Hardness and Approximation Results for Packing Steiner Trees

Joseph Cheriyan*
Department of Combinatorics and Optimization,
University of Waterloo, Waterloo, Canada N2L 3G1
jcheriya@math.uwaterloo.ca

Mohammad R. Salavatipour †
Department of Computing Science,
University of Alberta, Edmonton, Alberta, Canada T6G 2E8
mreza@cs.ualberta.ca

Abstract

We study approximation algorithms and hardness of approximation for several versions of the problem of packing Steiner trees. For packing edge-disjoint Steiner trees of undirected graphs, we show APX-hardness for 4 terminals. For packing Steiner-node-disjoint Steiner trees of undirected graphs, we show a logarithmic hardness result, and give an approximation guarantee of $O(\sqrt{n}\log n)$, where n denotes the number of nodes. For the directed setting (packing edge-disjoint Steiner trees of directed graphs), we show a hardness result of $\Omega(m^{\frac{1}{3}-\epsilon})$ and give an approximation guarantee of $O(m^{\frac{1}{2}+\epsilon})$, where m denotes the number of edges. We have similar results for packing Steiner-node-disjoint priority Steiner trees of undirected graphs.

1 Introduction

We study approximation algorithms and hardness of approximation for several versions of the problem of packing Steiner trees. Given an undirected graph $G = (V, E)$ and a set of *terminal* nodes $T \subseteq V$, a *Steiner tree* is a connected, acyclic subgraph that contains all the terminal nodes (nonterminal nodes, which are called *Steiner nodes*, are optional). The basic problem of packing edge-disjoint undirected Steiner trees (**IUE**-unitcap for short) is to find as many edge-disjoint Steiner trees as possible. Besides **IUE**-unitcap, we study some other versions; see below for details. All of the Steiner tree packing problems discussed in this paper are NP-hard, although some special cases may have polynomial-time algorithms.

The **IUE**-unitcap problem in its full generality (called **GUE**, see below) has applications in VLSI circuit design (e.g., see [13, 22]). Other applications include multicasting in wireless networks (see [9]) and broadcasting large data streams, such as videos, over the Internet (see [16]). There is significant motivation

*Supported by NSERC grant No. OGP0138432.
†Supported by an NSERC postdoctoral fellowship, Dept. of Combinatorics and Optimization at University of Waterloo, and a University start-up fund at University of Alberta.

from the areas of graph theory and combinatorial optimization. Menger's theorem on packing edge-disjoint s, t-paths [7] corresponds to the special case of packing edge-disjoint Steiner trees on two terminal nodes (i.e., $T = \{s, t\}$). Another special case is when all the nodes are terminals (i.e., $T = V$). Then the problem is to find a maximum set of edge-disjoint spanning trees. This topic was studied in the 1960's by graph theorists, and a min-max theorem was developed by Tutte and independently by Nash-Williams [7]. Subsequently, Edmonds and Nash-Williams derived such results in the more general setting of the matroid intersection theorem. One consequence is that efficient algorithms are available via the matroid intersection algorithm for the case of $T = V$. A set of nodes S is said to be λ-*edge connected* if there exist λ edge-disjoint paths between every two nodes of S. An easy corollary of the min-max theorem of Nash-Willams and Tutte is that if the node set V is $2k$-edge connected, then the graph has k edge-disjoint spanning trees. Recently, Kriesell [20] conjectured an exciting generalization: If the set of terminals is $2k$-edge connected, then there exist k edge-disjoint Steiner trees. He proved this for Eulerian graphs by an easy application of the splitting-off theorem together with the min-max theorem of Nash-Willams and Tutte. Note that a constructive proof of this conjecture may give a 2-approximation algorithm for **IUE**-unitcap. Jain, Mahdian, and Salavatipour [16] gave an approximation algorithm with guarantee (roughly) $\frac{|T|}{4}$. Moreover, using a versatile and powerful proof technique (that we will borrow and apply in the design of our algorithms), they showed that the fractional version of **IUE**-unitcap has an α-approximation algorithm if and only if the minimum-weight Steiner tree problem has an α-approximation algorithm (Theorem 4.1 in [16]). The latter problem is well studied and is known to be APX-hard. It follows that **IUE**-unitcap is APX-hard (Corollary 4.3 in [16]). Frank et al. [10] gave a 3-approximation algorithm for the special case of **IUE**-unitcap where no two Steiner nodes are adjacent. Very recently Lau [21], based on the result of Frank et al. [10], has given an $O(1)$-approximation algorithm for **IUE**-unitcap (but Kriesell's conjecture remains open).

Several of our proof techniques are inspired by results for disjoint-paths problems in the papers by Guruswami et al. [11], Baveja and Srinivasan [2], and Kolliopoulos and Stein [19]. In these problems, we are given a graph and a set of source-sink pairs, and the goal is to find a maximum set of edge/node disjoint source-sink paths. Although there is no direct relation between Steiner tree packing problems and disjoint-paths problems (neither problem is a special case of the other one) there is a common generalization of both these problems, namely, the problem of packing Steiner trees with different terminal sets: given ℓ (not necessarily disjoint) sets of terminals $T_1, T_2, \ldots, T_\ell$, ℓ polynomial in n, find a maximum set of edge-disjoint Steiner trees such that each tree contains one of the terminal sets $T_1, T_2, \ldots, T_\ell$. Also, see Carr and Vempala [3] and Vempala and Vöcking [23] for results on multicast congestion.

We use n and m to denote the number of nodes and edges, respectively. The underlying assumption for most of our hardness results is P$\neq$NP. Throughout, we use ϵ to denote any small positive real number. We denote some of the versions of the Steiner tree packing problem by three-letter abbreviations. The first letter is either I or G, and denotes whether or not all Steiner trees have identical terminal sets (e.g., the letter I in **IUE**-unitcap). The second letter is either D or U, and denotes whether the graph is directed or undirected. The third letter is either E or V, and denotes whether the Steiner trees in the packing are edge disjoint or vertex disjoint. Note that here and elsewhere, we use "vertex disjoint Steiner trees" to mean trees that are disjoint on the *Steiner vertices* (and of course they contain all the terminal vertices). In fact, we associate nonnegative integer-valued capacities with the edges or the vertices, and a feasible packing of Steiner trees is one that satisfies the capacity constraint of every edge or of every Steiner vertex. We denote the special case where all capacities are one by appending "unitcap" to the abbreviation. We discuss some other versions of the Steiner tree packing problem too. Most of our hardness (of approximation) results are presented for the most specialized version from the relevant family of problems (e.g., Theorem 2.4 pertains to the special case of **IDE**-unitcap, namely, packing directed edge-disjoint Steiner trees), and thus we immediately get the same hardness result for all of the problems in the relevant family (e.g., Theorem 2.4 implies the same hardness result for **GDE**); but better hardness results may be known for the most general problem in the relevant family (e.g., **GDE** contains the problem of packing edge-disjoint paths in directed graphs, for which a hardness lower bound of $\Omega(m^{\frac{1}{2}-\epsilon})$ is known [11], hence this lower bound applies to **GDE**). Most of our results on approximation algorithms and guarantees pertain to the most general version in the relevant family of problems, and thus we immediately get the same approximation guarantees for all of the problems in the relevant family, though better approximation guarantees may be known for some specialized problems in the relevant family.

