Packing Element-Disjoint Steiner Trees

Joseph Cheriyan
University of Waterloo
and
Mohammad R. Salavatipour
University of Alberta

Given an undirected graph $G(V, E)$ with terminal set $T \subseteq V$ the problem of packing element-disjoint Steiner trees is to find the maximum number of Steiner trees that are disjoint on the nonterminal nodes and on the edges. The problem is known to be NP-hard to approximate within a factor of $\Omega(\log n)$, where n denotes $|V|$. We present a randomized $O(\log n)$-approximation algorithm for this problem, thus matching the hardness lower bound. Moreover, we show a tight upper bound of $O(\log n)$ on the integrality ratio of a natural linear programming relaxation.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures; G.2.2 [**Discrete Mathematics**]: Graph Algorithms

General Terms: Algorithms, Approximation

Additional Key Words and Phrases: Packing, Steiner Trees, Element-disjoint, Approximation algorithms, Hardness of approximation

1. INTRODUCTION

Throughout we assume that $G = (V, E)$, with $n = |V|$, is a simple graph and $T \subseteq V$ is a specified set of nodes (although we do not allow multi-edges, these can be handled by inserting new nodes into the edges). The nodes in T are called *terminal* nodes or *black* nodes, and the nodes in $V - T$ are called *Steiner* nodes or *white* nodes. Following the notation on approximation algorithms for graph connectivity problems (e.g., see [Jain et al. 1999]), by an *element* we mean either an edge or a Steiner node. A *Steiner tree* is a connected, acyclic subgraph that contains all the terminal nodes (Steiner nodes are optional). The problem of packing element-disjoint Steiner trees is to find a largest set of element-disjoint Steiner trees. In

A preliminary version of these results appeared in Proceedings of 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2005), Lecture Notes in Computer Science, Volume 3624, Springer 2005, pp 52-61.

Authors' addresses: Joseph Cheriyan, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L3G1, Canada. email:jcheriyan@math.uwaterloo.ca. Supported by NSERC grant No. OGP0138432.

Mohammad R. Salavatipour, Department of Computing Science, University of Alberta, Edmonton, Alberta T6G2E8, Canada. Email:mreza@cs.ualberta.ca. Supported by NSERC and a university start-up grant at the University of Alberta.

other words, the goal is to find the maximum number of Steiner trees such that each edge and each white node is in at most one of these trees. We denote this problem by **IUV**. Here, **I** denotes identical terminal sets for different trees in the packing, **U** denotes an undirected graph, and **V** denotes disjointness for white nodes and edges.

By *bipartite* **IUV** we mean the special case where G is a bipartite graph with node partition $V = T \cup (V - T)$, that is, one of the sets of the vertex bipartition consists of all of the terminal nodes. We also consider the problem of packing Steiner trees fractionally, with constraints on the nodes; this corresponds to a natural linear programming relaxation of **IUV**, and we call this problem *fractional* **IUV**; a detailed discussion is given later.

IUV captures some of the fundamental problems of combinatorial optimization and graph theory. First, suppose that T consists of just two nodes s and t. Then the problem is to find a largest set of element-disjoint s,t-paths. This problem is addressed by one of the cornerstone theorems in graph theory, namely Menger's theorem [Diestel 2000, Theorem 3.3.1], which states that the maximum number of openly-disjoint s,t-paths equals the minimum number of white nodes whose deletion leaves no s,t-path. The algorithmic problem of finding a largest set of openly-disjoint s,t-paths can be solved in polynomial time via any polynomial-time maximum s,t-flow algorithm. Another key special case of **IUV** occurs for $T = V$, that is, all the nodes are terminals. Then the problem is to find a largest set of edge-disjoint spanning trees. This problem is addressed by another classical min-max theorem, namely the Tutte/Nash-Williams theorem [Diestel 2000, Theorem 3.5.1]. The algorithmic problem of finding a largest set of edge-disjoint spanning trees can be solved in polynomial time via the matroid intersection algorithm. In contrast, the problem **IUV** is known to be NP-hard [Frank et al. 2003; Cheriyan and Salavatipour 2006], and the optimal value cannot be approximated within a factor of $\Omega(\log n)$ modulo the P≠NP conjecture [Cheriyan and Salavatipour 2006]; moreover, this hardness result applies to bipartite **IUV** and to fractional **IUV**.

