Steiner Tree Approximation via Iterative Randomized Rounding[1]

JAROSŁAW BYRKA
EPFL, Lausanne, Switzerland and University of Wrocław, Poland
and
FABRIZIO GRANDONI
University of Rome Tor Vergata, Roma, Italy
and
THOMAS ROTHVOß
EPFL, Lausanne, Switzerland
and
LAURA SANITÀ
EPFL, Lausanne, Switzerland

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 1.55 [Robins,Zelikovsky-'05]. All these algorithms are purely combinatorial. A long-standing open problem is whether there is an LP relaxation of Steiner tree with integrality gap smaller than 2 [Vazirani,Rajagopalan-'99].

In this paper we present an LP-based approximation algorithm for Steiner tree with an improved approximation factor. Our algorithm is based on a, seemingly novel, *iterative randomized rounding* technique. We consider an LP relaxation of the problem, which is based on the notion of directed components. We sample one component with probability proportional to the value of the associated variable in a fractional solution: the sampled component is contracted and the LP is updated consequently. We iterate this process until all terminals are connected. Our algorithm delivers a solution of cost at most $\ln(4) + \varepsilon < 1.39$ times the cost of an optimal Steiner tree. The algorithm can be derandomized using the method of limited independence.

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

Additional Key Words and Phrases: Approximation algorithms, linear programming relaxations, network design, 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

[1]A preliminary version of this paper appeared in STOC'10 [Byrka et al. 2010].
Emails: J. Byrka jby@ii.uni.wroc.pl, F. Grandoni grandoni@disp.uniroma2.it, T. Rothvoß thomas.rothvoss@epfl.ch, L. Sanità laura.sanita@epfl.ch.

tree is one of the classic 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., [Hwang et al. 1992]).

The Steiner tree problem appears already in the list of **NP**-hard problems in the book by Garey and Johnson [1979]. In fact, it is **NP**-hard to find solutions of cost less than $\frac{96}{95}$ times the optimal cost [Bern and Plassmann 1989; Chlebík and Chlebíková 2008]. 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 input graph by its metric closure[2]. A *terminal spanning tree* is a Steiner tree without Steiner nodes: such a tree always exists in the metric closure of the graph. It is well-known that a minimum-cost terminal spanning tree is a 2-approximation for the Steiner tree problem [Gilbert and Pollak 1968; Vazirani 2001].

A sequence of improved approximation algorithms appeared in the literature [Karpinski and Zelikovsky 1997; Prömel and Steger 2000; Zelikovsky 1993], culminating with the famous $1 + \frac{\ln(3)}{2} + \varepsilon < 1.55$ approximation algorithm by Robins and Zelikovsky [2005]. (Here $\varepsilon > 0$ is an arbitrarily small constant). All these improvements are based on the notion of *k-restricted Steiner tree*, which is defined as follows. A *component* is a tree whose leaves coincide with a subset of terminals. A k-restricted Steiner tree S is a collection of components, with at most k terminals each (*k-components*), whose union induces a Steiner tree. The cost of S is the total cost of its components, counting duplicated edges with their multiplicity (see [Borchers and Du 1997] for more details). The *k-Steiner ratio* $\rho_k \geq 1$ is the supremum of the ratio between the cost opt_k of the optimal k-restricted Steiner tree and the cost opt of the optimal (unrestricted) Steiner tree. The following result by Borchers and Du [1997] shows that, in order to have a good approximation, it is sufficient to consider k-restricted Steiner trees for a large enough, constant k.

THEOREM 1. *[Borchers and Du 1997] Let r and s be the non-negative integers satisfying $k = 2^r + s$ and $s < 2^r$. Then*

$$\rho_k = \frac{(r+1)2^r + s}{r2^r + s} \leq 1 + \frac{1}{\lfloor \log_2 k \rfloor}.$$

The mentioned approximation algorithms exploit the notion of k-component within a *local-search* framework. They start with a minimum-cost terminal spanning tree (which is 2-approximate), and iteratively improve it. At each iteration, they add to the current solution a k-component, chosen according to some greedy strategy, and remove redundant edges. The process is iterated until no further improvement is achievable. Different algorithms use different greedy criteria.

Despite the efforts of many researchers in the last 10 years, the above framework did not provide any further improvement after [Robins and Zelikovsky 2000; 2005]. This motivated our search for alternative methods. One standard approach is to

[2]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.

exploit a proper LP relaxation (see, e.g., [Goemans and Myung 1993] for a list of LP relaxations of Steiner tree). A natural formulation for the problem is the *undirected cut formulation* (see [Goemans and Williamson 1995; Vazirani 2001]). Here 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 LP relaxation, 2-approximation algorithms can be obtained either using primal-dual schemes [Goemans and Williamson 1995] or iterative rounding [Jain 1998]. 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 [Vazirani 2001]).

Another well-studied, but more promising, LP is the *bidirected cut relaxation* [Chakrabarty et al. 2008; Edmonds 1967; Rajagopalan and Vazirani 1999]. 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

$$\begin{aligned} \min \sum_{e \in E} c(e) z_e & & & \text{(BCR)} \\ s.t. \sum_{e \in \delta^+(U)} z_e &\geq 1, & \forall U \subseteq V \setminus \{r\} : U \cap R \neq \emptyset; \\ z_e &\geq 0, & \forall e \in E. \end{aligned}$$

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 [1967] showed that BCR is integral in the spanning tree case.

THEOREM 2. *[Edmonds 1967] For $R = V$, the polyhedron of BCR is integral.*

The best-known lower bound on the integrality gap of BCR was 8/7 [Könemann et al. 2007; Vazirani 2001]. The best-known upper bound is 2, though BCR is believed to have a smaller integrality gap than the undirected cut relaxation [Rajagopalan and Vazirani 1999]. Chakrabarty et al. [2008] 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 [Rajagopalan and Vazirani 1999].

Finding a better-than-2 LP relaxation of the Steiner tree problem is a long-standing open problem [Chakrabarty et al. 2008; Rajagopalan and Vazirani 1999]. We remark that good LP-bounds, besides potentially leading to better approximation algorithms for Steiner tree, might have a much wider impact. This is because Steiner tree appears as a building block in several other problems, and the best approximation algorithms for some of those problems are LP-based. Strong LPs are also important in the design of (practically) efficient and accurate heuristics.

1.1 Our Results and Techniques

The main result of this paper is as follows.

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

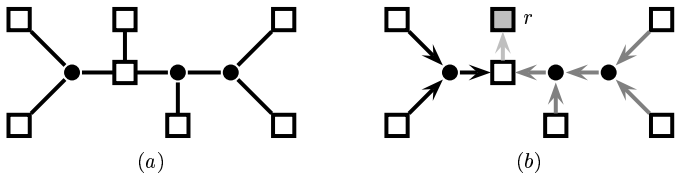

Fig. 1. (a) A Steiner tree S, where rectangles denote terminals and circles represent Steiner nodes. (b) Edges of S are directed towards a root r. The directed components of S are depicted with different colors.

This can be improved to $73/60+\varepsilon$ in the well-studied special case of *quasi-bipartite graphs* (where non-terminal nodes are pairwise not adjacent).

Our algorithm is based on the following *directed-component cut relaxation* for the Steiner tree problem (a similar relaxation is considered in [Polzin and Vahdati-Daneshmand 2003]). Consider any subset of terminals $R' \subseteq R$, and any $r' \in R'$. Let C be the minimum-cost Steiner tree on terminals R', with edges directed towards r' (*directed component*). For a given directed component C, we let $c(C)$ be its cost, and $sink(C)$ be its unique *sink* terminal. We call the remaining terminals $sources(C) := V(C) \cap R \setminus \{sink(C)\}$. The set of components obtained this way is denoted by $\mathbf{C}_n$. We say that a directed component $C \in \mathbf{C}_n$ *crosses* a set $U \subseteq R$ if C has at least one source in U and the sink outside. By $\delta^+_{\mathbf{C}_n}(U)$ we denote the set of directed components crossing U. Furthermore, we choose an arbitrary terminal r as a root. Our LP relaxation is then:

$$\begin{aligned} \min \sum_{C \in \mathbf{C}_n} c(C)x_C \qquad & \text{(DCR)} \\ s.t. \sum_{C \in \delta^+_{\mathbf{C}_n}(U)} x_C \;&\geq\; 1, \qquad && \forall U \subseteq R \setminus \{r\},\ U \neq \emptyset; \\ x_C \;&\geq\; 0, && \forall C \in \mathbf{C}_n. \end{aligned}$$

DCR is trivially a relaxation of the Steiner tree problem. In fact, one can direct the edges of the optimal Steiner tree S^* towards terminal r, and split the edge set of S^* at interior terminals. This yields a set of directed components $\mathbf{C} \subseteq \mathbf{C}_n$ (see Figure 1). Observe that any $C \in \mathbf{C}$ must be an optimal Steiner tree on terminals $R \cap V(C)$. Consequently, setting $x_C = 1$ for any $C \in \mathbf{C}$, and the remaining variables to zero, provides a feasible solution to DCR of cost $\sum_{C \in \mathbf{C}} c(C) = c(S^*) = opt$.

Unfortunately the cardinality of $\mathbf{C}_n$ is exponential. However, we will see that, for any constant $\varepsilon > 0$, one can compute a $(1+\varepsilon)$-approximate fractional solution to DCR in polynomial time. This is achieved by restricting $\mathbf{C}_n$ to the directed components $\mathbf{C}_k$ that contain at most a (big) constant number k of terminals (*directed k-components*).

We combine our LP with a (to the best of our knowledge) novel *iterative randomized rounding* technique. We solve the LP (approximately), sample one component C with probability proportional to its value x_C in the near-optimal fractional solution x, contract C into its sink node $sink(C)$, and reoptimize the LP. We iterate this process until only one terminal remains, i.e., until all terminals are connected

by the sampled components. A fairly simple analysis provides a $3/2 + \varepsilon$ bound on the approximation ratio. With a refined analysis, we improve this bound to $\ln(4) + \varepsilon$. Our algorithm can be derandomized.

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 Steiner tree in each iteration. To show this, we exploit a novel *Bridge Lemma*, relating the cost of terminal spanning trees with the cost of fractional solutions to DCR. The proof of the lemma is based on Theorem 2 [Edmonds 1967]. In our opinion, our analysis is simpler (or at least more intuitive) than the one in [Robins and Zelikovsky 2005].

As an easy consequence of our analysis, we obtain that the integrality gap of DCR is at most $1 + \ln(2) < 1.694$, hence answering to the mentioned open problem in [Chakrabarty et al. 2008; Rajagopalan and Vazirani 1999]. Combining our Bridge Lemma with the algorithm and analysis by Robins and Zelikovsky [2005], we obtain the following improved result.

THEOREM 4. *For any $\varepsilon > 0$, there is an algorithm for the Steiner tree problem which computes a solution of cost at most $1 + \frac{\ln(3)}{2} + \varepsilon$ times the cost of the optimal fractional solution to DCR. The running time of the algorithm is polynomial for constant ε.*

The above theorem immediately implies a $1 + \ln(3)/2 < 1.55$ upper bound on the integrality gap of DCR, by letting ε tend to zero (the running time is irrelevant with that respect). As mentioned before, integrality gap results of this type often provide new insights into 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.

We also show that the integrality gap of DCR and BCR are at least $8/7 > 1.142$ and $36/31 > 1.161$, respectively.

1.2 Related Work

A sign of importance of the Steiner tree problem 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* [Agrawal et al. 1995; Goemans and Williamson 1995], *prize-collecting Steiner tree* [Archer et al. 2009; Goemans and Williamson 1995], *virtual private network* [Eisenbrand and Grandoni 2005; Eisenbrand et al. 2007; Grandoni and Rothvoß 2010; Gupta et al. 2001; Rothvoß and Sanità 2009], *single-sink rent-or-buy* [Eisenbrand et al. 2008; Grandoni and Rothvoß 2010; Gupta et al. 2007; Jothi and Raghavachari 2009], *connected facility location* [Eisenbrand et al. 2008; Swamy and Kumar 2004], and *single-sink buy-at-bulk* [Grandoni and Italiano 2006; Gupta et al. 2007; Grandoni and Rothvoß 2010; Talwar 2002].

