. By Lemma 27, this will give a bound on apx_k. Let

$$gain_t^f := c_t(T^t) - opt_k^f \qquad \text{and} \qquad sgain_t^f := gain_t^f + loss_k^f.$$

Lemma 30. *For* $t = 1, 2, \ldots, \mu$, $\frac{sgain_{t-1}(C^t)}{loss(C^t)} \geq \frac{sgain_{t-1}^f}{loss_k^f}$.

Proof. We first note that

$$\begin{aligned}\frac{gain_{t-1}^f}{loss_k^f} &= \frac{c_{t-1}(T^{t-1}) - \sum_j x_j c(C_j)}{\sum_j x_j loss(C_j)} \overset{\text{Lemma 10}}{\leq} \frac{\sum_j x_j (br_{t-1}(C_j) - c(C_j))}{\sum_j x_j loss(C_j)} \\ &= \frac{\sum_j x_j gain_{t-1}(C_j)}{\sum_j x_j loss(C_j)} \leq \max_j \left\{ \frac{gain_{t-1}(C_j)}{loss(C_j)} \right\} \leq \frac{gain_{t-1}(C^t)}{loss(C^t)},\end{aligned}$$

where in the last inequality we used the fact that C^t maximizes $gain_{t-1}(C)/loss(C)$ over all the k-restricted components C. It follows that

$$\frac{sgain_{t-1}(C^t)}{loss(C^t)} = 1 + \frac{gain_{t-1}(C^t)}{loss(C^t)} \geq 1 + \frac{gain_{t-1}^f}{loss_k^f} = \frac{sgain_{t-1}^f}{loss_k^f}.$$

□

We need some more notation. Let $sgain_{\ell-1}(C^\ell) = sgain^1 + sgain^2$ such that

$$sgain^1 = c_{\ell-1}(T^{\ell-1}) - opt_k^f \overset{(1)}{>} 0. \tag{2}$$

We also let $loss(C^\ell) = loss^1 + loss^2$ such that

$$\frac{sgain_{\ell-1}(C^\ell)}{loss(C^\ell)} = \frac{sgain^1}{loss^1} = \frac{sgain^2}{loss^2}. \tag{3}$$

Eventually, we define

$$
\begin{aligned}
sgain_\ell^{f1} &:= sgain_{\ell-1}^f - sgain^1 \\
&\overset{(2)}{=} c_{\ell-1}(T^{\ell-1}) - opt_k^f + loss_k^f - (c_{\ell-1}(T^{\ell-1}) - opt_k^f) = loss_k^f.
\end{aligned} \tag{4}
$$

Lemma 31. $\sum_{t=1}^{\ell-1} loss(C^t) + loss^1 \leq loss_k^f \ln\left(\frac{sgain_0^f}{sgain_\ell^{f1}}\right)$.

Proof. For every $t = 1, 2, \ldots, \ell - 1$,

$$
sgain_t^f = sgain_{t-1}^f - sgain_{t-1}(C^t) \overset{\text{Lemma 30}}{\leq} sgain_{t-1}^f \left(1 - \frac{loss(C^t)}{loss_k^f}\right).
$$

Furthermore

$$
\frac{sgain_{\ell-1}^f}{loss_k^f} \overset{\text{Lemma 30}}{\leq} \frac{sgain_{\ell-1}(C^\ell)}{loss(C^\ell)} \overset{(3)}{=} \frac{sgain^1}{loss^1},
$$

from which

$$
sgain_\ell^{f1} = sgain_{\ell-1}^f - sgain^1 \leq sgain_{\ell-1}^f \left(1 - \frac{loss^1}{loss_k^f}\right).
$$

Then

$$
\frac{sgain_\ell^{f1}}{sgain_0^f} \leq \left(1 - \frac{loss^1}{loss_k^f}\right) \prod_{t=1}^{\ell-1} \left(1 - \frac{loss(C^t)}{loss_k^f}\right).
$$

Taking the logarithm of both sides and recalling that $x \geq \ln(1 + x)$,

$$
\ln\left(\frac{sgain_0^f}{sgain_\ell^{f1}}\right) \geq \frac{1}{loss_k^f} \left(\sum_{t=1}^{\ell-1} loss(C^t) + loss^1\right).
$$

□

We now have all the ingredients to bound the approximation factor of the algorithm with respect to opt_k^f. Let $mst = c(T^0) = c_0(T^0)$. The following theorem and corollary are straightforward adaptations of analogous results in [34].

Theorem 32. $apx_k \leq opt_k^f + loss_k^f \ln\left(\frac{mst - opt_k^f + loss_k^f}{loss_k^f}\right)$.

Proof. Since $sgain_{t-1}(C^t) \geq loss(C^t)$, it follows from (3) that

$$
sgain^2 \geq loss^2. \tag{5}
$$

Putting everything together we obtain

$$
\begin{aligned}
apx_k &\overset{\text{Lemma 27}}{\leq} \sum_{t=1}^{\ell} loss(C^t) + c_\ell(T^\ell) \\
&\overset{\text{Lemma 26}}{=} \sum_{i=t}^{\ell-1} loss(C^t) + loss(C^\ell) + c_{\ell-1}(T^{\ell-1}) - sgain_{\ell-1}(C^\ell) \\
&= \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + loss^2 + c_{\ell-1}(T^{\ell-1}) - sgain^1 - sgain^2 \\
&\overset{(5)}{\leq} \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + c_{\ell-1}(T^{\ell-1}) - sgain^1 \\
&\overset{(2)}{=} \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + opt_k^f \\
&\overset{\text{Lemma 31}}{\leq} opt_k^f + loss_k^f \ln\left(\frac{sgain_0^f}{sgain_\ell^{f1}}\right) \\
&\overset{(4)}{=} opt_k^f + loss_k^f \ln\left(\frac{mst - opt_k^f + loss_k^f}{loss_k^f}\right).
\end{aligned}
$$

□

We may now conclude with

Proof of Theorem 4. Recall, that from Lemma 7 we have $mst \leq 2opt_k^f$. By applying it to Theorem 32 we obtain

$$apx_k \leq opt_k^f + loss_k^f \ln\left(1 + \frac{2opt_k^f - opt_k^f}{loss_k^f}\right).$$

The right-hand side of the inequality above is an increasing function of $loss_k^f$. By Corollary 28, $loss_k^f \leq \frac{1}{2}opt_k^f$, which implies

$$apx_k \leq opt_k^f + \frac{1}{2}opt_k^f \ln\left(1 + 2\frac{2opt_k^f - opt_k^f}{opt_k^f}\right) = opt_k^f\left(1 + \frac{\ln(3)}{2}\right).$$

□