Consider the problems **IUV**, **GUE**, **GUV**-priority (to be defined later), and their special cases. For **IUV**, we are given an undirected graph G, a set $T \subseteq V$ of terminals, and a nonnegative vertex capacity c_v for each Steiner vertex v. We assume that there are no edges between terminal nodes (i.e., T is an independent set of G); this assumption may be enforced by subdividing each edge between two terminals by inserting a distinct Steiner vertex with unit capacity. The goal is to find the maximum number of Steiner trees such that each Steiner vertex v is contained in $\leq c_v$ trees. The problem **GUE** is the generalization where the instance has ℓ terminal sets $T_1, \ldots, T_\ell$ (where ℓ is polynomial in n) and the goal is to find the maximum number of Steiner trees, such that each Steiner tree contains one of the terminal sets $T_1, \ldots, T_\ell$, and such that each Steiner vertex v is contained in $\leq c_v$ trees. For this and other problems on packing (directed or undirected) vertex-capacitated Steiner trees with multiple terminal sets $T_1, \ldots, T_\ell$, our assumption is that each Steiner tree H has an associated index $i \in \{1, \ldots, \ell\}$ such that H contains T_i, and any vertex of $V - T_i$ may be present in H as a Steiner vertex; thus a Steiner tree with terminal set T_i may contain vertices from $(T_1 \cup \ldots \cup T_\ell) - T_i$ as Steiner vertices. Using the fact that **IUE**-unitcap is APX-hard, Floréen et al. [9] showed that **IUV**-unitcap is APX-hard. They raised the question whether this problem is in the class APX. We prove an $\Omega(\log n)$-hardness result (lower bound) for **IUV**-unitcap. This shows that **IUV**-unitcap is significantly harder than **IUE**-unitcap, and settles (in the negative) the open question of Floréen et al. [9]. We give an approximation guarantee (upper bound) of $O(\sqrt{n} \log n)$ for **GUV**, by an LP-based rounding algorithm. We study another natural generalization of **IUV**, namely *Packing undirected vertex-capacitated priority Steiner trees* (**IUV**-priority for short), which is motivated by the Quality of Service in network design problems (see [5] for applications). For **IUV**-priority, we show a lower-bound of $\Omega(n^{\frac{1}{3}-\epsilon})$ on the approximation guarantee; moreover, our approximation algorithm for **GUV** extends to **GUV**-priority to give a guarantee of $O(n^{\frac{1}{2}+\epsilon})$. We mention that a hardness lower bound of $\Omega(n^{\frac{1}{2}-\epsilon})$ is given in [11, Theorem 2] for another special case of **GUV**-priority, namely, the problem of packing vertex-disjoint priority s_i, t_i paths.

Now, consider a directed graph $G(V, E)$ with a positive capacity c_e for each edge e, a set $T \subseteq V$ of terminals, and a specified root vertex r, $r \in T$. A *directed Steiner tree* rooted at r is a rooted subtree of G that contains a directed path from r to t, for each terminal $t \in T$. In the problem of packing directed edge-capacitated Steiner trees (**IDE**) the goal is to find the maximum number of directed Steiner trees rooted at r such that each edge e is contained in $\leq c_e$ directed trees. The problem **GDE** is the generalization where the instance has ℓ terminal sets $T_1, \ldots, T_\ell$ and ℓ roots $r_1, \ldots, r_\ell$ (where ℓ is polynomial in n, and $r_i \in T_i$, $i = 1, \ldots, \ell$), and the goal is to find the maximum number of directed Steiner trees, each rooted at an r_i and containing all the nodes in T_i (for an $i = 1, \ldots, \ell$), such that each edge e is contained in $\leq c_e$ directed

Table 1: Summary of results

Problem	Approx.Guarantee	Hardness	Hardness for smallparameters
IUE-unitcap	26[21]	APX-hard[16]	APX-hard for 4 terminals (T3.3) NP-hard for 2 trees[18]
GUE **IUV** **IUV**-unitcap	$O(\log n\sqrt{n})$ (T3.11) $O(\log^2 n)$[6]	 $\Omega(\log n)$-hard (T3.9)	 APX-hard for 4 terminals (T3.8) NP-hard for 2 trees (T3.2)
GUV-priority **IUV**-priority	$O(n^{\frac{1}{2}+\epsilon})$ (T3.12)	$\Omega(n^{\frac{1}{2}-\epsilon})$[11] $\Omega(n^{\frac{1}{3}-\epsilon})$-hard (T3.15)	
GDE **IDE**-unitcap	$O(m^{\frac{1}{2}+\epsilon})$ (T2.15)	$\Omega(m^{\frac{1}{2}-\epsilon})$[11] $\Omega(m^{\frac{1}{3}-\epsilon})$ (T2.8)	 NP-hard for 3 terminals and 2 trees (T2.3)
GDV **IDV**-unitcap	$O(n^{\frac{1}{2}+\epsilon})$ (T2.16)	$\Omega(n^{\frac{1}{2}-\epsilon})$[11] $\Omega(n^{\frac{1}{3}-\epsilon})$ (T2.4)	 NP-hard for 3 terminals and 2 trees (T2.2)

trees. As mentioned above, one special case of **GDE** is the problem of packing edge-disjoint paths in a directed graph, and a hardness lower bound of $\Omega(m^{\frac{1}{2}-\epsilon})$ is given in [11]. We prove an $\Omega(m^{\frac{1}{3}-\epsilon})$-hardness result for **IDE**-unitcap. Moreover, we give an approximation algorithm with a guarantee of $O(m^{\frac{1}{2}+\epsilon})$ for **GDE**. We also consider the problem of packing directed vertex-capacitated Steiner trees (**IDV** and **GDV**), where instead of capacities on the edges we have a capacity c_v on every Steiner vertex v, and the goal is to find the maximum number of directed Steiner trees such that each Steiner vertex v is contained in $\leq c_v$ directed trees. For directed graphs, **IDE** and **IDV** (and also **GDE** and **GDV**) are similar, see Theorem 2.1. We get a lower-bound of $\Omega(n^{\frac{1}{3}-\epsilon})$ on the approximation guarantee for **IDV**-unitcap. Moreover, we give an approximation algorithm with a guarantee of $O(n^{\frac{1}{2}+\epsilon})$ for **GDV**.

We now focus on hardness (of approximation) results for several versions of the problem of packing Steiner trees (with identical terminal sets) when some of the key parameters are small. In particular, we discuss problems where the number of terminals is small, meaning $|T| = O(1)$, and also problems where the optimal value is small, meaning the number of Steiner trees in an optimal packing is either one or two. Kaski [18] showed that the problem **IUE**-unitcap is NP-hard even if the number of terminals is 7, and moreover, the problem of finding two edge-disjoint Steiner trees is NP-hard. Floréen et al. [9] showed that the special case of the problem **IUV**-unitcap with only 4 terminal nodes is NP-hard. Our hardness results for small-parameter problems are as follows.

• *Packing undirected edge-disjoint Steiner trees (***IUE***-unitcap)*: We show that the special case of the problem with four terminal nodes is APX-hard. (An early draft of our paper proved, independently of [18], that finding two edge-disjoint Steiner trees is NP-hard.)

• *Packing undirected vertex-disjoint Steiner trees (***IUV***-unitcap)*: We show essentially the same hardness results for **IUV**-unitcap as for **IUE**-unitcap, that is, the special case of the problem with four terminal nodes is APX-hard, and the problem of finding two vertex-disjoint Steiner trees is NP-hard.

• *Packing directed edge-disjoint Steiner trees (***IDE***-unitcap):* We show that the problem of finding two edge-disjoint directed Steiner trees with only three terminal nodes is NP-hard.

Table 1 summarizes the results of this paper and the previous works [9, 11, 16, 18, 21]; results from this paper are cited by theorem number, and results from other papers are indicated by citing the paper. Very recently, we have obtained a randomized $O(\log^2 n)$ approximation algorithm for **IUV**, using different methods from the ones used in this paper [6]; this result will appear elsewhere.

For the problems **IUV**, **IUV**-priority, **IDE**, and **IDV**, we are not aware of any previous results on approximation algorithms or hardness results other than [9], although there is extensive literature on approximation algorithms for the corresponding *minimum-weight* Steiner tree problems (e.g., [4] for minimum-weight directed Steiner trees and [14] for minimum-node-weighted Steiner trees).

Section 2 has our results on directed graphs for problems **GDE**, **GDV**, and their special cases. Section 3 has our results on undirected graphs for problems **IUE**, **GUE**, **GUV**-priority, and their special cases.

2 Packing Directed Steiner Trees

In this section, we study the problem of packing directed steiner trees. We start with an auxiliary result: The problems **IDE** and **IDV** are equivalent in the sense that there is a polynomial-time reduction from either problem to the other problem that preserves the optimal value (number of Steiner trees in an optimal packing). Then we present hardness results for **IDE**-unitcap (i.e., edge-disjoint directed case), and these immediately imply similar hardness results for **IDV**-unitcap (i.e., directed vertex-disjoint version). We also present an approximation algorithm for **GDE** which implies a similar approximation algorithm for **GDV**. The proof of the following theorem is easy. The idea for the first direction is to insert a new node in every edge, and for the second one is to split every vertex into two adjacent vertices.