One variant of **IUV** has attracted increasing research interest over the last few years, namely, the problem of packing edge-disjoint Steiner trees, that is, finding a largest set of edge-disjoint Steiner trees; we denote this problem by **IUE**. This problem in its full generality has applications in VLSI circuit design (e.g., see [Grotschel et al. 1997; Martin and Weismantel 1993]). Other applications include multicasting in wireless networks (see [Floréen et al. 2003]) and broadcasting large data streams, such as videos, over the Internet (see [Jain et al. 2003]). Almost a decade ago, Grötschel et al. studied the problem using methods from mathematical programming, in particular, polyhedral theory and cutting-plane algorithms, see [Grotschel et al. 1996c; 1996a; 1996d; 1996b; 1997]. Moreover, there is significant motivation from the areas of graph theory and combinatorial optimization, partly based on the relation to the classical results mentioned above, and partly fueled by an exciting conjecture of Kriesell [Kriesell 2003]. The conjecture states that the maximum number of edge-disjoint Steiner trees is at least half of the minimum number of edges in a cut that separates some pair of terminals. If this conjecture is settled by a constructive proof, then it may give a 2-approximation algorithm for **IUE**. Recently, Lau [Lau 2004] made a major advance on this conjecture by presenting a 26-approximation algorithm for **IUE** using new combinatorial ideas. Lau's construction is based on an earlier result of Frank, Kiraly, and Kriesell [Frank et al. 2003] that gives a 3-approximation for a special case of bipartite **IUV**.

Another related topic pertains to the domatic number of a graph and computing near-optimal domatic partitions. Feige et al. [Feige et al. 2002] presented approximation algorithms and hardness results for these problems; in particular, they presented algorithms with logarithmic approximation guarantees for the problem of finding a largest family of disjoint dominating sets. One of our key results (Theorem 3.1) is inspired by their work. In fact, the bipartite **IUV** problem is equivalent to the problem of packing one-sided connected dominating sets in bipartite graphs. It is well known that approximation algorithms for the minimum-cost dominating set problem extend to the minimum-cost connected dominating set problem with the same approximation guarantees (up to constant factors), see [Guha and Khuller 1999], but to the best of our knowledge, such extensions were not known for the corresponding packing problems.

Also, bipartite **IUV** is equivalent to the connected sub-hypergraph packing problem studied in [Bang-Jensen and Thomassé 2003]. It follows from [Bang-Jensen and Thomassé 2003] that there exist graphs that are $\Omega(\log n)$ element-connected on the terminal nodes but do not have two element-disjoint Steiner trees.

Frank et al. [Frank et al. 2003] studied bipartite **IUV**, and focusing on the restricted case where the degree of every white node is at most Δ they presented a Δ-approximation algorithm via the matroid intersection theorem and algorithm. The authors [Cheriyan and Salavatipour 2006] showed that (i) **IUV** is hard to approximate within a factor of $\Omega(\log n)$, even for bipartite **IUV** and even for the fractional version of bipartite **IUV**, (ii) **IUV** is APX-hard even if $|T|$ is a small constant, and (iii) there is an $O(\sqrt{n}\log n)$-approximation algorithm for a generalization of **IUV**. For **IUE**, Jain et al. [Jain et al. 2003] proved that the problem is APX-hard, and (as mentioned above) Lau [Lau 2004] presented a 26-approximation algorithm, based on the results of Frank et al. for bipartite **IUV**. [1]

Our main contribution is to settle the approximation guarantee for **IUV** and bipartite **IUV** up to constant factors. Moreover, our result extends to the capacitated version of **IUV**, where each white (Steiner) node v has a nonnegative integer capacity c_v, and the goal is to find a maximum collection of Steiner trees (allowing multiple copies of any Steiner tree) such that each white node v appears in at most c_v Steiner trees; there is no capacity constraint on the edges, i.e., each edge has infinite capacity. The capacitated version of **IUV** may be formulated as an integer program (IP) that has an exponential number of variables. Let $\mathcal{F}$ denote the collection of all Steiner trees in G. We have a binary variable x_F for each Steiner tree $F \in \mathcal{F}$.