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 [1998] provides a 2-approximation for the *Steiner network* problem, and the primal-dual framework developed by Goemans and Williamson [1995] gives the same approximation factor for a large class of *constrained forest* problems.

Regarding the integrality gap of LP relaxations of 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 [1999] (see also [Rizzi 2003]) gave an upper bound of 3/2 on the gap. This was recently improved to 4/3 by Chakrabarty, Devanur and Vazirani [2008]. Still, for this class of graphs the lower bound of 8/7 holds [Könemann et al. 2007; Vazirani 2001]. Könemann, Pritchard and Tan [2007] 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. All the mentioned LPs can be solved in polynomial time, while we solve DCR only approximately: from a technical point of view, we indeed solve exactly a relaxation of the k-restricted Steiner tree problem.

Finally, we remark that under additional constraints, Steiner tree admits better approximations. In particular, a PTAS can be obtained by the technique of Arora [1998] if the nodes are points in a fixed-dimension Euclidean space, and using the algorithm of Borradaile, Kenyon-Mathieu and Klein [2007] for planar graphs.

1.3 Organization

The rest of this paper is organized as follows. In Section 2 we give some definitions and basic results. In Section 3 we show how to approximate DCR and prove our Bridge Lemma. In Section 4 we present a simple expected $(1.5+\varepsilon)$-approximation for the problem. This result is improved to $\ln(4)+\varepsilon$ in Section 5. The special case of quasi-bipartite graphs is considered in Section 5.1. We derandomize our algorithm in Section 6. Finally, in Section 7 we discuss the integrality gap of DCR, and compare DCR with BCR.

2. PRELIMINARIES

We use Opt to denote the optimal integral solution, and $opt = c(Opt)$. The cost of an optimal solution to DCR (for the input instance) is termed opt_f. We will consider algorithms consisting of a sequence of iterations, each one considering different subproblems. We will use an apex to denote the considered iteration t. For example, opt_f^t denotes the cost of an optimal fractional solution at the beginning of iteration t.

For a given (directed or undirected) component C, $R(C) := R \cap V(C)$ is the set of its terminals. Recall that DCR has an exponential number of variables and constraints. For this reason, our algorithms will consider approximate solutions to DCR with a polynomial-size support. Therefore, it is notationally convenient to represent a solution to DCR as a pair $(x, \mathbf{C})$, where $\mathbf{C} \subseteq \mathbf{C}_n$ is a subset of directed components and $x = \{x_C\}_{C \in \mathbf{C}}$ denotes the values that are associated to each such component. (Other variables are assumed to have value zero).

Let T be a minimum-cost terminal spanning tree. It is a well-known fact that $c(T) \leq 2 \cdot opt$ (see e.g. Theorem 3.3 in [Vazirani 2001]). Extending the standard

proof, this bound also holds w.r.t. our LP relaxation.

LEMMA 5. *One has* $c(T) \leq 2 \cdot opt_f$.

PROOF. Let $(x, \mathbf{C})$ be an optimal fractional solution to DCR. For each component $C \in \mathbf{C}$, obtain an undirected TSP tour on $R(C)$ of cost at most $2c(C)$, remove one edge of the tour, and direct the remaining edges towards $sink(C)$. Install capacity x_C cumulatively on the directed edges of the resulting arborescence. This induces a fractional solution to DCR of cost at most $2 \cdot opt_f$, with the property that only components with 2 terminals and without Steiner nodes are used. This also provides a feasible fractional solution to BCR of the same cost. Since BCR without Steiner nodes is integral by Theorem 2, the claim follows. □

Let R' be a subset of k terminals. Consider a given Steiner tree S, with edge weights c, containing the terminals R'. The weight function c associated to S, if not specified, will be clear from the context. Let us collapse the terminals R' into one node, and call G' the resulting (possibly, multi-)graph. Let $S' \subseteq S$ be a minimum spanning tree of G'. 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 $k-1$. We call the latter edges the *bridges* of S w.r.t. R', and denote them by $Br_S(R')$[3]. 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. In other terms,

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

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.

In the analysis, it is often convenient to turn a given Steiner tree S into a rooted, possibly non-complete, binary tree as follows (see, e.g., [Karpinski and Zelikovsky 1997]). By adding dummy nodes and dummy edges of cost zero, we can assume that the leaves of S coincide with its terminals, and that internal (Steiner) nodes have degree exactly three. Then we split any edge by adding a dummy node v and a dummy edge of cost zero, and we root the tree at v. Note that the resulting rooted binary tree has height at most $|R|-1$. Given this reduction, it is easy to prove the following standard result.

LEMMA 6. *For any Steiner tree* S *on terminals* R, $br_S(R) \geq \frac{1}{2}c(S)$.

PROOF. Turn S into a rooted binary tree as described above. 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 $br_S(R) \geq c(B) \geq \frac{1}{2}c(S)$. □

Throughout this paper, we sometimes identify a subgraph G' with its set of edges $E(G')$.

[3]As usual, we break ties according to edge indexes.

3. A DIRECTED-COMPONENT CUT RELAXATION

In this section we show how to solve DCR approximately (Section 3.1), and prove our Bridge Lemma (Section 3.2).

3.1 Approximating DCR

We next show how to compute a $(1+\varepsilon)$-approximate solution to DCR, for any given constant $\varepsilon > 0$, in polynomial time. This is achieved in two steps. First of all, we introduce a relaxation k-DCR of the k-restricted Steiner tree problem. This relaxation can be solved exactly in polynomial time for any constant value of the parameter k (Lemma 8). Then we show that the optimal solutions to k-DCR and DCR are close for large-enough k (Lemma 7).

Let $\mathbf{C}_k \subseteq \mathbf{C}_n$ denote the set of directed components with at most k terminals, and let $\delta^+_{\mathbf{C}_k}(U) := \delta^+_{\mathbf{C}_n}(U) \cap \mathbf{C}_k$. By the same arguments as for the unrestricted case, the following is a relaxation of the k-restricted Steiner tree problem:

$$\begin{aligned} \min \sum_{C \in \mathbf{C}_k} c(C) x_C & & (k\text{-DCR}) \\ s.t. \sum_{C \in \delta^+_{\mathbf{C}_k}(U)} x_C &\geq 1, \ \forall U \subseteq R \setminus \{r\}, \ U \neq \emptyset; \\ x_C &\geq 0, & \forall C \in \mathbf{C}_k. \end{aligned}$$

Let $opt_{f,k}$ be the value of the optimal fractional solution to k-DCR. Trivially, $opt_{f,k} \geq opt_f$ since any feasible solution to k-DCR is also feasible for DCR. We can exploit the result by Borchers and Du [1997] to show that $opt_{f,k}$ is indeed close to opt_f for large k.

LEMMA 7. *One has $opt_{f,k} \leq \rho_k \cdot opt_f$.*

PROOF. Let $(x, \mathbf{C})$ be an optimal fractional solution for DCR. We show how to construct a solution $(x', \mathbf{C}')$ to k-DCR with the claimed property. For any component $C \in \mathbf{C}$, we can apply Theorem 1 to obtain a list of undirected components $C_1, \ldots, C_\ell$ such that: (a) $\bigcup_{i=1}^{\ell} C_i$ connects the terminals in C, (b) any C_i contains at most k terminals, and (c) $\sum_{i=1}^{\ell} c(C_i) \leq \rho_k \cdot c(C)$. Next, we direct the edges of all C_i's consistently towards $sink(C)$ and increase the value of x'_{C_i} by x_C for each C_i. The resulting solution $(x', \mathbf{C}')$ satisfies the claim. □

It remains to solve k-DCR for $k = O(1)$. For any fixed k, in polynomial time one can consider any subset $R' \subseteq R$ of at most k terminals, and compute an optimal Steiner tree Z on R'[4]. By considering each $r' \in R'$, and directing the edges of Z towards r', one obtains all the directed components on terminals R'. Consequently, $|\mathbf{C}_k| = O(kn^k)$ and the k-components can be listed in polynomial time.

LEMMA 8. *The optimal solution to k-DCR can be computed in polynomial time for any constant k.*

[4]We recall that, given k terminals, the dynamic-programming algorithm by Dreyfus and Wagner [1972] 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 [Mölle et al. 2006].

PROOF. We define a directed auxiliary graph $G' = (V', E')$, on node set $V' = R \cup \{v_C \mid C \in \mathbf{C}_k\}$. For every component C, insert edges (u, v_C) for any $u \in sources(C)$, and one edge $e_C = (v_C, sink(C))$. We observe that k-DCR is equivalent to a non-simultaneous multicommodity flow problem, where any terminal in R sends one unit of flow to the root and edges e_C have cost $c(C)$.

More precisely k-DCR is equivalent to the following compact LP:

$$\min \sum_{C \in \mathbf{C}_k} c(C)\, x_C$$

$$\begin{aligned} s.t. \sum_{e \in \delta^+(v)} f_s(e) - \sum_{e \in \delta^-(v)} f_s(e) &= \begin{cases} 1 & \text{if } v = s; \\ -1 & \text{if } v = r; \\ 0 & \text{if } v \in V \setminus \{r, s\}, \end{cases} && \forall s \in R \setminus \{r\}; \\ f_s(e_C) &\leq x_C, && \forall s \in R \setminus \{r\}, C \in \mathbf{C}_k; \\ f_s(e), x_C &\geq 0, && \forall s \in R \setminus \{r\}, e \in E', C \in \mathbf{C}_k. \end{aligned}$$

Here $f_s(e)$ denotes the flow that terminal s sends across edge e and the capacity on edge e_C is $x_C = \max_{s \in R \setminus \{r\}} f_s(e_C)$. An optimal solution of the latter LP can be computed in polynomial time, see e.g. [Khachiyan 1979; Grötschel et al. 1981][5]. □

Putting everything together, we obtain the desired approximate solution to DCR.

LEMMA 9. *For any fixed $\varepsilon > 0$, a $(1+\varepsilon)$-approximate solution $(x, \mathbf{C})$ to DCR can be computed in polynomial time.*

PROOF. It is sufficient to solve k-DCR for $k := 2^{\lceil 1/\varepsilon \rceil}$ with the algorithm from Lemma 8. Observe that $\rho_k \leq 1 + \varepsilon$ (see again Theorem 1). The claim follows from Lemma 7. □

3.2 The Bridge Lemma

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 DCR via the notion of bridges, and its proof is based on Edmonds' Theorem 2.

A key ingredient in the proof of our lemma is the construction of a proper weighted terminal spanning tree Y. Consider a Steiner tree S on terminals R. 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. Recall that $Br_S(R')$ is the set of bridges of S with respect to terminals R', and $br_S(R')$ denotes its cost.