Theorem 2.1 *Given an instance $I = (G(V,E), T \subseteq V, k)$ of* **IDE** *(of* **IDV***), there is an instance $I' = (G'(V',E'), T' \subseteq V, k)$ of* **IDV** *(of* **IDE***) with $|G'| = poly(|G|)$, such that I has k directed Steiner trees satisfying the capacities of the edges (vertices) if and only if I' has k directed Steiner trees satisfying the capacities of the vertices (edges). The same statement holds for* **GDE** *and* **GDV***.*

Proof: (1st direction)
We insert a new Steiner node v_{xy} in every edge xy, and we fix the capacity of v_{xy} (in G') to be the same as the capacity of xy (in G). All the other Steiner nodes of G' (corresponding to Steiner nodes of G) get infinite capacities. The root and the other terminals are the same in G and G'. It can be seen that G has k (directed) Steiner trees satisfying edge capacities if and only if G' has k (directed) Steiner trees satisfying vertex capacities.

(2nd direction)
We construct G' from G in the following way. For each node $v \in V$, G' contains two nodes v_1, v_2. If $v \in T$ then both v_1 and v_2 become terminals in G', and if $r \in T$ is the root then r_1 becomes the root in G'. We add v_1v_2 to E' and give it the same capacity as vertex v in G. If $v \in T$, then we give infinite capacity to v_1v_2. Furthermore, for every edge $uv \in E$ we create an edge u_2v_1 (with infinite capacity) in E' and for every edge $vw \in E$ we create an edge v_2w_1 (with infinite capacity) in E'.

It is easy to see that if $\mathcal{T}$ is a collection of k Steiner trees in G that satisfy vertex capacities then there is a collection $\mathcal{T}'$ of k Steiner trees in G' that satisfy edge capacities. Conversely, suppose that $\mathcal{T}'$ is a collection of k Steiner trees in G' satisfying edge capacities. Then for every edge v_1v_2 (corresponding to a vertex $v \in V(G)$ with capacity c_v in G) there are at most c_v trees containing that edge. Therefore, by contracting the edges of the form v_1v_2 on each tree of $\mathcal{T}'$ we obtain a collection of k Steiner trees in G such that for every vertex v there are at most c_v trees containing it. ∎

2.1 Hardness results

First we prove that **IDV**-unitcap is NP-hard even in the simplest non-trivial case where there are only three terminals (one root r and two other terminals) and we are asked to find only 2 vertex-disjoint Steiner trees. The problem becomes easy if any of these two conditions is tighter, i.e., if the number of terminals is reduced to 2 or the number of Steiner trees that we have to find is reduced to 1. If the number of terminals is arbitrary, then we show that **IDV**-unitcap is NP-hard to approximate within a factor of $O(n^{\frac{1}{3}-\epsilon})$ for any $\epsilon > 0$. The proof does *not* rely on the PCP theorem. Also, as we mentioned before, both of these hardness results carry over to **IDE**. For both reductions, we use the following well-known NP-hard problem (see [12]):

PROBLEM: **2DIRPATH:**
INSTANCE: A directed graph $G(V, E)$, distinct vertices $x_1, y_1, x_2, y_2 \in V$.
QUESTION: Are there two vertex-disjoint directed paths, one from x_1 to y_1 and the other from x_2 to y_2 in G?

Theorem 2.2 *Given an instance I of* **IDV***-unitcap and only three terminals (root r and two terminals t_1 and t_2), it is NP-hard to decide if it has 2 vertex-disjoint directed Steiner trees.*

Proof: Let $I = (G, x_1, y_1, x_2, y_2)$ be an instance of 2DIRPATH. Construct G' from G by adding three terminal nodes, r, t_1, t_2 with r being the root, and creating directed edges rx_1, rx_2, y_1t_1, x_2t_1, y_2t_2, and x_1t_2. We claim that I is a "Yes" instance if and only if G' has two Steiner trees rooted at r. If there are (vertex) disjoint paths $x_1P_1y_1$ and $x_2P_2y_2$ in G then clearly $x_1P_1y_1 \cup \{rx_1, y_1t_1, x_1t_2\}$ and $x_2P_2y_2 \cup \{rx_2, y_2t_2, x_2t_1\}$ form two vertex-disjoint directed Steiner trees. Conversely, if there are two vertex-disjoint directed Steiner trees T_1 and T_2 in G' then, since r has only two outgoing edges, we may assume that $rx_1 \in T_1$ and $rx_2 \in T_2$. Therefore, there is a path from x_1 to t_1 in T_1, which must go through y_1 (since x_2 is not in T_1), and a path from x_2 to t_2 in T_2, which must go through y_2 (since x_1 is not in T_2). These two paths are vertex-disjoint because T_1 and T_2 are vertex-disjoint. ∎

From Theorems 2.1 and 2.2, it follows that:

Theorem 2.3 *Given an instance I of* **IDE***-unitcap and only three terminals, (root r and two terminals t_1 and t_2) it is NP-hard to decide if it has 2 edge-disjoint directed Steiner trees.*

Now we show that, unless P=NP, any approximation algorithm for **IDV**-unitcap has a guarantee of $\Omega(n^{\frac{1}{3}-\epsilon})$. A similar construction shows a hardness of $\Omega(m^{\frac{1}{3}-\epsilon})$ for **IDE**-unitcap.

Theorem 2.4 *Given an instance of* **IDV***-unitcap, it is NP-hard to approximate the solution within $O(n^{\frac{1}{3}-\epsilon})$ for any $\epsilon > 0$.*