[1] Although not relevant to this paper, we mention that the directed version of **IUV** has been studied [Cheriyan and Salavatipour 2006], and the known approximation guarantees and hardness lower bounds are within the same "ballpark" according to the classification of Arora and Lund [Arora and Lund 1996].

$$\begin{array}{ll} \text{maximize} & \sum_{F \in \mathcal{F}} x_F \\ \text{subject to} & \forall v \in V - T : \sum_{F: v \in F} x_F \leq c_v \\ & \forall F \in \mathcal{F} : \quad x_F \geq 0,\ x_F \in \mathbb{Z} \end{array} \tag{1}$$

Note that in uncapacitated **IUV** we have $c_v = 1, \forall v \in V - T$. The fractional **IUV** (mentioned earlier) corresponds to the linear programming relaxation of this IP that is obtained by relaxing the integrality condition on x_F's to $x_F \geq 0$.

Our main result is the following:

THEOREM 1.1. *(a) There is a polynomial-time probabilistic approximation algorithm with a guarantee of* $O(\log n)$ *and a failure probability of* $\frac{O(1)}{\log n}$ *for (uncapacitated)* **IUV**. *The algorithm finds a solution that is within a factor* $O(\log n)$ *of the optimal solution to fractional* **IUV**.
(b) The same approximation guarantee holds for capacitated **IUV**.

For two nodes s, t, let $\kappa(s,t)$ denote the maximum number of element-disjoint s,t-paths (an s,t-path means a path with end-nodes s and t); in other words, $\kappa(s,t)$ denotes the maximum number of s,t-paths such that each edge and each white node is in at most one of these paths. The graph is said to be *k-element connected* if $\kappa(s,t) \geq k,\ \forall s,t \in T, s \neq t$, i.e., there are $\geq k$ element-disjoint paths between every pair of terminals. For a graph $G = (V,E)$ and edge $e \in E$, $G - e$ denotes the graph obtained from G by deleting e, and G/e denotes the graph obtained from G by contracting e; see [Diestel 2000, Chapter 1] for more details. We call an edge white if both its end-nodes are white, otherwise, the edge is called black (then at least one end-node is a terminal). For our purposes, any edge can be subdivided by inserting a white node. In particular, any edge with both end-nodes black can be subdivided by inserting a white node. We call a graph *bipartite* if every edge is black; thus *bipartite* **IUV** means the special case of **IUV** where every edge is black.

Here is a sketch of our algorithm and proof for Theorem 1.1(a). Let k be the maximum number such that the input graph G is k-element connected. Clearly, the maximum number of element-disjoint Steiner trees is at most k (because each Steiner tree in a set of element-disjoint Steiner trees contributes one to the element connectivity). Note that this upper bound also holds for the optimal *fractional* solution. We delete or contract white edges in G, while preserving the element connectivity, to obtain a bipartite graph G^*; thus, G^* too is k-element connected (details in Section 2). Then we apply our key result (Theorem 3.1 in Section 3) to G^* to obtain $O(k/\log n)$ element-disjoint Steiner trees; this is achieved via a simple algorithm that assigns a random colour to each Steiner node – it turns out that for each colour, the union of T and the set of nodes with that colour induces a connected subgraph, and hence this subgraph contains a Steiner tree. Finally, we uncontract some of the white nodes to obtain the same number of element-disjoint Steiner trees of G. Clearly, uncontracting white nodes in a set of element-disjoint Steiner trees preserves the Steiner trees (up to the deletion of redundant edges) and preserves the element-disjointness of the Steiner trees.

As mentioned above, any edge can be subdivided by inserting a white node. Thus, the problem of packing element-disjoint Steiner trees can be transformed into the problem of packing Steiner trees that are disjoint on the set of white nodes. We prefer the formulation in terms of element-disjoint Steiner trees; for example, this formulation immediately shows that **IUV** captures the problem of packing edge-disjoint spanning trees; of course, the two formulations are equivalent.

2. REDUCING IUV TO BIPARTITE IUV

To prove our main result, we first show that the problem can be reduced to bipartite **IUV** while preserving the approximation guarantee. The next result is due to Hind and Oellermann [Hind and Oellermann 1996, Lemma 4.2]. We had found the result independently (before discovering the earlier works), and have included a proof for the sake of completeness.