LEMMA 10. *Let S be any Steiner tree on terminals R, and $w : R \times R \to \mathbb{Q}_+$ be the associated bridge weight function. For any subset $R' \subseteq R$ of terminals, there is a tree $Y \subseteq R' \times R'$ such that*

(a) *Y spans R'.*
(b) *$w(Y) = br_S(R')$.*
(c) *For any $\{u, v\} \in Y$, the u-v path in S contains exactly one edge from $Br_S(R')$.*

[5]Note that this LP can even be solved in strongly-polynomial time using the Frank-Tardos algorithm [Frank and Tardos 1987]

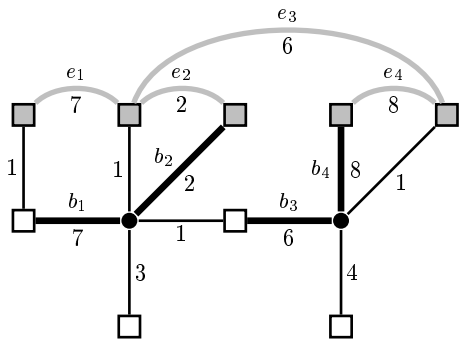

Fig. 2. Steiner tree S is drawn in black. Terminals of R' are gray shaded. Bold black edges indicate $Br_S(R') = \{b_1, \ldots, b_4\}$. The corresponding edges $e_1, \ldots, e_4$ of Y are drawn in gray and labeled with $w(e_i)$. Note that $w(e_i) = c(b_i)$. Observe also that b_3 is the unique bridge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

PROOF. Let $Br_S(R') = \{b_1, b_2, \ldots, b_{k-1}\}$ be the set of bridges. 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. Observe that Y contains k nodes and $k-1$ edges. Assume by contradiction that Y contains a cycle, say $e_1, e_2, \ldots, e_g$. Replace each $e_i = \{r_{i'}, r_{i''}\}$ with $F_{i'} \cup F_{i''} \cup \{b_i\}$: the resulting graph is a cyclic subgraph of S, a contradiction. Hence 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 claim follows. □

LEMMA 11. *[Bridge Lemma] Let T be a terminal spanning tree and $(x, \mathbf{C})$ be a feasible solution to DCR. Then*

$$c(T) \leq \sum_{C \in \mathbf{C}} x_C \cdot br_T(C).$$

PROOF. For every component $C \in \mathbf{C}$, we construct a spanning tree Y_C on $R(C)$ with weight $w(Y_C) = br_T(C)$ according to Lemma 10. Then we direct the edges of Y_C towards $sink(C)$. Let us install cumulatively capacity x_C on the (directed) edges of Y_C, for each $C \in \mathbf{C}$. This way we obtain a directed capacity reservation $y : R \times R \to \mathbb{Q}_+$, with $y(u,v) := \sum_{Y_C \ni (u,v)} x_C$. The directed tree Y_C supports at least the same flow as component C with respect to $R(C)$. 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 Edmond's 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 classic cycle rule for minimum spanning tree computation that $w(F) \geq c(T)$ (see, e.g., Theorem 6.2 in [Korte and Vygen 2002]).

(1) For $t = 1, 2, \ldots$
  (1a) Compute a $(1+\frac{\varepsilon}{2})$-approximate solution $(x^t, \mathbf{C}^t)$ to DCR (w.r.t. the current instance).
  (1b) Sample one component C^t, where $C^t = C$ with probability $x^t_C / \sum_{C' \in \mathbf{C}^t} x^t_{C'}$. Contract C^t into its sink.
  (1c) If a single terminal remains, return $\bigcup_{i=1}^t C^i$.

Fig. 3. A $(\ln(4)+\varepsilon)$-approximation algorithm for Steiner tree.

Altogether

$$\sum_{C \in \mathbf{C}} x_C br_T(C) = \sum_{C \in \mathbf{C}} x_C w(Y_C) = \sum_{e \in R \times R} w(e) y(e) \geq w(F) \geq c(T). \quad \square$$

4. ITERATIVE RANDOMIZED ROUNDING

In this section we present our approximation algorithm for 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 $3/2+\varepsilon$ approximation factor (which is already an improvement on the previous best 1.55 approximation). The more complex analysis leading to $\ln(4)+\varepsilon$ is postponed to Section 5.

The approximation algorithm for Steiner tree is described in Figure 3. In Step (1a) we use the algorithm from Lemma 9. Recall that the cardinality of $\mathbf{C}^t$ is upperbounded by a value M which, for any fixed $\varepsilon > 0$, is bounded by a polynomial in n. *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). We let Opt^t denote the optimal Steiner tree at the beginning of iteration t, and let opt^t be its cost. By opt^t_f we denote the cost of the optimal fractional solution at the beginning of iteration t.

Observe that $\sum_{C \in \mathbf{C}^t} x^t_C \leq M$, and this quantity might vary over the iterations t. In order to simplify the analysis, we add a dummy component $\bar{C}$ formed by the root only (hence of cost zero) to ensure that in fact

$$\sum_{C \in \mathbf{C}^t} x^t_C = M, \quad \forall t \geq 1.$$

Note that adding such dummy component corresponds to inserting idle iterations into the algorithm. But the expected cost of the produced solution remains the same.

The expected cost of the produced solution is:

$$\sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} \sum_{C \in \mathbf{C}^t} E\Big[\frac{x^t_C}{M} c(C)\Big] \leq \frac{1+\frac{\varepsilon}{2}}{M} \sum_{t \geq 1} E[opt^t_f] \leq \frac{1+\frac{\varepsilon}{2}}{M} \sum_{t \geq 1} E[opt^t] \quad (1)$$

Thus, in order to obtain a good approximation guarantee, it suffices to provide a good bound on $E[opt^t]$.

4.1 A first bound

A simple consequence of the Bridge Lemma is that the cost of the minimum terminal spanning tree decreases by a factor $(1-\frac{1}{M})$ per iteration in expectation. This implies an upper bound on opt^t_f via Lemma 5 (while later bounds will hold for opt^t only). The bound on opt^t_f implies the first non-trivial bounds on the approximation

guarantee of our algorithm (due to the fact that $opt_f^t \leq opt^t$) and on the integrality gap of our LP.

LEMMA 12. *One has* $E[opt_f^t] \leq \left(1 - \frac{1}{M}\right)^{t-1} \cdot 2opt_f$.

PROOF. Let T^t be the minimum-cost terminal spanning tree at the beginning of iteration t. By Lemma 5, $c(T^1) \leq 2opt_f$. For any iteration $t > 1$, the reduction in the cost of T^t w.r.t. T^{t-1} is at least $br_{T^{t-1}}(C^t)$. Therefore:

$$\begin{aligned} E[c(T^t)] &\leq c(T^{t-1}) - E[br_{T^{t-1}}(C^{t-1})] \\ &= c(T^{t-1}) - \frac{1}{M} \sum_{C \in \mathbf{C}^{t-1}} x_C^{t-1} \cdot br_{T^{t-1}}(C) \\ &\overset{\text{Bridge Lem 11}}{\leq} \left(1 - \frac{1}{M}\right) \cdot c(T^{t-1}). \end{aligned}$$

By induction

$$E[opt_f^t] \leq E[c(T^t)] \leq \left(1 - \frac{1}{M}\right)^{t-1} \cdot 2opt_f. \quad \square$$

Observe that the bound from Lemma 12 improves over the trivial bound $opt_f^t \leq opt_f$ for $t > M \cdot \ln(2)$. Nevertheless it suffices to prove the following result.

THEOREM 13. *For any fixed $\varepsilon > 0$, there is a randomized polynomial-time algorithm which computes a solution to the Steiner tree problem of expected cost at most* $(1 + \ln(2) + \varepsilon) \cdot opt_f$.

PROOF. Assume without loss of generality that $M \cdot \ln(2)$ is integral. Combining (1) with Lemma 12, the expected approximation factor is

$$\begin{aligned} E\left[\frac{\sum_{t \geq 1} c(C^t)}{opt_f}\right] &\leq \frac{1 + \varepsilon/2}{M} \sum_{t \geq 1} E\left[\frac{opt_f^t}{opt_f}\right] \\ &\leq \frac{1 + \varepsilon/2}{M} \sum_{t \geq 1} \min\left\{1, 2\left(1 - \frac{1}{M}\right)^{t-1}\right\} \\ &\leq \frac{1 + \varepsilon/2}{M} \left(M \cdot \ln(2) + \sum_{t \geq M \cdot \ln(2) + 1} 2\left(1 - \frac{1}{M}\right)^{t-1}\right) \\ &\leq \left(1 + \frac{\varepsilon}{2}\right) \left(\ln(2) + 2\left(1 - \frac{1}{M}\right)^{M \cdot \ln(2)}\right) \\ &\leq \left(1 + \frac{\varepsilon}{2}\right) \left(\ln(2) + 2e^{-\ln(2)}\right) \leq 1 + \ln(2) + \varepsilon. \end{aligned}$$

Above we used the equation $\sum_{t \geq t_0} x^t = \frac{x^{t_0}}{1-x}$ for $0 < x < 1$ and the inequality $(1 - 1/x)^x \leq 1/e$ for $x \geq 1$. $\square$

Observe that Theorem 13 implies that the integrality gap of DCR is at most $1+\ln(2)$. In Section 7 we will refine this bound on the gap to $1 + \ln(3)/2$.

4.2 A second bound

In order to further improve the approximation guarantee we show that, in each iteration, the cost opt^t of the optimal (integral) Steiner tree of the current instance decreases by a factor $(1 - \frac{1}{2M})$ in expectation. We remark that it is not known whether this bound holds also for opt_f^t. Also in this case the proof relies crucially on the Bridge Lemma.

LEMMA 14. *Let S be any Steiner tree and $(x, \mathbf{C})$ be a feasible solution to DCR. Sample a component $C \in \mathbf{C}$ such that $C = C'$ with probability $x_{C'}/M$. Then there is a subgraph $S' \subseteq S$ such that $S' \cup C$ spans R and*

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

PROOF. It suffices to prove that $E[br_S(C)] \geq \frac{1}{2M} c(S)$. Turn S into a rooted binary tree with the usual procedure. Then, for any Steiner node in S, choose the cheapest edge going to one of its children. The set $H \subseteq S$ of such selected edges has cost $c(H) \leq \frac{1}{2} c(S)$. Furthermore any Steiner node is connected to one terminal using edges of H. Consider the terminal spanning tree T that emerges from S by contracting H. By the Bridge Lemma 11,

$$E[br_T(C)] = \frac{1}{M} \sum_{C' \in \mathbf{C}} x_{C'} \cdot br_T(C') \geq \frac{1}{M} c(T).$$

Observe also that $Br_T(C)$ is a feasible set of bridges for S with respect to C, and thus $br_S(C) \geq br_T(C)$. Altogether:

$$E[br_S(C)] \geq E[br_T(C)] \geq \frac{1}{M} c(T) = \frac{1}{M}(c(S) - c(H)) \geq \frac{1}{2M} c(S). \quad \square$$

Iterating Lemma 14 yields the following corollary.

COROLLARY 15. *For every $t \geq 1$,*

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

We now have all the ingredients to show a $(3/2 + \varepsilon)$-approximation factor.

THEOREM 16. *For any $\varepsilon > 0$, there is a polynomial-time randomized approximation algorithm for Steiner tree with expected approximation ratio $3/2 + \varepsilon$.*

PROOF. Assume without loss of generality that $M \cdot \ln(4)$ is integral. Combining

(1) with Lemma 12 and Corollary 15, the expected approximation factor is:

$$
\begin{aligned}
E\left[\frac{\sum_{t\geq 1} c(C^t)}{opt}\right] &\leq \frac{1+\varepsilon/2}{M}\sum_{t\geq 1} E\left[\frac{opt^t}{opt}\right] \\
&\leq \frac{1+\varepsilon/2}{M}\sum_{t\geq 1}\min\left\{2\left(1-\frac{1}{M}\right)^{t-1},\ \left(1-\frac{1}{2M}\right)^{t-1}\right\} \\
&\leq \frac{1+\varepsilon/2}{M}\left(\sum_{t=1}^{M\cdot\ln(4)}\left(1-\frac{1}{2M}\right)^{t-1}+\sum_{t\geq M\cdot\ln(4)+1} 2\left(1-\frac{1}{M}\right)^{t-1}\right) \\
&= \left(1+\frac{\varepsilon}{2}\right)\cdot\left(2-2\cdot\left(1-\frac{1}{2M}\right)^{M\cdot\ln(4)}+2\left(1-\frac{1}{M}\right)^{M\cdot\ln(4)}\right) \\
&\leq \left(1+\frac{\varepsilon}{2}\right)\cdot\left(2-2\cdot e^{-\ln(4)/2}+2e^{-\ln(4)}\right)\leq 3/2+\varepsilon.
\end{aligned}
$$

Above we exploited the equation $\sum_{t=1}^{t_0} x^{t-1} = \frac{1-x^{t_0}}{1-x}$ for $0 < x < 1$. We also used the fact that $(1-\frac{1}{y})^{\ln(4)y} - (1-\frac{1}{2y})^{\ln(4)y}$ is an increasing function of $y > 1$, and that $\lim_{y\to\infty}(1-\frac{1}{y})^y = \frac{1}{e}$. □

5. A REFINED ANALYSIS

In this section we present a refined $(\ln(4)+\varepsilon)$ approximation bound for our Steiner tree algorithm.