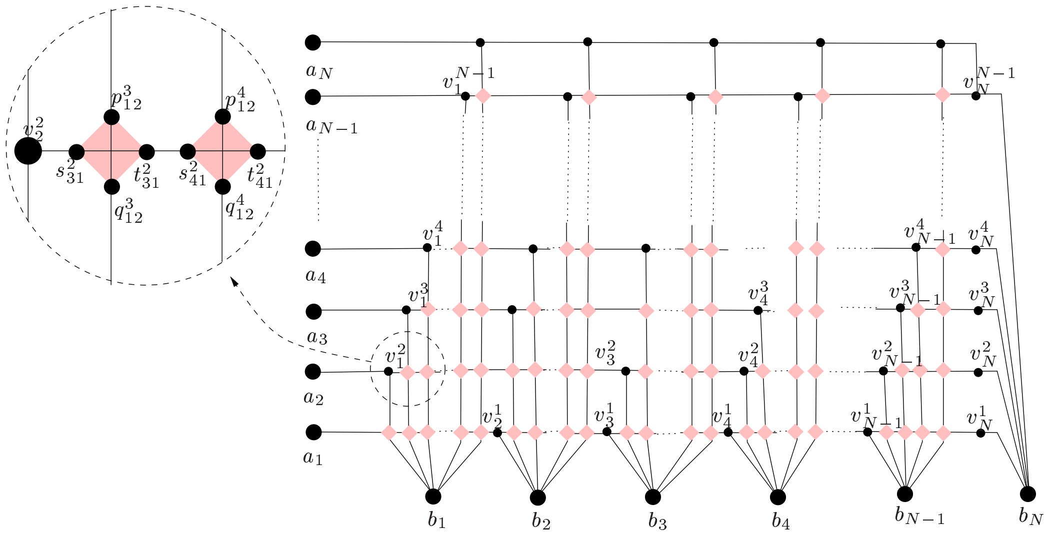

Figure 1: Construction of H: each gray box will be replaced with a copy of G

Proof: We use a reduction from the 2DIRPATH problem. Our proof is inspired by a reduction used in [11] for the edge-disjoint path problem. Assume that $I = (G, x_1, y_1, x_2, y_2)$ is an instance of 2DIRPATH and let $\epsilon > 0$ be given. We construct a directed graph H. First we construct a graph G' whose underlying structure is shown in Figure1. For $N = |V(G)|^{1/\epsilon}$, create two sets of vertices $A = \{a_1, \ldots, a_N\}$ and $B = \{b_1, \ldots, b_N\}$. In the figure, all the edges are directed from top to bottom and from left to right. For each gray box, there is a vertex at each of the four corners, and there are two edges, from left to right and from top to bottom. This graph may be viewed as the union of N vertex-disjoint directed trees $T_1, \ldots, T_N$, where T_i is rooted at a_i and has paths to all the vertices in $B - \{b_i\}$. Each tree T_i consists of one horizontal path H^i, which is essentially the ith horizontal row above b_j's, and starts with $a_i, v_1^i, \ldots$ and ends in v_N^i, together with $N-1$ vertical paths P_j^i $(1 \leq j \neq i \leq N)$, such that each of these vertical paths branches out from the horizontal path, starting at vertex v_j^i and ending at vertex $b_j \in B$. Each vertex v_j^i is in the ith horizontal row, and is the start vertex of a vertical path that ends at b_j; note that there are no vertices v_i^i. Also, note that each gray box corresponds to a triple (i, j, ℓ) where the box is in the ith horizontal line and is in the vertical path that starts at v_j^ℓ and ends at b_j; the corner vertices of the gray box are labeled $s_{\ell j}^i, t_{\ell j}^i, p_{ji}^\ell, q_{ji}^\ell$ for the left, right, top, and bottom corners, respectively. More specifically, for T_1 the horizontal and vertical paths are: $H^1 = a_1, s_{21}^1, t_{21}^1, \ldots, s_{N1}^1, t_{N1}^1, v_2^1, s_{32}^1, t_{32}^1, \ldots, v_3^1, \ldots, v_N^1$, and $P_j^1 = v_j^1, b_j$, for $2 \leq j \leq N$. For T_2 the horizontal and vertical paths are: $H^2 = a_2, v_1^2, s_{31}^2, t_{31}^2, \ldots, s_{N2}^2, t_{N2}^2, v_3^2, \ldots, v_N^2$, and $P_j^2 = v_j^2, p_{j1}^2, q_{j1}^2, b_j$ (for $1 \leq j \neq 2 < N$), and $P_N^2 = v_N^2, b_N$. In general, for T_i:

- $H^i = a_i, v_1^i, s_{(i+1)1}^i, t_{(i+1)1}^i, \ldots, s_{N1}^i, t_{N1}^i, v_2^i, s_{(i+1)2}^i, t_{(i+1)2}^i, \ldots, s_{N2}^i, t_{N2}^i, v_3^i, \ldots, v_{N-1}^i, s_{(i+1)(N-1)}^i, t_{(i+1)(N-1)}^i, \ldots, s_{N(N-1)}^i, t_{N(N-1)}^i, v_N^i$,
- For $j \neq i < N$: $P_j^i = v_j^i, p_{j(i-1)}^i, q_{j(i-1)}^i, p_{j(i-2)}^i, q_{j(i-2)}^i, \ldots, p_{j1}^i, q_{j1}^i, b_j$, and $P_N^i = v_N^i, b_N$.

Graph H is obtained from G' by making the following modifications:

- For each gray box, with corresponding triple say (i, j, ℓ) and with vertices $s^i_{\ell j}, t^i_{\ell j}, p^\ell_{ji}, q^\ell_{ji}$, we first remove the edges $s^i_{\ell j}t^i_{\ell j}$ and $p^\ell_{ji}q^\ell_{ji}$, then we place a copy of graph G and identify vertices x_1, y_1, x_2, and y_2 with $s^i_{\ell j}, t^i_{\ell j}, p^\ell_{ji}$, and q^ℓ_{ji}, respectively.
- Add a new vertex r (root) and create directed edges from r to $a_1, \ldots, a_N$.
- Create N new directed edges $a_i b_i$, for $1 \leq i \leq N$.

The set of terminals (for the directed Steiner trees) is $B \cup \{r\} = \{b_1, \ldots, b_N, r\}$. The proof follows from the following two Lemmas.

Lemma 2.5 *If I is a "Yes" instance of 2DIRPATH, then H has N vertex-disjoint directed Steiner trees.*

Proof: Consider the vertex-disjoint trees $T_1, \ldots, T_N$ explained above. At every gray-box intersection with vertices $s^i_{\alpha\beta}, t^i_{\alpha\beta}, p^\alpha_{\beta i}, q^\alpha_{\beta i}$, instead of using edges $s^i_{\alpha\beta}t^i_{\alpha\beta}$ and $p^\alpha_{\beta i}q^\alpha_{\beta i}$ described in G', we use the disjoint paths that exist in the local copy of G from $s^i_{\alpha\beta}$ (equivalent to x_1 in G) to $t^i_{\alpha\beta}$ (equivalent to y_1 in G) and from $p^\alpha_{\beta i}$ (equivalent to x_2 in G) to $q^\alpha_{\beta i}$ (equivalent to y_2 in G). Now by adding edges ra_i and $a_i b_i$ to T_i, we obtain a Steiner tree for H. Thus H has N vertex-disjoint Steiner trees. This proves Lemma 2.5. ■

Lemma 2.6 *If I is a "No" instance, then H has exactly 1 vertex-disjoint directed Steiner tree.*

Proof: First, note that H always has at least one Steiner tree, namely, the union of paths $L_i = r, a_i, b_i$, for $1 \leq i \leq N$. Now assume that I is a "No" instance and by way of contradiction assume that there is a set $\mathcal{T} = \{T_1, \ldots, T_k\}$, with $k \geq 2$, of vertex-disjoint Steiner trees in H. Note that each a_i belongs to at most one Steiner tree $T_\alpha \in \mathcal{T}$.

Claim 2.7 *There cannot be a directed path from a_i to any b_j (with $j > i$) in any tree $T_\alpha \in \mathcal{T}$.*

Proof: We prove this by induction on i. For the basis of the induction, consider a_1 and suppose that there is a path $P_\alpha(a_1, b_j)$ from a_1 to b_j ($j \geq 2$) in some tree $T_\alpha \in \mathcal{T}$. Let $T_\beta \in \mathcal{T}$ be another tree in $\mathcal{T}$ and look at the path $P_\beta(r, b_1)$ from r to b_1 in T_β. Consider the embedding of these two paths $P_\alpha(a_1, b_j)$ and $P_\beta(r, b_1)$ on the plane. There has to be an intersecting point (on the horizontal path from v^1_1 to v^1_2) of these two paths. In other words, there has to be a gray-box in which these two paths cross each other without having any vertex in common. But since G is a "No" instance, this is not possible. So there is no path from r to b_1 in any other tree $T_\beta \in \mathcal{T}$, a contradiction.

For the induction step, let $i \geq 2$ and assume that there is a path $P_\alpha(a_i, b_j)$ from a_i to b_j ($j > i$) in some tree $T_\alpha \in \mathcal{T}$. Let $T_\beta \in \mathcal{T}$ be any other tree in $\mathcal{T}$ and $P_\beta(r, b_i)$ be a path from r to b_i in T_β. We assume this path goes through a_l, for some $1 \leq l \leq N$. By induction hypothesis, there is no path from $a_1, \ldots, a_{i-1}$ to b_i in any tree. Also, $a_i \in T_\alpha$. So $l > i$.

Again, if we consider the embeddings of these two paths $P_\alpha(a_i, b_j)$ and $P_\beta(r, b_i)$ on the plane, there is an intersecting gray box in which these two paths cross each other without having any vertex in common. But this is impossible because G is a "No" instance. This proves Claim 2.7. ■

Therefore, the only possible path from r to b_N goes through a_N. Thus, there can be only one Steiner tree in $\mathcal{T}$: the one that contains a_N. This proves Lemma 2.6. ■

The number of copies of G in the construction of H is $O(N^3)$ where $N = |V(G)|^{1/\epsilon}$. So the number of vertices in H is $O(N^{3+\epsilon})$. By Lemmas 2.5 and 2.6 it is NP-hard to decide if H has at least N or at most one directed Steiner trees. This creates a gap of $\Omega(n^{\frac{1}{3}-\epsilon})$. This proves Theorem 2.4. ■

For **IDE**-unitcap we use a similar reduction. The only differences are: (i) the instance that we use as the building block in our construction (corresponding to graph G above) is an instance of another well-known NP-hard problem, namely edge-disjoint 2DIRPATH (instead of vertex-disjoint), (ii) the parameter N above is $|E(G)|^{1/\epsilon}$. Using this reduction we can show:

Theorem 2.8 *Given an instance of* **IDE**-*unitcap, it is NP-hard to approximate the solution within* $O(m^{\frac{1}{3}-\epsilon})$ *for any* $\epsilon > 0$.

2.2 Approximation algorithms

In this section we show that, although **GDE** is hard to approximate within a ratio of $O(m^{\frac{1}{3}-\epsilon})$, there is an approximation algorithm with a guarantee of $O(m^{\frac{1}{2}+\epsilon})$ (details in Theorem 2.15). The algorithm is LP-based with a simple rounding scheme similar to those in [2, 19]. The main idea of the algorithm is to start with one of the known approximation algorithms for finding a Minimum-weight Directed Steiner Tree. Using this and an extension of Theorem 4.1 in [16], we obtain an approximate solution to the fractional version of **GDE**. After that, a simple randomized rounding algorithm yields an integral solution. A similar method yields an approximation algorithm for **GDV** that has a guarantee of $O(n^{\frac{1}{2}+\epsilon})$.