THEOREM 2.1. *Given a graph $G = (V, E)$ with terminal set T that is k-element connected (and has no edge with both end-nodes black), there is a polynomial-time algorithm to obtain a bipartite graph G^* from G such that G^* has the same terminal set and is k-element connected, by repeatedly deleting or contracting white edges.*

PROOF. Consider any white edge $e = pq$. We prove that either deleting or contracting e preserves the k-element connectivity of G.

Suppose that $G - e$ is *not* k-element connected. Then by Menger's theorem $G - e$ has a set D of $k-1$ white nodes whose deletion "separates" two terminals. That is, every terminal is in one of two components of $G - D - e$ and each of these components has at least one terminal; call these two components C_p and C_q. Let s be a terminal in C_p and let t be a terminal in C_q. Let $\mathcal{P}(s, t)$ denote any set of k element-disjoint s, t-paths in G, and observe that one of these s, t-paths, say P_1, contains e (since the k-set $D \cup \{e\}$ "covers" $\mathcal{P}(s, t)$).

By way of contradiction, suppose that the graph $G'' = G/e$, obtained from G by contracting e, is *not* k-element connected. Then focus on G and note that, again by Menger's theorem, it has a set R of k white nodes, $R \supseteq \{p, q\}$, whose deletion "separates" two terminals. That is, there are two terminals that are in different components of $G - R$ (R is obtained by taking a "cut" of $k-1$ white nodes in G'' and uncontracting one node). This gives a contradiction because: (1) For s, t as above, the s, t-path P_1 in $\mathcal{P}(s, t)$ contains both nodes $p, q \in R$; since $|R| = k$ and $\mathcal{P}(s, t)$ has k element disjoint paths another one of the s, t-paths in $\mathcal{P}(s, t)$ say P_k is disjoint from R, by the Pigeonhole Principle; hence, $G - R$ has an s, t-path. (2) For terminals v, w that are both in C_p, $G - R$ has a v, t-path arguing as in (1) and also it has a w, t-path, thus $G - R$ has a v, w-path. (3) For terminals v, w that are both in C_q, $G - R$ has a v, s-path arguing as in (1) and also it has a w, s-path, thus $G - R$ has a v, w-path.

It is easy to complete the proof: we repeatedly choose any white edge and either delete e or contract e, while preserving the k-element connectivity, until no white edges are left; we take G^* to be the resulting k-element connected bipartite graph.

Clearly, this procedure can be implemented in polynomial time. In more detail, we choose any white edge e (if there exists one) and delete it. Then we compute whether or not the new graph is k-element connected by finding whether $\kappa(s, t) \geq k$ in the new graph for every pair of terminals s, t; this computation takes $O(k|T|^2|E|)$ time. If the new graph is k-element connected, then we proceed to the next white edge, otherwise, we identify the two end nodes of e (this has the effect of contracting

e in the old graph). Thus each iteration decreases the number of white edges (which is $O(|E|)$), hence, the overall running time is $O(k|T|^2|E|^2)$. $\square$

3. BIPARTITE **IUV**

This section has the key result of the paper, namely, a randomized $O(\log n)$-approximation algorithm for bipartite **IUV**.

THEOREM 3.1. *Given an instance of bipartite* **IUV** *such that the graph is k-element connected, there is a randomized polynomial-time algorithm that with probability $1 - \frac{1}{\log n}$ finds a set of $\Omega(\frac{k}{\log n})$ element-disjoint Steiner trees.*

PROOF. Without loss of generality, assume that the graph is connected, and there is no edge between any two terminals (if there exists any, then subdivide each such edge by inserting a Steiner node).

For ease of exposition, assume that n is a power of two and k is an integer multiple of $R = 6 \log n$; here, R is a parameter of the algorithm. The algorithm is simple: we color each Steiner node u.r. (uniformly at random) with one of $\frac{k}{R}$ super-colors $j = 1, \ldots, k/R$. For each $j = 1, \ldots, k/R$, let $\mathcal{D}^j$ denote the set of nodes that get the super-color j. We claim that for each j, the subgraph induced by $\mathcal{D}^j \cup T$ is connected with high probability, and hence this subgraph contains a Steiner tree. If the claim holds, then we are done, since we get a set of k/R element-disjoint Steiner trees.