We first give a high-level description of our analysis. Let $S^* := Opt$ be the optimal Steiner tree for the original instance (in particular, $c(S^*) = opt$). 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 $\ln(4)\cdot M$ iterations. This immediately implies the approximation factor of $\ln(4)+\varepsilon$.

In order to track which edges can be safely deleted from S^*, we will construct an artificial terminal spanning tree W (the *witness tree*) and assign a random subset $W(e)$ of edges of W to each edge $e \in S^*$ (the *witnesses* of e). At each iteration, when component C^t is sampled, we mark a proper *random* subset $\overline{Br}_W(C^t)$ of edges of W. As soon as all the edges of $W(e)$ are marked, edge e is deleted from S^*. Summarizing, we consider the following random process:

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

The subgraph S^t is formed by the edges of S^* which are not yet deleted at the beginning of iteration t.

The choice of $\overline{Br}_W(C^t)$ guarantees that, deterministically, the unmarked edges W' plus the sampled components connect all the terminals. The choice of $W(e)$

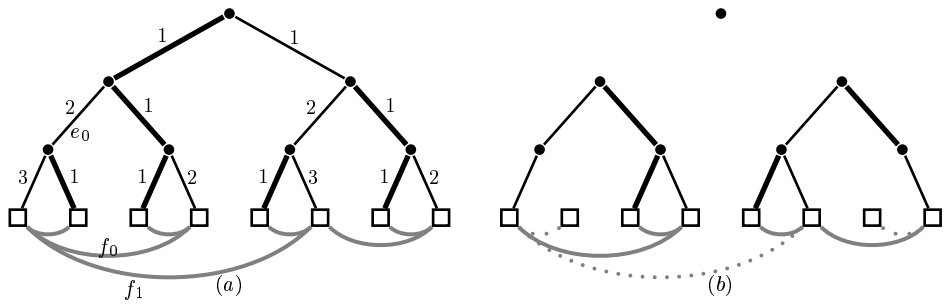

Fig. 4. (a) Optimal Steiner tree S^* in black, where bold edges indicate the chosen edges $\tilde{B}$, and the associated terminal spanning tree W in gray. Edges e in S^* are labeled with $|W(e)|$. For example $W(e_0) = \{f_0, f_1\}$. (b) Marked edges in W at a given iteration t are drawn dotted; the non-deleted edges in S^* (i.e., edges of S^t) are drawn in black. Non-marked edges of W and non-deleted edges of S^* support the same connectivity on R.

ensures that, deterministically, if W' plus the sampled components connect all the terminals, then the sampled components plus the undeleted edges $S^t = \{e \in S^* \mid W(e) \cap W' \neq \emptyset\}$ do the same. Hence the S^t's have the claimed connectivity properties. The analysis then reduces to show that all the edges in $W(e)$ are marked within a small enough number of iterations (in expectation).

We next define W, $W(\cdot)$, and $\overline{Br}_W(\cdot)$. Turn S^* into a rooted binary tree with the usual procedure. Recall that the height of the binary tree is at most $|R| - 1$. For each Steiner node, choose uniformly at random one of the two edges to its children. Let $\tilde{B}$ denote the chosen edges. Clearly $\Pr[e \in \tilde{B}] = \frac{1}{2}$ for any $e \in S^*$. Let $P_{uv} \subseteq S^*$ be the unique u-v path in S^*. The witness tree is

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

Similarly to arguments in Lemma 10, W is a terminal spanning tree. For each edge $e \in S^*$, define

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

See Figure 4(a) for an illustration. As we will see, $W(e)$ is small in expectation. It remains to define $\overline{Br}_W(\cdot)$. For a given component $C \in \mathbf{C}$, let the set of *candidate* bridges $\mathcal{B}_W(C)$ be

$$\mathcal{B}_W(C) := \{B \subseteq W \mid |B| = |R(C)| - 1, (W \backslash B) \cup C \text{ connects } V(W)\}.$$

Intuitively, $\mathcal{B}_W(C)$ is the family of bridge sets of W with respect to C that one obtains for varying cost functions. The set $\overline{Br}_W(C^t)$ is chosen randomly in $\mathcal{B}_W(C)$, according to a proper probability distribution $w_C : \mathcal{B}_W(C) \to [0, 1]$, which will be described in the following. Observe that $Br_W(C) \in \mathcal{B}_W(C)$. The intuitive reason for using a random element of $\mathcal{B}_W(C)$ rather than $Br_W(C)$ is that we wish to mark the edges of W in a *more uniform* way. This, in combination with the small size of $W(e)$, guarantees that edges are deleted quickly enough.

Next lemma shows that the undeleted edges plus the sampled components connect the terminals.

LEMMA 17. *The graph* $S^t \cup \bigcup_{t'=1}^{t-1} C^{t'}$ *spans* R.

PROOF. Let $W' \subseteq W$ be the set of edges which are not yet marked at the beginning of iteration t (see also Figure 4(b)). By the definition of $\mathcal{B}_W(C)$ and being $\overline{Br}_W(C) \in \mathcal{B}_W(C)$, $W' \cup \bigcup_{t'=1}^{t-1} C^{t'}$ spans R. Consider any edge $\{u,v\} \in W'$. 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. □

Note that $1 \leq |W(e)| \leq |R|-1$. Observe also that $|W(e)| = 1$ if $e \in \tilde{B}$. Indeed, the expected cardinality of $W(e)$ is small also for the remaining edges.

LEMMA 18. *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 $\tilde{B}$ 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 $\{u,v\} \in W$ that contains e (see also Figure 4(a)). This event happens with probability $1/2^q$. If there is no edge from $\tilde{B}$ on the path $v_0, v_1, \ldots, v_{k_e}$, $|W(e)| = k_e$ by a similar argument. The latter event happens with probability $1/2^{k_e}$. The claim follows. □

Next lemma proves the existence of random variables $\overline{Br}_W(\cdot)$ such that each edge of W is marked at each iteration with probability at least $1/M$. Its proof is based on a combination of Farkas' Lemma with our Bridge Lemma.

LEMMA 19. *There is a choice of the random variables* $\overline{Br}_W(\cdot)$ *such that each edge* $e \in W$ *is marked with probability at least* $1/M$ *at each iteration.*

PROOF. Consider any given iteration. Let $(x, \mathbf{C})$ be the corresponding solution to DCR, and C^* be the sampled component in that iteration. In particular, $C^* = C$ with probability $x_C/M = x_C/\sum_{C' \in \mathbf{C}} x_{C'}$. In this iteration we mark the edges $\overline{Br}_W(C^*)$, where $\Pr[\overline{Br}_W(C^*) = B] = w_{C^*}(B)$ for any $B \in \mathcal{B}_W(C^*)$. We will show that there is a choice of the w_C's, $C \in \mathbf{C}$, such that

$$\sum_{(C,B): B \in \mathcal{B}_W(C), e \in B} x_C \cdot w_C(B) \geq 1, \quad \forall e \in W.$$

This implies the claim since

$$\Pr[e \in \overline{Br}_W(C^*)] = \sum_{(C,B): B \in \mathcal{B}_W(C), e \in B} \frac{x_C}{M} \cdot w_C(B) \geq \frac{1}{M}.$$

Suppose by contradiction that such probability distributions w_C do not exist.

Then the following linear system has no solution[6]:

$$\begin{aligned} \sum_{B \in \mathcal{B}_W(C)} w_C(B) &\leq 1, \quad \forall C \in \mathbf{C}; \\ \sum_{(C,B): B \in \mathcal{B}_W(C), e \in B} x_C \cdot w_C(B) &\geq 1, \quad \forall e \in W; \\ w_C(B) &\geq 0, \quad \forall C \in \mathbf{C}, \forall B \in \mathcal{B}_W(C). \end{aligned}$$

Farkas' Lemma[7] yields that there is a vector $(y, c) \geq 0$ with

$$\begin{aligned} &(a) \quad y_C \geq \textstyle\sum_{e \in B} c_e x_C, \quad \forall C \in \mathbf{C}, \forall B \in \mathcal{B}_W(C); \\ &(b) \quad \textstyle\sum_{C \in \mathbf{C}} y_C < \sum_{e \in W} c_e. \end{aligned}$$

Let us interpret c as an edge cost function. In particular, $c(W) := \sum_{e \in W} c_e$ and $br_W(C)$ is the cost of the bridges of W with respect to component C and this cost function. One has

$$y_C \overset{(a)}{\geq} x_C \cdot \max\{c(B) \mid B \in \mathcal{B}_W(C)\} = x_C \cdot br_W(C).$$

Then

$$\sum_{C \in \mathbf{C}} x_C \cdot br_W(C) \leq \sum_{C \in \mathbf{C}} y_C \overset{(b)}{<} \sum_{e \in W} c_e = c(W),$$

which contradicts the Bridge Lemma 11. □

We next show that, for small $|W(e)|$, all the edges of $W(e)$ are marked (and hence e is deleted) within a small number of iterations. A handwaving argument works as follows. Let $|W(e)| = q$. Similarly to the *Coupons Collector problem* (see e.g. [Mitzenmacher and Upfal 2005]), it takes in expectation $\frac{M}{q}$ iterations until the first edge is marked, then $\frac{M}{q-1}$ iterations to hit the second one and so forth. Finally all edges are marked after an expected number of $M \cdot (\frac{1}{q} + \frac{1}{q-1} + \ldots + 1) = H_q \cdot M$ iterations. (Here $H_q := \sum_{i=1}^{q} \frac{1}{i}$ denotes the *q-th harmonic number*). However, this argument does not reflect the fact that a set $\overline{Br}_W(C^t)$ might contain several edges from $W(e)$. A more careful argument incorporates this complication.

For $\tilde{W} \subseteq W$, let $X(\tilde{W})$ denote the first iteration when all the edge in $\tilde{W}$ are marked. Observe that $S^t = \{e \in S^* \mid X(W(e)) \geq t\}$.

LEMMA 20. *Let $\tilde{W} \subseteq W$. Then the expected number of iterations until all edges in $\tilde{W}$ are marked satisfies*

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

PROOF. Let $q = |\tilde{W}|$. By m_q we denote the best possible upper bound on the expected number of iterations until all edges of $\tilde{W}$ are marked (over all feasible probability distributions). We will prove that $m_q \leq H_q \cdot M$ by induction on q.

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

For $q = 1$, the only edge in $\tilde{W}$ is marked with probability at least $\frac{1}{M}$ at each iteration, hence $m_1 \leq M$. Next, let $q > 1$ and consider the first iteration. Suppose that λ_i is the probability that at least i many edges are marked in this iteration. Since the expected number of marked edges must be at least $q \cdot \frac{1}{M}$ in the first iteration, this distribution has to satisfy $\sum_{i=1}^{q} \lambda_i \geq \frac{q}{M}$. Note that $\lambda_0 = 1$ and $\lambda_{q+1} = 0$. For notational convenience, let $m_0 := 0$.