We may formulate **GDE** as an integer program (IP). Recall that we have a digraph $G(V,E)$, ℓ roots $r_1,\ldots,r_\ell$, and ℓ sets of terminals $T_1,\ldots,T_\ell$. In the following, $\mathcal{F}$ denotes the collection of all directed Steiner trees in G. We use F to denote an element of $\mathcal{F}$, i.e., F denotes a directed Steiner tree of G. For each $F \in \mathcal{F}$, there is an $i \in \{1,\ldots,\ell\}$ such that F contains T_i and has a directed path from r_i to each node in T_i.

$$\begin{array}{ll} \text{maximize} & \sum_{F\in\mathcal{F}} x_F \\ \text{subject to} & \forall e \in E : \sum_{F:e\in F} x_F \le c_e \\ & \forall F \in \mathcal{F} : \quad x_F \in \{0,1\} \end{array} \tag{1}$$

The *fractional packing edge capacitated directed Steiner tree* problem (fractional **GDE**, for short) is the linear program (LP) obtained by relaxing the integrality condition in the above IP to $x_F \ge 0$. For any instance I of the (integral) packing problem, we denote the fractional instance by I_f. The proof of Theorem 4.1 in [16] may be adapted to prove the following:

Theorem 2.9 *There is an α-approximation algorithm for fractional* **GDE** *if and only if there is an α-approximation algorithm for the minimum (edge weighted) directed Steiner tree problem.*

Charikar et al. [4] gave an $O(n^\epsilon)$-approximation algorithm for the minimum-weight directed Steiner tree problem. This, together with Theorem 2.9 implies:

Corollary 2.10 *There is an $O(n^{\epsilon})$-approximation algorithm for fractional* **GDE***.*

The key lemma in the design of our approximation algorithm for **GDE** is as follows.

Lemma 2.11 *Let I be an instance of* **GDE***, and let φ^* be the (objective) value of a (not necessarily optimal) feasible solution $\{x^*_F : F \in \mathcal{F}\}$ to I_f such that the number of non-zero x^*_F's is polynomially bounded and each $x^*_F < 1$. Then, we can find in polynomial time, a solution to I with value at least $\Omega(\max\{\varphi^*/\sqrt{m},\ \min\{\varphi^{*2}/m,\ \varphi^*\}\})$.*

Proof: We will use the following simple and well-known deviation bound.

Lemma 2.12 *(Chernoff-Hoeffding Bounds) Let $X_1, X_2, \ldots, X_q$ be a set of q independent random variables with $X_i \in \{0,1\}$ and let $X = \sum_{i=1}^{q} X_i$. Then for $0 \le \delta < 1$:*

$$\Pr[X < (1-\delta)\mathrm{E}[X]] \le e^{-\delta^2 \mathrm{E}[X]/2}.$$

The following simple lemma has been used (with $k = 2$) in [2]:

Lemma 2.13 *Assume that $A = \{a_1, \ldots, a_n\}$ is a set of n non-negative reals and let $\mathcal{A}_k$ be the set of all subsets of size k of A. If $\sum_{i=1}^{n} a_i \le Q$, then $\sum_{\{a_{i_1},\ldots,a_{i_k}\}\in\mathcal{A}_k} a_{i_1}a_{i_2}\ldots a_{i_k} \le \binom{n}{k}(Q/n)^k$.*

Proof: For $\alpha, \beta \in \{1, \ldots, n\}$, if $a_\alpha < a_\beta$, then adding any $0 < \epsilon \le a_\beta - a_\alpha$ to a_α and subtracting it from a_β will increase the value of $\sum a_{i_1}a_{i_2}\ldots a_{i_k}$, while keeping the $\sum_{i=1}^{n} a_i$ unchanged. So the maximum value of $\sum a_{i_1}a_{i_2}\ldots a_{i_k}$ is obtained when all a_α's are equal. This proves Lemma 2.13. ■

If $\varphi^* \le 10e\sqrt{m}$ (e is the base of natural logarithm) then it is enough to just find one Steiner tree and return it. So from now on we assume that $\varphi^* \ge 10e\sqrt{m}$. For every tree $F \in \mathcal{F}$ for which $x^*_F > 0$, let's pick that tree with probability x^*_F/λ, for some $\lambda \ge 1$ to be defined later. Note that we assumed $x^*_F < 1$. Let X_F be the random variable that is 1 if we pick tree F and 0 otherwise. Then for $X = \sum_{F\in\mathcal{F}} X_F$ (i.e. the total number of trees picked by the algorithm), we have:

$$\mathrm{E}[X] = \sum_{F\in\mathcal{F}} \Pr[X_F = 1] = \sum_{F\in\mathcal{F}} \frac{x^*_F}{\lambda} = \frac{\varphi^*}{\lambda}.$$

For every edge $e \in E$, define the bad event A_e to be the event that the capacity constraint of e is violated, i.e. more than c_e trees containing e are picked. Our goal is to show that with some positive probability, none of these bad events happen (i.e. all $\overline{A}_e$'s hold) *and* that the total number of trees picked is not too small. We want to find a good upper bound for $\Pr[A_e]$. For every edge e, denote the number of trees F with $x^*_F > 0$ that contain e by ψ_e. By this definition:

$$\Pr[A_e] \le \sum \prod_{i=1}^{c_e+1} x^*_{T_{a_i}}/\lambda,$$

where the summation is over all subsets $\{F_{a_1}, \ldots, F_{a_{c_e+1}}\}$ of size c_e+1 of trees with $x^*_{F_{a_i}} > 0$ that contain edge e. Therefore, using Lemma 2.13:

$$\Pr[A_e] \le \binom{\psi_e}{c_e+1}\left(\frac{c_e}{\lambda\psi_e}\right)^{c_e+1} \le \left(\frac{e\psi_e}{c_e+1}\right)^{c_e+1}\left(\frac{c_e}{\lambda\psi_e}\right)^{c_e+1} \le \frac{e^2}{\lambda^2},$$

where we have used the fact $\binom{n}{k} \le (\frac{en}{k})^k$ for the second inequality. It is intuitively clear that if $\overline{A_e}$ holds then it does not increase the probability of any other $A_{e'}$. In other words, events $\overline{A_e}$ are "positively correlated". This will be formalized in the following lemma that follows easily from FKG inequality:

Lemma 2.14 $\Pr[\bigwedge_{e\in E}\overline{A}_e] \ge \prod_{e\in E}\Pr[\overline{A}_e] \ge (1-\frac{e^2}{\lambda^2})^m$.

So, the probability that at least one event A_e happens is at most $1-(1-e^2/\lambda^2)^m$. Also, by Lemma 2.12, for $0 \le \delta < 1$: $\Pr[X < (1-\delta)\mathrm{E}[X]] \le e^{-\delta^2\varphi^*/2\lambda}$. Thus:

$$\Pr[(X < (1-\delta)\mathrm{E}[X]) \vee (\exists e \in E : A_e)] \le e^{-\delta^2\varphi^*/2\lambda} + 1 - (1-e^2/\lambda^2)^m.$$

Using the approach of [2] (which is essentially the method of conditional probability), if we can show that for suitable δ and λ: $(1-e^2/\lambda^2)^m > e^{-\delta^2\varphi^*/2\lambda}$ then we can efficiently find a selection of trees such that $X \ge (1-\delta)\varphi^*/\lambda$ and that no edge constraint is violated.

Case 1: If $\varphi^* \le m$ and we set $\delta = \frac{1}{2}$ and $\lambda = e\sqrt{m}$, then (recall that $\varphi^* \ge 10e\sqrt{m}$) we can find a collection $\mathcal{F}' \subseteq \mathcal{F}$ of directed Steiner trees that obey the edge capacities with $|\mathcal{F}'| \ge \varphi^*/2e\sqrt{m}$.

Case 2: If $\varphi^* \le m$ then by setting $\delta = \frac{1}{2}$ and $\lambda = 32em/\varphi^*$, we can find a collection $\mathcal{F}' \subseteq \mathcal{F}$ of directed Steiner trees that obey the edge constraints with $|\mathcal{F}'| \ge \varphi^{*2}/64em$.