For the purpose of analysis, it is easier to present the algorithm in an equivalent form that has two phases. In phase one, we color every Steiner node u.r. with one of k colors $i = 1, \ldots, k$ and we denote the set of nodes that get the color i by C^i $(i = 1, \ldots, k)$. In phase two, we partition the color classes into k/R super-classes where each super-class $\mathcal{D}^j$ $(j = 1, \ldots, k/R)$ consists of R consecutive color classes $C_{(j-1)R+1}, C_{(j-1)R+2}, \ldots, C_{jR}$. We do this in R rounds, where in round $1 \le \ell \le R$ we have $\mathcal{D}^j_\ell = \bigcup_{i=(j-1)R+1}^{(j-1)R+\ell} C^i$; thus we have $\mathcal{D}^j = \mathcal{D}^j_R$. Consider an arbitrary super-class $\mathcal{D}^j$, $1 \le j \le k/R$. For an arbitrary ℓ, $1 \le \ell \le R$, let H_ℓ denote the graph induced by $\mathcal{D}^j_\ell \cup T$, and let $d_\ell \ge 1$ denote the number of connected components of H_ℓ.

LEMMA 3.2. *Let H_ℓ and d_ℓ be as defined above, and suppose that $d_\ell > 1$, i.e., H_ℓ is not connected. Let $G_1, \ldots, G_{d_\ell}$ be the connected components of H_ℓ. Consider any connected component of H_ℓ, say G_a $(1 \le a \le d_\ell)$. Then there is a set $U \subseteq V - T - V(G_a)$ (of white nodes) with $|U| \ge k$ such that each node in U is adjacent to a terminal in G_a and to a terminal in $G - V(G_a)$.*

PROOF. Let $U \subseteq V - V(G_a)$ be a maximum-size set of Steiner nodes such that each node in U has a neighbour in each of G_a and $G - V(G_a)$; note that none of the nodes in U is in G_a. By way of contradiction, assume that $|U| < k$. Consider $G - U$. Note that there is at least one terminal in each of G_a and $G - U - V(G_a)$. An important observation is that every edge of G between G_a and $G - V(G_a)$ is between a terminal of G_a and a Steiner node of $G - V(G_a)$; this holds because G is bipartite and G_a is a subgraph induced by T and some set of white nodes. From this, and by definition of U, there is no edge between G_a and $G - U - V(G_a)$, i.e., $G - U$ is disconnected. This contradicts the assumption that G is k element-connected. $\square$

Consider a set U as in the above lemma. If a vertex $s \in U$ has the color $\ell + 1$, then when we add $C^{\ell+1}$ to $\mathcal{D}^j_\ell$, we see that s connects G_a and another connected component of H_ℓ, because s is adjacent to a terminal in G_a and to a terminal in $G - V(G_a)$. For every node $s \in U$ we have $\Pr[s \in C^{\ell+1}] = \frac{1}{k}$. Thus, the probability that none of the vertices in U has been colored $\ell + 1$ is at most:

$$\left(1 - \frac{1}{k}\right)^{|U|} \leq \left(1 - \frac{1}{k}\right)^{k} \leq e^{-1}. \tag{2}$$

This is an upper bound on the probability that when we add $C^{\ell+1}$ to $\mathcal{D}^j_\ell$, component G_a *does not* become connected to another connected component G_b, for some $1 \leq b \leq d_\ell, b \neq a$. Note that G_a is an arbitrary component of H_ℓ. If every connected component G_i, $1 \leq i \leq d_\ell$, becomes connected to another component, then the number of connected components of H_ℓ decreases to at most $\frac{d_\ell}{2}$ in round $\ell + 1$. If in every round and for every super-class, the number of connected components decreases by a constant factor then, after $O(\log n)$ rounds, every $\mathcal{D}^j \cup T$ forms a connected graph. We show that this happens with sufficiently high probability.