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 following bound:

$$\begin{aligned}
m_q &\leq 1 + \sum_{i=0}^{q} \Pr\left[\begin{smallmatrix}\text{exactly } i \text{ edges marked} \\ \text{at the first iteration}\end{smallmatrix}\right] \cdot m_{q-i} \\
&\overset{\substack{\text{inductive}\\ \text{hypothesis}}}{\leq} 1 + M \cdot \sum_{i=1}^{q} (\lambda_i - \lambda_{i+1}) H_{q-i} + (1 - \lambda_1) m_q \\
&= 1 + M \cdot \sum_{i=1}^{q} \lambda_i \cdot \underbrace{(H_{q-i} - H_{q-i+1})}_{\leq -1/q} + \lambda_1 H_q M + (1 - \lambda_1) m_q \\
&\leq 1 - \frac{1}{q} M \cdot \underbrace{\sum_{i=1}^{q} \lambda_i}_{\geq q/M} + \lambda_1 H_q M + (1 - \lambda_1) m_q \\
&\leq \lambda_1 H_q M + (1 - \lambda_1) m_q.
\end{aligned}$$

From $\lambda_1 > 0$ we obtain $m_q \leq H_q \cdot M$. The claim follows. □

Now we have all the ingredients to prove the expected $(\ln(4) + \varepsilon)$ approximation factor.

THEOREM 21. *For any constant $\varepsilon > 0$, there is a polynomial-time randomized approximation algorithm for the Steiner tree problem with expected approximation ratio* $\ln(4) + \varepsilon$.

PROOF. 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

$$\begin{aligned}
E[D(e)] &= \sum_{q=1}^{k_e} \Pr[|W(e)| = q] \cdot E[D(e) \mid |W(e)| = q] \\
&\overset{\text{Lem 20}}{\leq} \sum_{q=1}^{k_e} \Pr[|W(e)| = q] \cdot H_q \cdot M \\
&\overset{\text{Lem 18}}{=} \sum_{q=1}^{k_e-1} \left(\frac{1}{2}\right)^q \cdot H_q \cdot M + \frac{2}{2^{k_e}} \cdot H_{k_e} \cdot M \\
&\leq \sum_{q\geq 1} \left(\frac{1}{2}\right)^q \cdot H_q \cdot M \\
&= M \cdot \sum_{q\geq 1} \frac{1}{q} \sum_{i\geq 0} \left(\frac{1}{2}\right)^{q+i} \\
&= M \cdot \sum_{q\geq 1} \frac{1}{q} \left(\frac{1}{2}\right)^{q-1} = \ln(4) \cdot M.
\end{aligned}$$

The expected cost of the approximate solution satisfies

$$\begin{aligned}
E\Big[\sum_{t\geq 1} c(C^t)\Big] &\leq \sum_{t\geq 1} \frac{1+\varepsilon/2}{M} E\big[opt_f^t\big] \\
&\leq \frac{1+\varepsilon/2}{M} \sum_{t\geq 1} E\left[c(S^t)\right] \\
&= \frac{1+\varepsilon/2}{M} \sum_{e\in S^*} E[D(e)] \cdot c(e) \leq (\ln(4)+\varepsilon) \cdot opt.
\end{aligned}$$

The claim follows. □

5.1 A $(\frac{73}{60}+\varepsilon)$-Approximation for Quasi-Bipartite Graphs

In this section we consider the special case of quasi-bipartite graphs. Recall that we call a graph $G = (V, E)$ quasi-bipartite if no pair of non-terminal nodes $u, v \in V \backslash R$ is connected by an edge. We show that our algorithm has an approximation ratio of at most $\frac{73}{60} + \varepsilon < 1.217$ (for ε small enough). This improves over the previously best known factor of 1.28 in [Robins and Zelikovsky 2005]. Note that Gröpl et al. [2002] show the bound of $\frac{73}{60}$ for the more restricted case of *uniform* quasi-bipartite graphs, where all edges incident to a non-terminal node have the same cost. For this class the integrality gap of the hypergraphic LP relaxation by Chakrabarty et al. [2010a] can also be bounded by $\frac{73}{60}$.

Again let S^* be an optimal Steiner tree, which now is quasi-bipartite. Let $Z_1, \ldots, Z_\ell$ be its (undirected) components. In particular, each Z_i is a star with a single Steiner node as center and terminals as leaves. We can improve the approximation guarantee by choosing the witness tree W in a more economical way, exploiting the structure of S^*. For each $i = 1, \ldots, \ell$, we add to the chosen edges $\tilde{B}$ all the edges of Z_i but one edge chosen uniformly at random. Again for $u, v \in R$,

let P_{uv} be the unique u-v path in S^*. We let

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

Observe that W will in fact be a terminal spanning tree. The analysis is now much simpler.

THEOREM 22. *For any constant $\varepsilon > 0$, there is a polynomial-time randomized approximation algorithm for the Steiner tree problem on quasi-bipartite graphs with expected approximation ratio $\frac{73}{60} + \varepsilon$.*

PROOF. We still consider the algorithm in Figure 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. Let k be the number of terminals in the star Z_i that contains e. With probability $\frac{1}{k}$ one has $|W(e)| = k-1$, and otherwise $|W(e)| = 1$. Hence, by Lemma 20,

$$E[D(e)] \leq \frac{1}{k} \cdot H_{k-1} \cdot M + \left(1 - \frac{1}{k}\right) \cdot H_1 \cdot M = \left(\frac{1}{k} \cdot H_{k-1} + \frac{k-1}{k}\right) \cdot M \leq \frac{73}{60} \cdot M.$$

In the last inequality we used the fact that $\frac{1}{k} \cdot H_{k-1} + \frac{k-1}{k}$ is maximized for $k = 5$. The claim follows along the same line as in Theorem 21. □

6. DERANDOMIZATION

In this section, we show how to derandomize the result from Section 5 using the method of *limited independence* (see, e.g., [Alon and Spencer 2008]). This way, we prove Theorem 3.

We start (Section 6.1) by presenting an alternative, phase-based algorithm, which updates the LP only a constant number of times (the phases). Then we show (Section 6.2) how to sample components in each phase with a logarithmic number of random bits.

6.1 A Phase-Based Randomized Algorithm

Consider the algorithm from Figure 5. The basic idea behind the algorithm is grouping iterations into phases. In each phase, we keep the LP unchanged. The details on how to sample components in each phase are given later.

(1) For phase $s = 1, 2, \ldots, 1/\varepsilon^2$
  (1a) Compute a $(1+\varepsilon)$-approximate *well-rounded* solution $(x^s, \mathbf{C}^s)$ to DCR (w.r.t. the current instance).
  (1b) Sample ϕ^s components $C^{s,1}, \ldots, C^{s,\phi^s}$ from $\mathbf{C}^s$ *according to* x^s, and contract them.
(2) Compute a minimum-cost terminal spanning tree T in the remaining instance.
(3) Output $T \cup \bigcup_{s=1}^{1/\varepsilon^2} \bigcup_{i=1}^{\phi^s} C^{s,i}$.

Fig. 5. Phase-based sampling algorithm

We may assume that the computed DCR solution $(x^s, \mathbf{C}^s)$ is *well-rounded*, i.e.,

- $|\mathbf{C}^s| = m$ for a prime number m,
- $x^s_C = \frac{1}{N}$ for all $C \in \mathbf{C}^s$ and $N \geq 1$ is bounded by a polynomial in n.

This can be achieved as follows: One computes a $(1+\frac{\varepsilon}{2})$-approximate solution $(x, \mathbf{C})$. Say $h = |\mathbf{C}|$. Then we round up all entries in x to the nearest multiple of $\frac{1}{N}$ for $N := 8h/\varepsilon$ and term the obtained solution x'. Using the generous estimate $c(C) \leq 2opt_f$ (following from Lemma 5) we obtain, for $\varepsilon \leq 1$,

$$\sum_{C \in \mathbf{C}} x'_C \cdot c(C) \leq \left(1 + \frac{\varepsilon}{2}\right) \cdot \left(opt_f + \sum_{C \in \mathbf{C}} \frac{\varepsilon}{8h} c(C)\right) \leq (1+\varepsilon) opt_f.$$

Next, replace a component C by $x'_C \cdot N$ many copies. Let m' be the number of obtained components (counted with multiplicities). Then we can compute a prime number $m \in [m', 2m']$ (see e.g. [Niven et al. 1991]) and add $m - m'$ dummy components C containing only the root, each one with $x'_C := \frac{1}{N}$. This yields a feasible well-rounded solution as desired. We furthermore assume[8] that $m \geq N/\varepsilon^2$ and $1/\varepsilon$ is integer.

For $\tilde{W} \subseteq W$, let $\bar{X}(\tilde{W})$ denote the first phase when all edges in $\tilde{W}$ are marked. Analogously, $\bar{D}(e)$ is the phase when all the edges in $W(e)$ are marked. For notational convenience, we interpret Step (2) as a final phase when all the edges of W are marked (so that $\bar{X}(\tilde{W})$ and $\bar{D}(e)$ are well defined). The next lemma is a simple adaptation of Lemma 20.

LEMMA 23. *Let $\tilde{W} \subseteq W$. Suppose each edge is marked at each phase with probability at least $p \in]0,1]$. Then the expected number of phases until all edges in $\tilde{W}$ are marked satisfies*

$$E[\bar{X}(\tilde{W})] \leq H_{|\bar{W}|} \cdot \frac{1}{p}.$$

PROOF. By a coupling argument, we can assume that the number of phases is unbounded. The claim follows along the same line as the proof of Lemma 20, replacing the notion of iteration with the notion of phase and the probability $1/M$ with p. □

We next bound the approximation factor of the algorithm for a generic sampling procedure (satisfying some properties).

LEMMA 24. *Suppose that Step (1b) satisfies the following two properties:*

(a) Each component C is sampled with probability at most $\alpha \cdot x^s_C$
(b) Each edge e in the witness tree is marked with probability at least β.

Then the approximation factor of the algorithm in Figure 5 is at most $\ln(4) \cdot (\frac{(1+\varepsilon)\alpha}{\beta} + \frac{2\varepsilon^2}{\beta})$.

PROOF. As in the proof of Theorem 21, one has

$$\begin{aligned} E[\bar{D}(e)] &= \sum_{q=1}^{k_e} \Pr[|W(e)| = q] \cdot E[\bar{D}(e) \mid |W(e)| = q] \\ &\overset{\text{Lem 18+23}}{\leq} \sum_{q \geq 1} \left(\frac{1}{2}\right)^q \cdot H_q \cdot \frac{1}{\beta} = \ln(4) \cdot \frac{1}{\beta}. \end{aligned} \tag{2}$$

[8] If $\mathbf{1}^T x = \frac{m}{N} = O(1)$ and $|C| = O(1)$ for $C \in \mathbf{C}$, then the number of terminals would be bounded by a constant – in this case an optimum solution can be computed in polynomial time.

Let opt^s be the cost of an optimal Steiner tree at the beginning of phase s. The expected cost of the sampled components satisfies

$$\begin{aligned} E\Big[\sum_{s=1}^{1/\varepsilon^2}\sum_{i=1}^{\phi_s} c(C^{s,i})\Big] &\le \sum_{s=1}^{1/\varepsilon^2}\sum_{C\in\mathbf{C}^s} E[\alpha\cdot x_C^s\cdot c(C)] \\ &\le \alpha(1+\varepsilon)\cdot\sum_{s=1}^{1/\varepsilon^2} E[opt^s] \\ &\le \alpha(1+\varepsilon)\sum_{e\in S^*} E[\bar{D}(e)]\cdot c(e) \overset{(2)}{\le} \ln(4)\cdot\frac{\alpha(1+\varepsilon)}{\beta}opt. \end{aligned}$$

Let $S' := \{e \in S^* \mid \bar{D}(e) > 1/\varepsilon^2\}$ be a feasible Steiner tree at the end of the last phase. By Markov's inequality and (2),

$$\Pr[\bar{D}(e) > 1/\varepsilon^2] \le \ln(4)\frac{\varepsilon^2}{\beta}.$$

Therefore $E[c(S')] \le \ln(4)\frac{\varepsilon^2}{\beta}\cdot opt$. The minimum-cost terminal spanning tree is at most twice that expensive, hence $E[c(T)] \le 2\ln(4)\frac{\varepsilon^2}{\beta}\cdot opt$. The claim follows. □

Lemma 24 suggests an alternative way to implement the algorithm from Section 5. Consider the following natural implementation of Step (1b):

> (Independent Phase Sampling)
> Sample $\phi^s = \varepsilon\cdot M$ components $C^{s,1},\ldots,C^{s,\phi^s}$ independently (with repetitions), where $C^{s,i} = C \in \mathbf{C}^s$ with probability x_C^s/M.