Case 3: if $\varphi^* > m$ then there is a constant $c_0 > 0$ such that with $\delta = \frac{1}{2}$ and $\lambda = c_0$: $(1-e^2/\lambda^2)^m > e^{-\delta^2\varphi^*/2\lambda}$. Again, we can find a collection $\mathcal{F}' \subseteq \mathcal{F}$ of directed Steiner trees with $|\mathcal{F}'| \ge \frac{\varphi^*}{2c_0}$. This proves Lemma 2.11. ■

Theorem 2.15 *Let $\epsilon > 0$ be a constant. There is a polynomial-time algorithm for* **GDE** *that finds a set of directed Steiner trees (satisfying the edgecapacity constraints) of size $\Omega(\max\{\varphi_f/m^{\frac{1+\epsilon}{2}},\ \varphi_f^2/m^{1+\epsilon}\})$ if $\varphi_f \le m$, and of size $\Omega(\varphi_f/m^{\frac{\epsilon}{2}})$ otherwise, where φ_f denotes the optimal value of the instance of fractional* **GDE**.

Proof: Let I_f be the fractional instance. By Corollary 2.10, we can find an approximate solution x with objective value φ^* for I_f such that $\varphi^* \ge c\varphi_f/m^{\frac{\epsilon}{2}}$ for some constant c and the given $\epsilon > 0$.

Then we apply a preprocessing step to the fractional solution x. For every Steiner tree F with $x_F \ge 1$ we "take out" $\lfloor x_F \rfloor$ copies of that tree and put it in the final integral solution, we decrease x_F by $\lfloor x_F \rfloor$, and also we update the capacities of the edges accordingly. This decomposes x into a (multi)set of Steiner trees $\mathcal{F}_1$ and a fractional part (with each entry $x_F < 1$). We will "round" the fractional part x to an integer solution (using Lemma 2.11). For the rest of the proof we may assume that the fractional solution x has each entry < 1, since the other case reduces to this one.

Note that the approximate fractional solution x contains only a polynomial number of Steiner trees with non-zero fractional values (this follows from the proof of Theorem 2.9 which is essentially the same as Theorem 4.1 in [16]). If we substitute φ^* in Lemma 2.11 we obtain an approximation algorithm that finds a set $\mathcal{F}'$ of directed Steiner trees such that $\mathcal{F}'$ has the required size. ■

For **GDV** we do the following. Given an instance I of **GDV** with graph $G(V, E)$ and terminal sets $T_1, \ldots, T_\ell \subseteq V$ (with $|V| = n$ and $|E| = m$), we first apply Theorem 2.1 to produce an instance I' of **GDE** with graph $G'(V', E')$ and terminal sets $T'_1, \ldots, T'_\ell \subseteq V'$. By the construction of G' we have $|V'| = 2|V| = 2n$ and there are at most n edges in E' with bounded capacities (corresponding to the vertices of G). Therefore, if we use the algorithm of Theorem 2.15, the number of bad events will be n, rather than m. Using this observation we have the following:

Theorem 2.16 *Let $\epsilon > 0$ be a constant. There is a polynomial-time algorithm for* **GDV** *that finds a set of directed Steiner trees (satisfying the vertex capacity constraints) of size $\Omega(\max\{\varphi_f/n^{\frac{1+\epsilon}{2}}, \varphi_f^2/n^{1+\epsilon}\})$ if $\varphi_f \leq n$, and of size $\Omega(\varphi_f/n^{\frac{\epsilon}{2}})$ otherwise, where φ_f denotes the optimal value of the instance of fractional* **GDV**.

3 Packing Undirected Steiner Trees

For packing edge-disjoint undirected Steiner trees (**IUE**-unitcap), Jain et al. [16] showed that the (general) problem is APX-hard, and Kaski [18] showed the special case of the problem with only 7 terminal nodes is NP-hard. Here we show that **IUE**-unitcap is APX-hard even when there are only 4 terminals. In an early draft of this paper we also showed, independently of [18], that finding two edge-disjoint Steiner trees is NP-hard. Both of these hardness results carry over (using similar constructions) to **IUV**-unitcap. The following observation will be used in our proofs:

Observation 3.1 *For any solution of any of our Steiner tree packing problems, we may assume that: (1) In any Steiner tree, none of the leaves is a Steiner node (otherwise we simply remove it). (2) Every Steiner node with degree 3 belongs to at most one Steiner tree.*

3.1 Hardness results for small-parameter problems

Using the above observation, the proof of NP-hardness for finding two edge-disjoint Steiner trees (for instance see [18]) implies the following theorem.

Theorem 3.2 *Finding 2 undirected vertex disjoint Steiner trees is NP-hard.*

Theorem 3.3 **IUE***-unitcap is APX-hard even if there are only 4 terminals.*

Proof: We use a reduction from Bounded 3-Dimensional Matching (B3DM). Assume that we are given three disjoint sets X, Y, Z (each corresponding to one part of a 3-partite graph G), with $|X| = |Y| = |Z| = n$, and a set $E \subseteq X \times Y \times Z$ containing m triples. Furthermore, we assume that each vertex in $X \cup Y \cup Z$

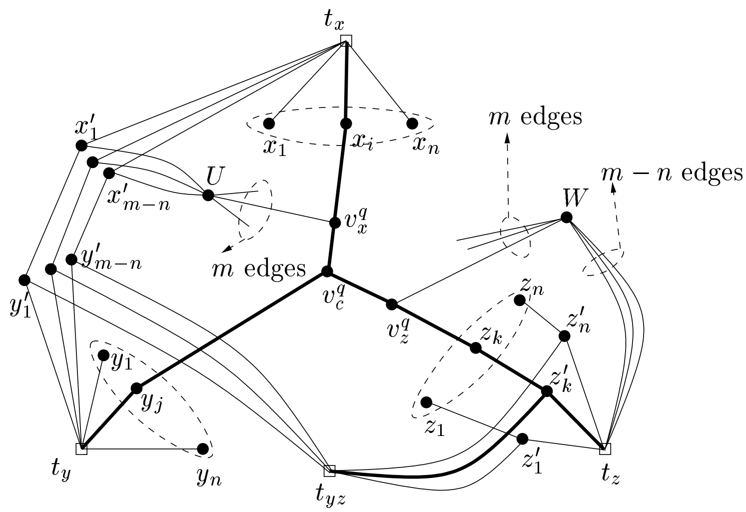

Figure 2: Construction with 4 terminals from B3DM

belongs to at most 5 triples. It is known [17] that there is an absolute constant $\epsilon_0 > 0$ such that it is NP-hard to distinguish between instances of B3DM where there is a perfect matching (i.e., n vertex-disjoint triples) and those in which every matching (set of vertex-disjoint triples) has size at most $(1-\epsilon_0)n$. Assume that $x_1, \ldots, x_n$, $y_1, \ldots, y_n$, and $z_1, \ldots, z_n$ are the nodes of X, Y, and Z, respectively. We construct a graph H which consists of:

- 4 terminals t_x, t_y, t_z, and t_{yz}.
- Non-terminals $x_1, \ldots, x_n, y_1, \ldots, y_n$, and $z_1, \ldots, z_n$ (corresponding to the nodes in X, Y, Z), $x'_1, \ldots, x'_{m-n}$, $y'_1, \ldots, y'_{m-n}$, and $z'_1, \ldots, z'_n$.
- Two non-terminals U and W.
- Edges $t_x x_i$, $t_y y_i$, $t_z z'_i$, $z'_i z_i$, and $t_{yz} z'_i$, for $1 \le i \le n$.
- Edges $t_x x'_i$, $x'_i y'_i$, $x'_i U$, $y'_i t_y$, and $y'_i t_{yz}$, for $1 \le i \le m-n$.
- $m-n$ parallel edge from W to t_z.
- For each triple $e_q = x_i y_j z_k \in E$, 3 non-terminals v_c^q, v_x^q, v_z^q and the following edges: $v_x^q x_i$, $v_c^q y_j$, $v_z^q z_k$, $v_c^q v_x^q$, $v_c^q v_z^q$, $v_x^q U$, and $v_z^q W$.

See Figure 2. Now we prove that (a)[completeness] if G has a perfect matching then H has m edge-disjoint Steiner trees and (b)[soundness] if every matching in G has size at most $(1-\epsilon_0)n$ then H has at most $(1-\epsilon_1)m$ edge-disjoint Steiner trees, for $\epsilon_1 \ge \epsilon_0/110$.