By (2), in round ℓ, any fixed connected component of H_ℓ becomes connected to another component with probability at least $1 - e^{-1}$. So the expected number of connected components of H_ℓ that become connected to another component is $(1 - e^{-1}) \cdot d_\ell$. Thus, if $d_\ell \geq 2$ then defining $\sigma = \frac{1+e^{-1}}{2}$ we have:

$$\mathrm{E}[d_{\ell+1} \mid d_\ell] \leq \sigma \cdot d_\ell. \tag{3}$$

Define $X_\ell = d_\ell - 1$. Therefore, $X_1, X_2, \ldots, X_\ell, \ldots,$ is a sequence of integer random variables that starts with $X_1 = d_1 - 1$. Moreover, for every $\ell \geq 1$, we have $X_\ell \geq 0$, and if $X_\ell = 0$ then $\mathrm{E}[X_{\ell+1}] = 0$, and if $X_\ell \geq 1$ then

$$\begin{aligned}
\mathrm{E}[X_{\ell+1}|X_\ell] &= \mathrm{E}[d_{\ell+1} - 1 | d_\ell - 1 \geq 1] \\
&= \mathrm{E}[d_{\ell+1} | d_\ell \geq 2] - 1 \\
&\leq \sigma d_\ell - 1 \qquad\qquad \text{by (3)} \\
&= \sigma X_\ell + \sigma - 1 \\
&\leq \sigma X_\ell.
\end{aligned}$$

An easy induction shows that $\mathrm{E}[X_{\ell+1}] \leq \sigma^\ell X_1$. Since $X_1 \leq n - 1$ and $\sigma < \frac{3}{4}$, we have $\mathrm{E}[X_R] \leq \frac{1}{n}$ (recall that $R = 6 \log n$). Therefore, Markov's inequality implies that $\Pr[X_R \geq 1] \leq \frac{1}{n}$. This implies that $\Pr[d_R \geq 2] \leq \frac{1}{n}$, i.e., the probability that $H_R = D^j \cup T$ is *not* connected is at most $\frac{1}{n}$. As there are $\frac{k}{R}$ super-classes, a simple union-bound shows that the probability that there is at least one D^j $(1 \leq j \leq \frac{k}{R})$ such that $D^j \cup T$ is not connected is at most $\frac{k}{Rn} \leq \frac{1}{\log n}$. Thus, with probability at least $1 - \frac{1}{\log n}$, every super-class D^j (together with T) induces a connected graph, and hence, the randomized algorithm finds $\Omega(k/\log n)$ element-disjoint Steiner trees.

Remark: To get an approximation guarantee of $O(\log |T|)$ or $O(\log k)$ instead of $O(\log n)$, one way may be to replace the parameter $R = 6 \log n$ by an appropriate function of $|T|$ or k; but then the claim $\Pr[X_R \geq 1] \leq \frac{1}{n}$ (in the last paragraph of the proof) fails to hold; thus the failure probability may be high.

4. IUV AND CAPACITATED IUV

Now we complete the proof of Theorem 1.1 using Theorems 2.1 and 3.1.

First, we prove part (a). Let k be the maximum number such that the input graph G is k-element connected. Clearly, the maximum number of element-disjoint Steiner trees is at most k. Apply Theorem 2.1 to obtain a bipartite graph G^* that is k-element connected. Apply Theorem 3.1 to find $\Omega(\frac{k}{\log n})$ element-disjoint Steiner trees in G^*. Then uncontract white nodes to obtain the same number of element-disjoint Steiner trees of G. Moreover, it can be seen that the optimal value of the LP relaxation is at most k (because there exists a set of k white nodes whose deletion leaves no path between some pair of terminals). Thus our integral solution is within a factor $O(\log n)$ of the optimal fractional solution.

Now, we prove part (b) of Theorem 1.1, by using ideas from [Cheriyan and Salavatipour 2006; Jain et al. 2003; Lau 2004]. Consider the IP formulation (1) of capacitated **IUV**. The *fractional capacitated* **IUV** problem is the linear program (LP) obtained by relaxing the integrality condition in the IP to $x_F \geq 0$. As we said earlier, this LP has exponentially many variables, however, we can solve it approximately. Then we show that either rounding the approximate LP solution will result in an $O(\log n)$-approximation or we can reduce the problem to the uncapacitated version of **IUV** and use Theorem 1.1(a).

Note that the separation oracle for the dual of the LP is the problem of finding a minimum node-weighted Steiner tree. Using this fact, the proof of Theorem 4.1 in [Jain et al. 2003] may be adapted to prove the following:

LEMMA 4.1. *There is an α-approximation algorithm for fractional capacitated* **IUV** *if and only if there is an α-approximation algorithm for the minimum node-weighted Steiner tree problem.*

Klein and Ravi [Klein and Ravi 1995] (see also Guha and Khuller [Guha and Khuller 1999]) give an $O(\log n)$-approximation algorithm for the problem of computing a minimum node-weighted Steiner tree. Their result, together with Lemma 4.1, implies the next lemma. The lemma also follows from [Carr and Vempala 2002].