The Independent Phase Sampling samples a component C with a probability of at most $\phi^s\cdot\frac{1}{M}\cdot x_C^s = \varepsilon\cdot x_C^s$. On the other hand, the probability that edge $e \in W$ is marked is essentially lower bounded by ε. Inspecting Lemma 24, we see that $\alpha \approx \beta \approx \varepsilon$, which gives the following corollary.

Corollary 25. *The algorithm from Figure 5 which implements Step (1b) with the* Independent Phase Sampling *is* $(\ln(4)+O(\varepsilon))$*-approximate in expectation.*

This provides a 1.39-approximation algorithm that needs to solve just a constant (rather than polynomial) number of LPs. In particular, its running time might be competitive with the better-than-2 approximation algorithms in the literature. But a drawback of the Independent Phase Sampling implementation is that it needs too many (namely polynomially many) random bits: hence it is not easy to derandomize. For this reason we introduce a more complex sampling procedure in the next subsection.

6.2 A Dependent Sampling Procedure

We next describe an alternative implementation of Step (1b), which still guarantees $\alpha \approx \beta \approx \varepsilon$, and requires only $O(\log n)$ random bits. We focus on a specific phase s and an edge $e \in W$. Let $(x, \mathbf{C}) := (x^s, \mathbf{C}^s)$. We renumber the components such that $\mathbf{C} = (C_0,\ldots,C_{m-1})$, and we let $x_j := x_{C_j} = \frac{1}{N}$.

> (Dependent Phase Sampling)

(i) Choose $A \in \{0, \ldots, m-1\}$ and $B \in \{1, \ldots, m-1\}$ uniformly and independently at random.
(ii) Select C_j with $j \in J := \{A + i \cdot B \mod m \mid i = 1, \ldots, \lfloor \frac{\varepsilon}{N} m \rfloor\}$.

Observe that Step (ii) requires only $O(\log m)$ random bits. Since $m = n^{O(1)}$, this number of bits is $O(\log n)$.

We will show that: (1) any component C_j is sampled with probability no more than $\alpha\, x_j$, $\alpha := \varepsilon$ and (2) edge e is marked with probability at least $\beta := \varepsilon(1 - 2\varepsilon)$. The first claim is easy to show.

LEMMA 26. *Implementing Step (1b) with the* DEPENDENT PHASE SAMPLING, *each component C_j is sampled with probability at most $\varepsilon \cdot x_j$.*

PROOF. For any component C_j, $\Pr[j \in J] = \frac{1}{m} \cdot \lfloor \frac{\varepsilon}{N} m \rfloor \leq \frac{\varepsilon}{N} = \varepsilon \cdot x_j$. □

Showing claim (2) is more involving.

LEMMA 27. *Implementing Step (1b) with the* DEPENDENT PHASE SAMPLING, *each edge $e \in W$ is marked with probability at least $\varepsilon(1 - 2\varepsilon)$.*

PROOF. Let w_{C_j} be the probability distribution for component C_j as in Lemma 19. Recall that $\Pr[\overline{Br}_W(C_j) = B] = w_{C_j}(B)$ and

$$\delta := \sum_{j=0}^{m-1} x_j \sum_{B \in \mathcal{B}_W(C_j): e \in B} w_{C_j}(B) \geq 1.$$

Let $y_j := \sum_{B \in \mathcal{B}_W(C_j): e \in B} w_{C_j}(B)$ denote the probability that e is marked, given that C_j is sampled. Since $x_j = \frac{1}{N}$, we have $\sum_{j=0}^{m-1} y_j = \delta N$. There lies no harm in assuming that $\delta = 1$, since the probability that e is marked is increasing in the y_j's.

Let E_j be the event that C_j is sampled *and* $e \in \overline{Br}_W(C_j)$. It is sufficient to show that $\Pr[\bigcup_{j=0}^{m-1} E_j] \geq \varepsilon(1 - 2\varepsilon)$. The crucial insight is to obtain a *lower* bound on $\Pr[E_j]$ and an *upper* bound on $\Pr[E_j \cap E_{j'}]$ for $j \neq j'$. First of all, we have

$$\Pr[E_j] = y_j \cdot \Pr[j \in J] = y_j \cdot \left\lfloor \frac{\varepsilon m}{N} \right\rfloor \cdot \frac{1}{m} \geq \varepsilon(1 - \varepsilon) \frac{y_j}{N}, \tag{3}$$

using that $\frac{\varepsilon m}{N} \geq \frac{1}{\varepsilon}$ by assumption. Secondly, let $j, j' \in \{0, \ldots, m-1\}$ be distinct component indices. Then $j, j' \in J$ if and only if the system

$$\begin{aligned} j &\equiv_m A + Bi \\ j' &\equiv_m A + Bi' \end{aligned} \tag{4}$$

has a solution i, i'. But since $\mathbb{Z}_m$ is a field, for any distinct pair $i, i' \in \{1, \ldots, \lfloor \frac{\varepsilon m}{N} \rfloor\}$, there is precisely one pair $(A, B) \in \{0, \ldots, m-1\} \times \{1, \ldots, m-1\}$ satisfying (4). Hence

$$\Pr[E_j \cap E_{j'}] \leq y_j \cdot y_{j'} \cdot \left\lfloor \frac{\varepsilon m}{N} \right\rfloor \cdot \left(\left\lfloor \frac{\varepsilon m}{N} \right\rfloor - 1 \right) \cdot \frac{1}{m \cdot (m-1)} \leq y_j \cdot y_{j'} \cdot \frac{\varepsilon^2}{N^2}. \tag{5}$$

By the inclusion-exclusion principle (see, e.g., Corollary A.2 in [Arora and Barak 2009]),

$$\begin{aligned}
\Pr\Big[\bigcup_{j=0}^{m-1} E_j\Big] &\geq \sum_{j=0}^{m-1} \Pr[E_j] - \sum_{j=0}^{m-1}\sum_{j'\neq j} \Pr[E_j \cap E_{j'}] \\
&\overset{(3)+(5)}{\geq} \sum_{j=0}^{m-1} \varepsilon(1-\varepsilon)\frac{y_j}{N} - \frac{\varepsilon^2}{N^2} \underbrace{\sum_{j=0}^{m-1} y_j}_{=N} \cdot \underbrace{\sum_{j'\neq j} y_{j'}}_{\leq N} \\
&\geq \varepsilon(1-\varepsilon) - \varepsilon^2 = \varepsilon(1-2\varepsilon),
\end{aligned}$$

which proves the claim. □

A deterministic $(\ln(4)+\varepsilon)$-approximation algorithm easily follows.

Proof of Theorem 3. Consider the algorithm from Figure 5 which implements Step (1b) with the Dependent Phase Sampling. This algorithm can be derandomized by considering all the possible outcomes of random variables A and B in each phase, which are at most m^{2/ε^2}. The claim on the approximation follows from Lemmas 24, 26, and 27. □

We can similarly derandomize the result for quasi-bipartite graphs.

Theorem 28. *For any constant $\varepsilon > 0$, there is a deterministic polynomial-time algorithm for the Steiner tree problem on quasi-bipartite graphs with approximation ratio $\frac{73}{60}+\varepsilon$.*

Proof. Consider the same algorithm as in Theorem 3. Lemmas 23, 26, and 27 still hold. Under the same assumptions as in Lemma 24, and by the different choice of the witness tree W in this case, we now have

$$E[\bar{D}(e)] \leq \frac{1}{k}\cdot H_{k-1}\cdot\frac{1}{\beta} + \Big(1-\frac{1}{k}\Big)\cdot H_1\cdot\frac{1}{\beta} = \Big(\frac{1}{k}\cdot H_{k-1} + \frac{k-1}{k}\Big)\cdot\frac{1}{\beta} \leq \frac{73}{60}\cdot\frac{1}{\beta}.$$

Then the expected cost of the sampled components satisfies $E[\sum_{s,i} c(C^{s,i})] \leq \frac{73}{60}\frac{\alpha(1+\varepsilon)}{\beta}\cdot opt$. Similarly, the expected cost of the final spanning tree satisfies $E[c(T)] \leq 2\cdot\frac{73}{60}\frac{\varepsilon^2}{\beta}\cdot opt$. Altogether, the approximation factor from Lemma 24 now reduces to $\frac{73}{60}\cdot(\frac{(1+\varepsilon)\alpha}{\beta}+\frac{2\varepsilon^2}{\beta})$. The claim follows along the same line as in Theorem 3. □

7. INTEGRALITY GAP

In this section we upper bound (Section 7.1) and lower bound (Section 7.2) the integrality gap of DCR. Furthermore, we compare DCR with BCR (Section 7.3).

7.1 An Upper Bound

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) upper bound on the integrality gap of the studied LP. It only provides a $1+\ln(2)$ upper bound, as shown in Theorem 13 (by letting ε tend to zero). This is because the LP changes during

the iterations of the algorithm, and its solution is only bounded with respect to the initial optimal integral solution. In this section we prove that our LP has integrality gap at most $1 + \ln(3)/2 < 1.55$. Before proceeding with our (fairly technical) argument, let us remark that, after the conference version of this paper appeared, a shorter and perhaps more elegant proof (still based on the Bridge Lemma) of the same claim was recently given in [Chakrabarty et al. 2010b].

In order to prove the 1.55 upper bound on the integrality gap of DCR, claimed in Theorem 4, we consider the algorithm R&Z by Robins and Zelikovsky [2005]. We show that this algorithm produces solutions of cost bounded with respect to the optimal fractional solutions to k-DCR (and hence of DCR). This is achieved by combining the original analysis in [Robins and Zelikovsky 2005] with our new Bridge Lemma 11. This approach was, to some extent, inspired by an argument in [Charikar and Guha 2005] in the context of facility location. We leave it as an interesting open problem to prove a $\ln(4)$ (or even 1.5) upper bound on the integrality gap of DCR (if possible). This might involve the development of a fractional version of Lemma 14.

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 (k-component). Let $T^t[C]$ denote the minimum spanning tree of the graph $T^t \cup C$, where the edges $e \in C$ have weight 0 and the edges $f \in T^t$ weight $c^t(f)$. The subset of edges in T^t but not in $T^t[C]$ are denoted by $Br_{T^t}(C)$. In fact, $Br_{T^t}(C)$ is the set of bridges of T^t with respect to $R(C)$ and the above weight function. For a given component C, we denote as $Loss(C)$ the minimum-cost subforest of C with the property that there is a path between each Steiner node in C and some terminal in $R(C)$. In the terminology from Section 3, $Loss(C)$ is the complement of the set of bridges of the subtree C after contracting $R(C)$. We let $loss(C) = c(Loss(C))$.

It is convenient to define the following quantities:

$$gain^t(C) = br_{T^t}(C) - c(C) \qquad \text{and} \qquad sgain^t(C) = gain^t(C) + loss(C).$$

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 29. *[Robins and Zelikovsky 2005] 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 30. *[Robins and Zelikovsky 2005] For any $\ell \leq \mu$,*

$$apx_k \leq \sum_{t=1}^{\ell} loss(C^t) + c^\ell(T^\ell).$$

Recall that $opt_{f,k}$ is the cost of the optimal fractional solution to k-DCR. Let $(x, \mathbf{C})$ be an optimal fractional solution to k-DCR. Define $loss_{f,k} := \sum_{C \in \mathbf{C}} x_C \, loss(C)$.

COROLLARY 31. $loss_{f,k} \leq \frac{1}{2} opt_{f,k}$.

PROOF. From Lemma 6, for any $C \in \mathbf{C}$, $loss(C) = c(C) - br_C(R(C)) \leq \frac{1}{2} c(C)$. As a consequence, $loss_{f,k} \leq \frac{1}{2} \sum_{C \in \mathbf{C}} x_C \cdot c(C) = \frac{1}{2} opt_{f,k}$. □

COROLLARY 32. $c^{\mu}(T^{\mu}) \leq opt_{f,k}$.