Lemma 3.4 (completeness) *If G has a perfect matching $M = \{e_{a_1}, e_{a_2}, \ldots, e_{a_n}\}$ then H has m edge-disjoint Steiner trees.*

Proof: For each triple $e_q = x_i y_j z_k \in M$ we construct a tree T_q by using the following edges: $t_x x_i$, $t_y y_j$, $t_z z'_k$, $z'_k z_k$, $x_i v^q_x$, $v^q_x v^q_c$, $v^q_c y_j$, $v^q_c v^q_z$, $v^q_z z_k$, and $z'_k t_{zy}$. (See the tree shown by bold lines in Figure 2). This gives a set S_1 of n edge-disjoint trees. Without loss of generality assume that $e_1, \ldots, e_{m-n}$ are the triples that are *not* in M. For each triple $e_p = x_{i'} y_{j'} z_{k'} \notin M$, $1 \leq p \leq m-n$, we construct a tree T_p by using the following edges: $t_x x'_p$, $x'_p y'_p$, $y'_p t_y$, $y'_p t_{yz}$, $x'_p U$, $U v^p_x$, $v^p_x v^p_c$, $v^p_c v^p_z$, $v^p_z W$, and (one of the parallel edges) $W t_z$. This gives a set S_2 of $m-n$ edge-disjoint trees. It is not hard to see that all these trees in S_1 and S_2 are edge-disjoint. This proves Lemma 3.4. ■

Now assume that H has a set $\mathcal{T} = \{T_1, \ldots, T_{m'}\}$ of edge-disjoint Steiner trees, with $m' = (1-\epsilon_1)m$. Our goal is to show that, G will have a matching of size at least $(1-110\epsilon_1)n$.

Claim 3.5 *There is a subset $\mathcal{T}' \subseteq \mathcal{T}$ of size at least $(1-11\epsilon_1)m$ such that every tree $T_i \in \mathcal{T}'$ has the following properties: (i) all the terminals have degree 1, and (ii) there is exactly one (unique) vertex v^q_c (for some $1 \leq q \leq m$) in T_i and furthermore both $v^q_x v^q_c$ and $v^q_c v^q_z$ are in T_i, and there is no $q' \neq q$ for which $v^{q'}_x$ or $v^{q'}_z$ is in T_i.*

Proof: Since degree of t_x is exactly m in H, there are at most $\epsilon_1 m$ trees in $\mathcal{T}$ in which t_x is not a leaf. To see this, let α be the number of Steiner trees in $\mathcal{T}$ that each have at least 2 edges incident with t_x; since these trees "use" at least 2α edges incident with t_x, there are at most $m - 2\alpha$ other Steiner trees in $\mathcal{T}$; then we have $m - \epsilon_1 m \leq |\mathcal{T}| \leq m - \alpha$, and this implies that $\alpha \leq \epsilon_1 m$. The same claim applies to all the terminals, because they all have degree exactly m.

It follows that there is a set $\mathcal{T}'' \subseteq \mathcal{T}$ of size at least $(1-3\epsilon_1)m$ of trees in which t_{yz} and t_y both have degree 1. For each tree of $\mathcal{T}''$, there is at least one $1 \leq q \leq m$, such that the path that connects t_z to t_x goes through edge $v^q_z v^q_c$. To see this note that t_{yz} has degree 1 in every tree of $\mathcal{T}''$, hence, for each of these trees, the t_z, t_x path does not use any edge incident to t_{yz}; moreover, if we delete t_{yz} and all the edges $v^q_z v^q_c$ for $q = 1, \ldots, m$, then t_x and t_z are disconnected; thus the t_z, t_x path must use one of the edges $v^q_z v^q_c$. Then the number of trees in $\mathcal{T}''$ that have at least two vertices v^q_c and $v^{q'}_c$ is at most $3\epsilon_1 m$; also, the same claim holds for vertices v^q_z and $v^{q'}_z$. (To see this, let α be the number of trees in $\mathcal{T}''$ that each have at least 2 vertices v^q_c and $v^{q'}_c$; note that the vertices v^q_c and v^q_z have degree 3 so by Observation 3.1(2) each such vertex is in at most one tree; then there are at most $m - 2\alpha$ other trees in $\mathcal{T}''$, since each of the trees in $\mathcal{T}''$ has a vertex v^q_z; thus we have $(1-3\epsilon_1)m \leq |\mathcal{T}''| \leq m - \alpha$, and this implies that $\alpha \leq 3\epsilon_1$.) Therefore, there is a set $\mathcal{T}^* \subseteq \mathcal{T}''$ of size at least $(1-6\epsilon_1)m$ of trees for which there is a unique q such that both v^q_c and v^q_z are in the tree and there is no $q' \neq q$ such that either $v^{q'}_c$ or $v^{q'}_z$ is in the tree. Similarly, since t_y has degree 1, in every tree of $\mathcal{T}''$, the path between t_z and t_x cannot contain t_y. Hence, each tree of $\mathcal{T}''$ contains at least one vertex v^p_x, for some $1 \leq p \leq m$. In particular each tree of $\mathcal{T}^*$ that contains v^q_c and v^q_z (for some $1 \leq q \leq m$), contains v^q_x as well. Then the number of trees in $\mathcal{T}''$ that have at least 2 vertices v^p_x is at most $3\epsilon_1 m$. Thus, at least $(1-6\epsilon_1-3\epsilon_1)m$ of the trees in $\mathcal{T}^*$ do not violate condition(ii). There are at most $2\epsilon_1 m$ trees in $\mathcal{T}''$ such that either t_x or t_z is a non-leaf (by the argument at the start of this proof). Hence, the number of trees that violate neither (i) nor (ii) is at least $(1-9\epsilon_1-2\epsilon_1)m = (1-11\epsilon_1)m$. This proves Claim 3.5. ■

Consider a set $\mathcal{T}'$ of Steiner trees of H as described in the previous claim. Note that $|\mathcal{T}'| \geq (1-11\epsilon_1)m$. Pick any tree $T_a \in \mathcal{T}'$ that contains $t_x x'_i$ for some $1 \leq i \leq m-n$. Clearly the path that connects t_x to t_z goes through the unique vertex v^q_c that belongs to T_a (because t_{yz} has degree 1). We claim that y'_i cannot belong to any tree in $\mathcal{T}'$ other than T_a. Otherwise, let $y'_i \in T_b$, for some $b \neq a$. Therefore, because $x'_i \notin T_b$ and by Observation 3.1(1), $t_y y'_i$ and $t_{yz} y'_i$ must be in T_b. But since both t_y and t_{yz} are leaves in every tree in $\mathcal{T}'$ and in particular in T_b, $y'_i x'_i$ must be in T_b, a contradiction. Then we may add y'_i to T_a (if it is not already in T_a), and add the edges $y'_i t_y$ and $y'_i t_{yz}$ (if other edges are incident to t_y or t_{yz}, then we remove those edges). We still have a Steiner tree which is edge-disjoint from the other trees in $\mathcal{T}'$. We apply this modifications for any tree $T_a \in \mathcal{T}'$ that contains some edge $t_x x'_i$ for some $1 \leq i \leq m-n$.

Claim 3.6 *There is a set $\mathcal{T}'' \subseteq \mathcal{T}'$ of size at least $(1-22\epsilon_1)m$ such that every tree in $\mathcal{T}''$ contains at most one vertex from $Q = \{y'_1, \ldots, y'_{m-n}\} \cup \{z'_1, \ldots, z'_n\}$.*

Proof: Since all vertices in Q have degree 3, each of them belongs to at most one Steiner tree. So, once a vertex $v \in Q$ is in a tree $T_a \in \mathcal{T}'$ then the edges $t_{yz}v$ cannot be in any other tree in $\mathcal{T}'$. Therefore, if there are α trees in $\mathcal{T}'$ that each contain two or more vertices from Q, then they "use" at least 2α edges incident with t_{yz}, and there can be at most $m-2\alpha$ other Steiner trees in $\mathcal{T}'$. Then we have $m-11\epsilon_1 m \leq |\mathcal{T}'| \leq m-\alpha$, and this implies that $\alpha \leq 11\epsilon_1 m$. We remove from $\mathcal{T}'$ all the trees that have ≥ 2 vertices from Q. This gives the desired set $\mathcal{T}''$, and this proves Claim 3.6. ■

Consider the subset $\mathcal{T}'' \subseteq \mathcal{T}'$ as defined in the previous claim. Recall that for every tree $T_a \in \mathcal{T}''$, (i) terminals have degree 1, (ii) there is one unique vertex v_c^q in T_a and both edges $v_c^q v_x^q$ and $v_c^q v_z^q$ are in T_a, and (iii) there is no other vertex $v_x^{q'}$ or $v_z^{q'}$ in T_a, for $q' \neq q$, (iv) there is at most one vertex from set Q in T_a, and (v) if $t_x x'_i \in T_a$ for some $1 \leq i \leq m-n$ then $t_y y'_i$, $t_{yz} y'_i$, and $y'_i x'_i$ are all in T_a, and therefore no vertex from $\{z'_1, \ldots, z'_n\}$ is in T_a, i.e., the edge incident with t_z in T_a is $t_z W$. Remove all the trees in $\mathcal{T}''$ that satisfy condition (v) above to obtain set $\mathcal{T}_{new}$. Since $m \leq 5n$, we have $|\mathcal{T}_{new}| \geq (1-22\epsilon_1)m-(m-n) \geq (1-110\epsilon_1)n$.