LEMMA 4.2. *There is a polynomial-time $O(\log n)$-approximation algorithm for fractional capacitated* **IUV**.

Define φ and φ_f to be the optimal objective values for capacitated **IUV** and for fractional capacitated **IUV**, respectively. Consider an approximately optimal solution to fractional capacitated **IUV** obtained by Lemma 4.2. Let φ^* denote the approximately optimal objective value, and let $Y = \{x_1, \ldots, x_d\}$ denote the set of primal variables that have positive values. One of the features of the algorithm of Lemma 4.2 (which is also a feature of the algorithm of [Jain et al. 2003]) is that d (the number of fractional Steiner trees computed) is polynomial in n, even though the LP has an exponential number of variables. If $\sum_{i=1}^{d} \lfloor x_i \rfloor \geq \frac{1}{2} \sum_{i=1}^{d} x_i$ then $Y' = \{\lfloor x_1 \rfloor, \ldots, \lfloor x_d \rfloor\}$ is an integral solution (i.e., a solution for capacitated **IUV**) with value at least $\frac{\varphi^*}{2}$, which is at least $\Omega(\frac{\varphi_f}{\log n})$, and this in turn is at least

$\Omega(\frac{\varphi}{\log n})$. In this case the algorithm returns the Steiner trees corresponding to the variables in Y' and stops. This is within an $O(\log n)$ factor of the optimal solution. Otherwise, if $\sum_{i=1}^{d} \lfloor x_i \rfloor < \frac{1}{2}\sum_{i=1}^{d} x_i$ then

$$\varphi^* = \sum_{i=1}^{d} x_i = \sum_{i=1}^{d} \lfloor x_i \rfloor + \sum_{i=1}^{d} (x_i - \lfloor x_i \rfloor) < \frac{\varphi^*}{2} + d.$$

Therefore $\varphi^* < 2d$. This implies that for every Steiner node v, at most a value of $\min\{c_v, O(d \log n)\}$ of the capacity of v is used in any optimal (fractional or integral) solution. So we can decrease the capacity c_v of every Steiner node $v \in V - T$ to $\min\{c_v, O(d \log n)\}$. Note that this value is upper bounded by a polynomial in n. Let this new graph be G'. We are going to modify this graph to another graph G'' which will be an instance of uncapacitated **IUV**. For every Steiner node $v \in G'$ with capacity c_v we replace v with c_v copies of it called $v_1, \ldots, v_{c_v}$ each having unit capacity. The set of terminal nodes stays the same in G' and G''. Then for every edge $uv \in G'$ we create a complete bipartite graph on the copies of v (as one part) and the copies of u (the other part) in G''. This new graph G'' will be the instance of uncapacitated **IUV**. It follows that the size of G'' is polynomial in G. Also, it is straightforward to verify that G'' has α element-disjoint Steiner trees if and only if there are α Steiner trees in G satisfying the capacity constraints of the Steiner nodes. Finally, we apply the algorithm of Theorem 1.1(a) to graph G''.

A referee suggested the following modification of the proof for part (b) where the input data (capacities of Steiner nodes) is perturbed to give an "approximately equivalent" uncapacitated problem. Let k denote the element connectivity between terminals with respect to the Steiner node capacities; k may be computed via an algorithm for maximum flows. Replace c_v by $\min\{c_v, k\}$, for each Steiner node v; this does not change the optimal value. We may assume $k \geq n$, otherwise, as in the first proof, we find a near-optimal solution by using the algorithm for part (a). Next, delete each Steiner node v with $c_v \leq \frac{k}{2n}$. This decreases the element connectivity, but it stays $\geq \frac{k}{2}$, since the deletion of each such node decreases the element connectivity by at most $\frac{k}{2n}$. Now the node capacities are all within a factor at most $2n$ of each other. Let $c_{\min}$ be the smallest capacity of a remaining Steiner node. For each Steiner node v, let $c'_v = \lfloor c_v / c_{\min} \rfloor$. Now all the c'_v values are in the range $\{1, \ldots, 2n\}$. As in the first proof, we find a near-optimal solution by using the algorithm for part (a), after replacing each Steiner node v with c'_v copies and replacing each edge with a complete bipartite graph. At the end, we multiply each tree by $c_{\min}$ to obtain a solution to the original problem that is within a factor $O(\log n)$ of the optimum.