PROOF. Using the fact that $gain^{\mu}(C) = br_{T^{\mu}}(C) - c(C) \leq 0$ for any component C,

$$c^{\mu}(T^{\mu}) \overset{\text{Bridge Lem 11}}{\leq} \sum_{C \in \mathbf{C}} x_C br_{T^{\mu}}(C) \leq \sum_{C \in \mathbf{C}} x_C c(C) = opt_{f,k}.$$

□

By Corollary 32, and since $c^t(T^t)$ is a non-increasing function of t, there must be a value of $\ell \leq \mu$ such that:

$$c^{\ell-1}(T^{\ell-1}) > opt_{f,k} \geq c^{\ell}(T^{\ell}). \tag{6}$$

In the following we will bound $\sum_{t=1}^{\ell} loss(C^t) + c^{\ell}(T^{\ell})$. By Lemma 30, this will give a bound on apx_k. Let

$$gain_f^t := c^t(T^t) - opt_{f,k} \qquad \text{and} \qquad sgain_f^t := gain_f^t + loss_{f,k}.$$

LEMMA 33. *For* $t = 1, 2, \ldots, \mu$, $\frac{sgain^{t-1}(C^t)}{loss(C^t)} \geq \frac{sgain_f^{t-1}}{loss_{f,k}}$.

PROOF. We first note that

$$\begin{aligned}
\frac{gain_f^{t-1}}{loss_{f,k}} &= \frac{c^{t-1}(T^{t-1}) - \sum_{C \in \mathbf{C}} x_C c(C)}{\sum_{C \in \mathbf{C}} x_C loss(C)} \\
&\overset{\text{Bridge Lem 11}}{\leq} \frac{\sum_{C \in \mathbf{C}} x_C (br_{T^{t-1}}(C) - c(C))}{\sum_{C \in \mathbf{C}} x_C loss(C)} \\
&= \frac{\sum_{C \in \mathbf{C}} x_C gain^{t-1}(C)}{\sum_{C \in \mathbf{C}} x_C loss(C)} \\
&\leq \max_{C \in \mathbf{C}} \left\{ \frac{gain^{t-1}(C)}{loss(C)} \right\} \leq \frac{gain^{t-1}(C^t)}{loss(C^t)},
\end{aligned}$$

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

$$\frac{sgain^{t-1}(C^t)}{loss(C^t)} = 1 + \frac{gain^{t-1}(C^t)}{loss(C^t)} \geq 1 + \frac{gain_f^{t-1}}{loss_{f,k}} = \frac{sgain_f^{t-1}}{loss_{f,k}}.$$

□

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

$$sgain^1 = c^{\ell-1}(T^{\ell-1}) - opt_{f,k} \overset{(6)}{>} 0. \tag{7}$$

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

$$\frac{sgain^{\ell-1}(C^\ell)}{loss(C^\ell)} = \frac{sgain^1}{loss^1} = \frac{sgain^2}{loss^2}. \tag{8}$$

Eventually, we define

$$\begin{aligned} sgain_f^{\ell 1} &:= sgain_f^{\ell-1} - sgain^1 \\ &\overset{(7)}{=} c^{\ell-1}(T^{\ell-1}) - opt_{f,k} + loss_{f,k} - (c^{\ell-1}(T^{\ell-1}) - opt_{f,k}) = loss_{f,k}. \end{aligned} \tag{9}$$

LEMMA 34. $\sum_{t=1}^{\ell-1} loss(C^t) + loss^1 \leq loss_{f,k} \ln\left(\frac{sgain_f^0}{sgain_f^{\ell 1}}\right)$.

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

$$sgain_f^t = sgain_f^{t-1} - sgain^{t-1}(C^t) \overset{\text{Lem 33}}{\leq} sgain_f^{t-1}\left(1 - \frac{loss(C^t)}{loss_{f,k}}\right).$$

Furthermore

$$\frac{sgain_f^{\ell-1}}{loss_{f,k}} \overset{\text{Lem 33}}{\leq} \frac{sgain^{\ell-1}(C^\ell)}{loss(C^\ell)} \overset{(8)}{=} \frac{sgain^1}{loss^1},$$

from which

$$sgain_f^{\ell 1} = sgain_f^{\ell-1} - sgain^1 \leq sgain_f^{\ell-1}\left(1 - \frac{loss^1}{loss_{f,k}}\right).$$

Then

$$\frac{sgain_f^{\ell 1}}{sgain_f^0} \leq \left(1 - \frac{loss^1}{loss_{f,k}}\right) \prod_{t=1}^{\ell-1} \left(1 - \frac{loss(C^t)}{loss_{f,k}}\right).$$

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

$$\ln\left(\frac{sgain_f^0}{sgain_f^{\ell 1}}\right) \geq \frac{1}{loss_{f,k}} \left(\sum_{t=1}^{\ell-1} loss(C^t) + loss^1\right).$$

□

We now have all the ingredients to bound the approximation factor of the algorithm with respect to $opt_{f,k}$. Let $mst = c(T^0) = c^0(T^0)$. The following theorem and corollary are straightforward adaptations of analogous results in [Robins and Zelikovsky 2005].

THEOREM 35. $apx_k \leq opt_{f,k} + loss_{f,k} \ln\left(\frac{mst - opt_{f,k} + loss_{f,k}}{loss_{f,k}}\right)$.

PROOF. Since $sgain^{t-1}(C^t) \geq loss(C^t)$, it follows from (8) that

$$sgain^2 \geq loss^2. \tag{10}$$

Putting everything together we obtain

$$\begin{aligned}
apx_k &\overset{\text{Lem 30}}{\leq} \sum_{t=1}^{\ell} loss(C^t) + c^\ell(T^\ell) \\
&\overset{\text{Lem 29}}{=} \sum_{t=1}^{\ell-1} loss(C^t) + loss(C^\ell) + c^{\ell-1}(T^{\ell-1}) - sgain^{\ell-1}(C^\ell) \\
&= \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + loss^2 + c^{\ell-1}(T^{\ell-1}) - sgain^1 - sgain^2 \\
&\overset{(10)}{\leq} \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + c^{\ell-1}(T^{\ell-1}) - sgain^1 \\
&\overset{(7)}{=} \sum_{t=1}^{\ell-1} loss(C^t) + loss^1 + opt_{f,k} \\
&\overset{\text{Lem 34}}{\leq} opt_{f,k} + loss_{f,k} \ln\left(\frac{sgain_f^0}{sgain_f^{\ell 1}}\right) \\
&\overset{(9)}{=} opt_{f,k} + loss_{f,k} \ln\left(\frac{mst - opt_{f,k} + loss_{f,k}}{loss_{f,k}}\right).
\end{aligned}$$

□

LEMMA 36. *For any constant $k \geq 2$, there exists a polynomial-time algorithm for Steiner tree which computes a solution of cost at most $1+\ln(3)/2$ times the cost of the optimal fractional solution to k-DCR.*

PROOF. A straightforward adaptation of Lemma 5 implies that

$$mst \leq 2opt_{f,k}.$$

Combining the inequality above with Theorem 35, we obtain

$$apx_k \leq opt_{f,k} + loss_{f,k} \ln\left(1 + \frac{2opt_{f,k} - opt_{f,k}}{loss_{f,k}}\right).$$

The right-hand side of the inequality above is an increasing function of $loss_{f,k}$. By Corollary 31, $loss_{f,k} \leq \frac{1}{2} opt_{f,k}$, which implies

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

□

Theorem 4 follows.

PROOF OF THEOREM 4. From Lemma 7, $opt_{f,k} \leq \rho_k \cdot opt_f$. The claim follows from Lemma 36 and Theorem 1 by choosing a large-enough k. □

7.2 A Lower Bound

The best-known lower bound on the integrality gap of BCR (prior to our work) is 8/7 [Könemann et al. 2007]. We can use the same family of instances to prove the same lower bound for DCR.

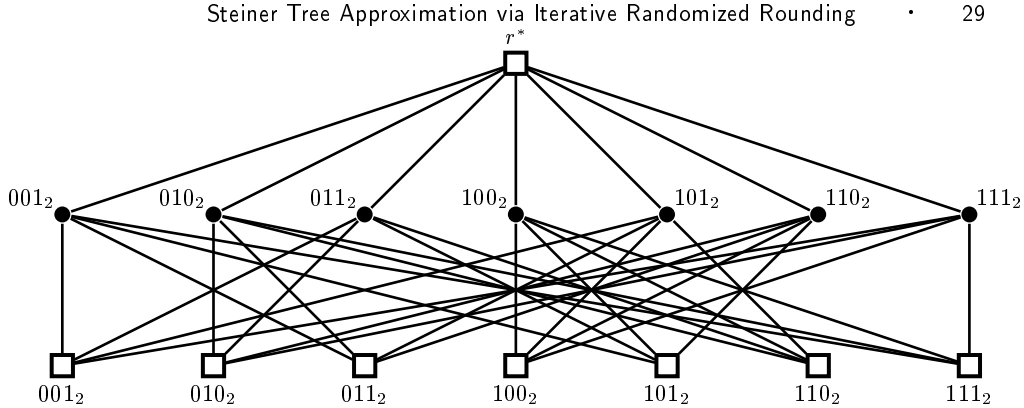

Fig. 6. Skutella's graph. Nodes are labeled with their indices in binary representation.

THEOREM 37. *The integrality gap of DCR is at least* $8/7 > 1.142$.

PROOF. We will use Skutella's graph [Könemann et al. 2007]. Consider a Set Cover instance with elements $U = \{1, \ldots, 7\}$ and sets $S_1, \ldots, S_7$. Let $b(i)$ be a vector from $\mathbb{Z}_2^3$ that is the binary representation of i, for example $b(3) = (0, 1, 1)$. We define the sets by $S_j := \{i \in U \mid b(i) \cdot b(j) \equiv_2 1\}$. Note that this is exactly the definition of the instance which yields a $\Omega(\log n)$ lower bound on the integrality gap of Set Cover for $n = 7$ [Vazirani 2001]. The critical property is that for our particular instance one needs 3 sets to cover all elements, but choosing each set to an extent of $1/4$ gives a fractional Set Cover solution of cost $7/4$.

Next we define a graph where each element forms a terminal and each set is a non-terminal node connected to the root and to the contained elements by unit cost edges (see Figure 6).

If we direct all the edges upwards, the graph can be decomposed into 7 edge-disjoint components, each one containing one non-terminal node and the 5 edges incident into it. On one hand installing $1/4$ on each of these components gives a fractional solution of cost $35/4$, while on the other hand at least 3 Steiner nodes must be included for an integer solution. Consequently $opt = 10$ and we obtain the promised gap of $\frac{10}{35/4} = \frac{8}{7}$. □

7.3 Comparison with BCR

We start by observing that DCR is a relaxation strictly stronger than BCR.

LEMMA 38. *Let* opt_{DCR} *and* opt_{BCR} *be the optimal fractional solutions to DCR and BCR, respectively, for a given input instance. Then* $opt_{DCR} \geq opt_{BCR}$ *and there are examples where strict inequality holds.*

PROOF. Any feasible solution to 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. This proves the first part of the claim. An example of strict inequality is given in Figure 7. □

Observe that the 1.55 upper bound on the integrality gap of DCR does not imply the same bound on the integrality gap of BCR. It remains as a challenging open problem to show whether the integrality gap of BCR is smaller than 2 or not.