5. CONCLUDING REMARKS

Although **IUE** seems to be more natural compared to **IUV**, and although there are many more papers (applied, computational, and theoretical) on **IUE**, the only known $O(1)$-approximation guarantee for **IUE** is based on solving bipartite **IUV**. This shows that **IUV** is a fundamental problem in this area. We presented a simple randomized algorithm for **IUV** that finds an integral solution that is within a factor $O(\log n)$ of the optimal integral (and in fact optimal fractional) solution.

Acknowledgments: We thank David Kempe, whose comments led to an improved analysis in Theorem 3.1, and Lap Chi Lau who brought reference [Hind and Oellermann 1996] to our attention. We thank the referees for their comments.

REFERENCES

Arora, S. and Lund, C. 1996. Hardness of approximations. In *Approximation Algorithms for NP-hard Problems*, D. Hochbaum, Ed. PWS Publishing.

Bang-Jensen, J. and Thomassé, S. 2003. Highly connected hypergraphs containing no two edge-disjoint spanning connected subhypergraphs. *Discrete Applied Mathematics 131,* 2, 555–559.

Carr, B. and Vempala, S. 2002. Randomized meta-rounding. *Random Structures and Algorithms 20,* 3, 343–352.

Cheriyan, J. and Salavatipour, M. 2006. Hardness and approximation results for packing steiner trees. *Algorithmica 45,* 1, 21–43.

Diestel, R. 2000. *Graph Theory.* Springer, New York, NY.

Feige, U., Halldorsson, M., Kortsarz, G., and Srinivasan, A. 2002. Approximating the domatic number. *SIAM J.Computing 32,* 1, 172–195.

Floréen, P., Kaski, P., Kohonen, J., and Orponen, P. 2003. Multicast time maximization in energy constrained wireless networks. In *Joint Workshop on Foundations of Mobile Computing, DIALM-POMC*.

Frank, A., Király, T., and Kriesell, M. 2003. On decomposing a hypergraph into k connected sub-hypergraphs. *Discrete Applied Mathematics 131,* 2, 373–383.

Grotschel, M., Martin, A., and Weismantel, R. 1996a. Packing steiner trees: a cutting plane algorithm. *Math. Prog. 72,* 2, 125–145.

Grotschel, M., Martin, A., and Weismantel, R. 1996b. Packing steiner trees: further facets. *European J. Combinatorics 17,* 1, 39–52.

Grotschel, M., Martin, A., and Weismantel, R. 1996c. Packing steiner trees: polyhedral investigations. *Math. Prog. 72,* 2, 101–123.

Grotschel, M., Martin, A., and Weismantel, R. 1996d. Packing steiner trees: separation algorithms. *SIAM J. Disc. Math. 9*, 233–257.

Grotschel, M., Martin, A., and Weismantel, R. 1997. The steiner tree packing problem in vlsi design. *Mathematical Programming 78*, 265–281.

Guha, S. and Khuller, S. 1999. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and Computation 150*, 57–74.

Hind, H. R. and Oellermann, O. 1996. Menger-type results for three or more vertices. *Congressus Numerantium 113*, 179–204.

Jain, K., Mahdian, M., and Salavatipour, M. 2003. Packing steiner trees. In *Proceedings of the 14th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 266–274.

Jain, K., Mandoiu, I., Vazirani, V., and Williamson, D. 1999. A primal-dual schema based approximation algorithm for the element connectivity problem. In *Proceedings of 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 99–106.

Klein, P. and Ravi, R. 1995. A nearly best-possible approximation algorithm for node-weighted steiner trees. *Journal of Algorithms 19*, 104–115.

Kriesell, M. 2003. Edge-disjoint trees containing some given vertices in a graph. *J. Combinatorial Theory (B) 88*, 53–65.

Lau, L. 2004. An approximate max-steiner-tree-packing min-steiner-cut theorem. In *Proc. of 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 61–70.

Martin, A. and Weismantel, R. 1993. Packing paths and steiner trees: Routing of electronic circuits. *CWI Quarterly*, 185–204.