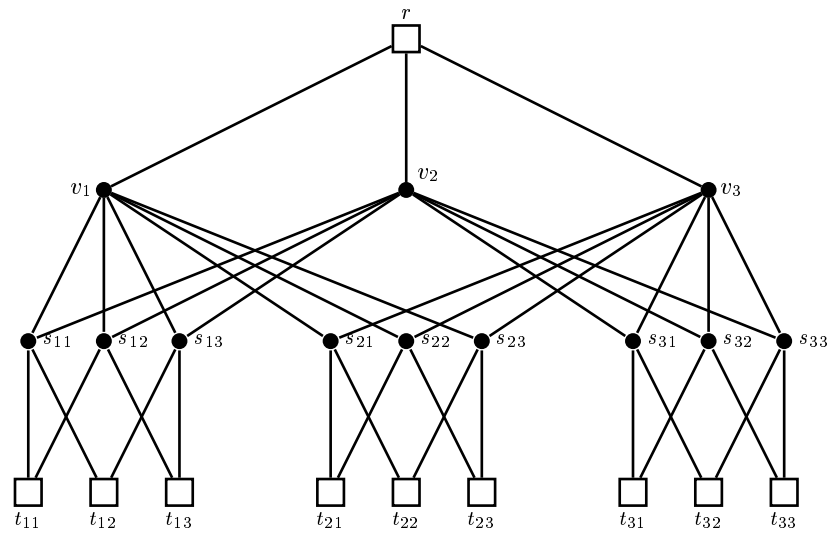

Fig. 7. All edges have cost 1. The unique optimal solution to BCR, of cost 15, installs capacity 1/4 on the central edges and capacity 1/2 on the remaining edges (always directed upwards). The cheapest solution to DCR has cost $7/4 \cdot 9 = 15.75$. The overall capacity reserved on each edge is the same as in the BCR case, excluding the top edges, where the capacity is 3/4. The (integral) optimal Steiner tree has cost 17.

The best-known lower bound on the integrality gap of BCR is $8/7 > 1.142$ [Könemann et al. 2007; Vazirani 2001]. In particular, the family of instances which provides this bound is the same as in Section 7.2. We next present an improved lower bound of 36/31 on the integrality gap of BCR.

THEOREM 39. *The integrality gap of BCR is at least* $36/31 > 1.161$.

PROOF. The basic idea is generalizing the construction used in Section 7.2. Let $p \in \mathbb{N}$ be a parameter. We create a graph with $p+2$ *levels* and unit cost edges. For $i \in \{1, \ldots, p\}$ one has 7^i non-terminal nodes on the ith level, each represented by a vector from U^i, where $U = \{1, \ldots, 7\}$. Furthermore we have a root terminal on level 0 and 7^p terminals on the $(p+1)$th level, represented by vectors from U^p. We connect the root to all nodes in the first level. For $i = 1, \ldots, p$, consider nodes $u = (u_1, \ldots, u_i) \in U^i$ on level i and $v = (v_1, \ldots, v_{i+1}) \in U^{i+1}$ on level $i+1$. We connect u and v by an edge if $(u_1, \ldots, u_{i-1}) = (v_1, \ldots, v_{i-1})$ and $b(u_i) \cdot b(v_i) \equiv_2 1$. We connect the non-terminal node $u \in U^p$ on level p with terminal $v \in U^p$ on level $p+1$ in a similar manner, namely if and only if $(u_1, \ldots, u_{p-1}) = (v_1, \ldots, v_{p-1})$ and $b(u_p) \cdot b(v_p) \equiv_2 1$. Observe that, for $p = 1$, we obtain exactly Skutella's graph. The graph obtained for $p = 2$ is depicted in Figure 8.

Let us consider any integer optimal solution, of cost *opt*, and direct the edges towards r^*. Each time we have an edge going from a level i downwards to level $i+1$ we can replace it by an edge to level $i-1$ without disconnecting the tree. Observe that, for $i = 0, \ldots, p-1$, we need at least $3 \cdot 7^i$ edges between level i and $i+1$ and that 7^p edges are needed between the last two levels. This amount of

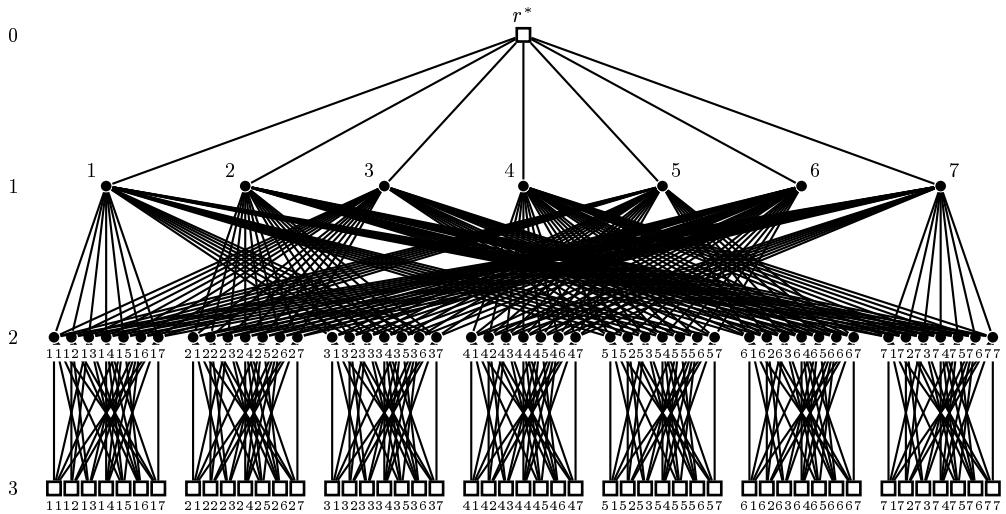

Fig. 8. Instance for $p = 2$. Nodes are labeled with the corresponding vector in abbreviated notation; all edges have unit costs. The optimal fractional solution consist of installing capacity 1/16 on each edge from level 2 to level 1 and capacity 1/4 otherwise (always directed "upwards"), thus $opt_f = 7^2+7^2/4+7/4 = 63$. On the other hand for an integer solution one needs $3+3\cdot 7+7^2 = 73$ edges. The gap for this instance is consequently $\frac{73}{63} \approx 1.158$.

edges is also sufficient, thus

$$opt = 3 \cdot (7^0 + 7^1 + \ldots + 7^{p-1}) + 7^p = \frac{3}{2} \cdot 7^p - \frac{1}{2}.$$

Consider now the optimal fractional solution to BCR for the same instance. Let opt_f^p denote its cost. This solution installs capacity 1/4 on the edges incident to the root and to the terminals, and capacity 1/16 on the remaining edges (all directed upwards). Hence

$$opt_f^p = \frac{4}{4}7^p + \frac{4}{16} \cdot (7^1 + 7^2 + \ldots + 7^p) = \frac{31}{24} \cdot 7^p - \frac{7}{24}.$$

The claim follows since

$$\lim_{p\to\infty} \frac{opt}{opt_f^p} = \frac{36}{31}.$$

□

ACKNOWLEDGMENTS

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 of the conference version of this paper for many suggestions on how to improve the presentation.

REFERENCES

Agrawal, A., Klein, P., and Ravi, R. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing 24*, 440–456.

Alon, N. and Spencer, J. 2008. *The probabilistic method.* John Wiley & Sons Inc.

Archer, A., Bateni, M., Hajiaghayi, M., and Karloff, H. 2009. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *IEEE Symposium on Foundations of Computer Science (FOCS).* 427–436.

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

Arora, S. and Barak, B. 2009. *Computational complexity.* Cambridge University Press, Cambridge. A modern approach.

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

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

Borradaile, G., Kenyon-Mathieu, C., and Klein, P. 2007. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *ACM-SIAM Symposium on Discrete Algorithms (SODA).* 1285–1294.

Byrka, J., Grandoni, F., Rothvoss, T., and Sanità, L. 2010. An improved LP-based approximation for Steiner tree. In *ACM Symposium on Theory of Computing (STOC).* 583–592.

Chakrabarty, D., Devanur, N. R., and Vazirani, V. V. 2008. New geometry-inspired relaxations and algorithms for the metric Steiner tree problem. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO).* 344–358.

Chakrabarty, D., Könemann, J., and Pritchard, D. 2010a. Hypergraphic LP relaxations for Steiner trees. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO).* To appear.

Chakrabarty, D., Könemann, J., and Pritchard, D. 2010b. Integrality gap of the hypergraphic relaxation of Steiner trees: a short proof of a 1.55 upper bound. Submitted.

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

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

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

Edmonds, J. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B71*, 233–240.

Eisenbrand, F. and Grandoni, F. 2005. An improved approximation algorithm for virtual private network design. In *ACM-SIAM Symposium on Discrete Algorithms (SODA).* 928–932.

Eisenbrand, F., Grandoni, F., Oriolo, G., and Skutella, M. 2007. New approaches for virtual private network design. *SIAM Journal on Computing 37,* 3, 706–721.

Eisenbrand, F., Grandoni, F., Rothvoss, T., and Schäfer, G. 2008. Approximating connected facility location problems via random facility sampling and core detouring. In *ACM-SIAM Symposium on Discrete Algorithms (SODA).* 1174–1183.

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

Garey, M. R. and Johnson, D. S. 1979. *Computers and intractability.* W. H. Freeman and Co.

Gilbert, E. N. and Pollak, H. O. 1968. Steiner minimal trees. *SIAM Journal on Applied Mathematics 16,* 1, 1–29.

Goemans, M. X. and Myung, Y. 1993. A catalog of Steiner tree formulations. *Networks 23,* 1, 19–28.

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

Grandoni, F. and Italiano, G. F. 2006. Improved approximation for single-sink buy-at-bulk. In *International Symposium on Algorithms and Computation (ISAAC).* 111–120.

Grandoni, F. and Rothvoss, T. 2010. Network design via core detouring for problems without a core. In *International Colloquium on Automata, Languages and Programming (ICALP)*.

Gröpl, C., Hougardy, S., Nierhoff, T., and Prömel, H.-J. 2002. Steiner trees in uniformly quasi-bipartite graphs. *Information Processing Letters 83,* 4, 195–200.

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

Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R., and Yener, B. 2001. Provisioning a virtual private network: a network design problem for multicommodity flow. In *ACM Symposium on Theory of Computing (STOC)*. 389–398.

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

Hwang, F., Richards, D., and Winter, P. 1992. *The Steiner tree problem*. Monograph in Annals of Discrete Mathematics, 53. Elsevier.

Jain, K. 1998. A factor 2 approximation algorithm for the generalized Steiner network problem. In *IEEE Symposium on Foundations of Computer Science (FOCS)*. 448–457.

Jothi, R. and Raghavachari, B. 2009. Improved approximation algorithms for the single-sink buy-at-bulk network design problems. *Journal of Discrete Algorithms 7,* 2, 249–255.

Karpinski, M. and Zelikovsky, A. 1997. New approximation algorithms for the Steiner tree problem. *Journal of Combinatorial Optimization 1,* 1, 47–65.

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

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

Korte, B. and Vygen, J. 2002. *Combinatorial Optimization - Theory and Algorithms*. Springer-Verlag.

Mitzenmacher, M. and Upfal, E. 2005. *Probability and computing*. Cambridge University Press, Cambridge. Randomized algorithms and probabilistic analysis.

Mölle, D., Richter, S., and Rossmanith, P. 2006. A faster algorithm for the Steiner tree problem. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*. 561–570.

Niven, I., Zuckerman, H. S., and Montgomery, H. L. 1991. *An introduction to the theory of numbers*. John Wiley & Sons Inc.

Polzin, T. and Vahdati-Daneshmand, S. 2003. On Steiner trees and minimum spanning trees in hypergraphs. *Operations Research Letters 31,* 1, 12–20.

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

Rajagopalan, S. and Vazirani, V. V. 1999. On the bidirected cut relaxation for the metric Steiner tree problem. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 742–751.

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

Robins, G. and Zelikovsky, A. 2000. Improved Steiner tree approximation in graphs. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 770–779.

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

Rothvoss T. and Sanità, L. 2009. On the complexity of the asymmetric VPN problem. In *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 326–338.

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

Talwar, K. 2002. The single-sink buy-at-bulk LP has constant integrality gap. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 475–486.

Vazirani, V. V. 2001. *Approximation Algorithms*. Springer-Verlag.

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