Lemma 3.7 (soundness) *$\mathcal{T}_{new}$ induces a matching of size $|\mathcal{T}_{new}|$ in G.*

Proof: By definition of $\mathcal{T}_{new}$, in every tree $T_a \in \mathcal{T}_{new}$: (i) t_x and t_y and t_z are adjacent to vertices x_i, y_j, and z'_k, respectively (for some unique $1 \leq i,j,k \leq n$), and $z'_k z_k \in T_a$, and (ii) there is exactly one (unique) v_c^q that belongs to T_a, and $v_x^q v_c^q \in T_a$ and $v_c^q v_z^q \in T_a$, and (iii) there is no other vertex $v_x^{q'}$ or $v_z^{q'}$ in T_a, that is $x_i v_x^q \in T_a$ and $v_z^q z_k \in T_a$ and $v_c^q y_j \in T_a$. This implies that T_a induces a triple (x_i, y_j, z_k) in the 3-partite graph G. Since the trees in T_{new} are edge-disjoint and moreover, each of these trees contains exactly one node from each of the 3 sets $\{x_1, \ldots, x_n\}$, $\{y_1, \ldots, y_n\}$, $\{z_1, \ldots, z_n\}$, it follows that these $|T_{new}|$ triples are vertex-disjoint. Thus they form a matching of size at least $(1-110\epsilon_1)n$ in G. This proves Lemma 3.7. ■

By Lemma 3.7, if every matching in G has size at most $(1-\epsilon_0)n$ then H has at most $(1-\epsilon_0/110)m$ edge-disjoint Steiner trees. This completes the proof of Theorem 3.3. ■

The constant 110 in the above theorem is not optimal. We can find explicit lower bounds for the hardness of **IUE**-unitcap with constant number of terminals using the known hardness results for kDM, for higher values of k. For instance, Hazan et al. [15] proved that 4DM (with upper bounds on the degree of each vertex) is hard to approximate within a factor $\frac{53}{54}$. Using this and a reduction similar to the one presented in Theorem 3.3 it seems possible to show that **IUE**-unitcap with 5 terminals is hard to approximate within a factor $(1+\frac{1}{2000})$. The proof of Theorem 3.3 extends to give the next result.

Theorem 3.8 **IUV**-*unitcap is APX-hard with only 4 terminals.*

Proof: First, we prove APX-hardness with only 6 terminals, by using the same reduction as in Theorem 3.3. The only difference is that vertices U and W are also terminals in the construction. Now it is not hard to prove that: (i) if the given 3-partite graph G has a perfect matching then H has m vertex disjoint Steiner trees, and (ii) if every matching in G has size at most $(1-\epsilon_0)n$ then H has at most $(1-\epsilon_1)m$ vertex-disjoint Steiner trees, where ϵ_1 is within a constant factor of ϵ_0. We skip the details.

A more careful reduction similar to the one in Theorem 3.3 improves the number of terminals from 6 to 4. The basic change is to replace U with $m-n$ copies of it $u_1,\ldots,u_{m-n}$. We have an edge from x'_i to each u_i (for $i=1,\ldots,m-n$). We connect every $u_1,\ldots,u_{m-n}$ to every vertex v^q_x (for $q=1,\ldots,m$). We do similar changes for W, that is, we replace it with $m-n$ copies $w_1,\ldots,w_{m-n}$, and connect each of $w_1,\ldots,w_{m-n}$ to t_z and to each vertex v^q_z (for $q=1,\ldots,m$). We skip the details as they are similar to those of Theorem 3.3. ∎

3.2 The unrestricted IUV and GUE problems

The next theorem shows that **IUV**-unitcap is significantly harder than **IUE**-unitcap. We show this by a reduction from the set-cover packing problem (or domatic number problem). Given a bipartite graph $G(V_1\cup V_2, E)$, a *set-cover* (of V_2) is a subset $S\subseteq V_1$ such that every vertex of V_2 has a neighbor in S. A *set-cover packing* is a collection of pairwise disjoint set-covers of V_2. The goal is to find a packing of set-covers of maximum size. Feige et al. [8] show that, unless P=NP, there is no $o(\log n)$-approximation algorithm for set-cover packing, where $n=|V_1|+|V_2|$. We have the following theorem.

Theorem 3.9 **IUV**-*unitcap, even restricted to the case that both the set of terminals and the set of Steiner nodes are independent, cannot be approximated within ratio $c_0\log n$, for some constant $c_0>0$, unless P=NP.*

Proof: Given a bipartite graph $G(V_1\cup V_2,E)$ as the instance of set-cover packing problem, the instance for **IUV**-unitcap problem will be G' that is obtained from G by adding a vertex t_0 and connecting it to all the vertices in V_1. Let the terminal set of G' be $t_0\cup V_2$. We claim that G' has set-cover packing of size p if and only if G' has p vertex-disjoint Steiner trees. If sets $S_1,\ldots,S_p$ form a set-cover packing then it is easy to see that $T_i=S_i\cup V_2\cup\{t_0\}$, for $1\le i\le p$, forms a set of vertex-disjoint Steiner trees. Conversely, if $T_1,\ldots,T_p$ are vertex-disjoint Steiner trees then, since V_2 is an independent set, for each T_i there has to be a set $S_i\subseteq V_1$ of vertices such that every vertex in V_2 has a neighbor in S_i in order to be connected to the rest of the tree. ∎

On the other hand, we obtain an $O(\sqrt{n}\log n)$ algorithm for **GUE** (which contains **IUV**-unitcap as a special case). To do so, consider the fractional version of **GUE** obtained by relaxing the integrality condition in the IP formulation. The separation problem for the dual of this LP is the minimum node-weighted Steiner tree problem. For this problem, Guha and Khuller [14] give an $O(\log n)$ approximation algorithm. Using the following analog of Theorem 2.9 (or Theorem 4.1 in [16]) we obtain a polytime $O(\log n)$-approximation algorithm for fractional **GUE**.

Lemma 3.10 *There is an α-approximation for fractional* **GUE** *if and only if there is an α-approximation for the minimum node-weighted Steiner tree problem.*

Remark: Lemma 3.10 and the fact that the minimum node-weighted Steiner tree problem is hard to approximate within $O(\log k)$ (with k being the number of terminals) yields an alternative proof for the $\Omega(\log k)$ hardness of **IUV**-unitcap.

The algorithm for **GUE** is similar to the ones we presented for **GDE** and **GDV**. That is, we apply randomized rounding to the solution of the fractional **GUE** instance. Skipping the details, this yields the following:

Theorem 3.11 *Let $\epsilon > 0$ be a constant. There is a polynomial-time algorithm for* **GUE** *that finds a set of Steiner trees (satisfying the vertex capacity constraints) of size $\Omega(\max\{\varphi_f/\sqrt{n}\log n,\ \varphi_f^2/n\log^2 n\})$ if $\varphi_f \leq n$, and of size $\Omega(\varphi_f/\log n)$ otherwise, where φ_f denotes the optimal value of the instance of fractional* **GUE**.

3.3 Packing vertex-disjoint priority Steiner trees

The priority Steiner tree problem has been studied by Charikar et al. [5]. Here, we study the problem of packing vertex-disjoint priority Steiner trees of undirected graphs. (One difference with the earlier work in [5] is that we associate weights and priorities with vertices rather than with edges.) Consider an undirected graph $G = (V, E)$ with a set of terminals $T \subseteq V$, one of which is distinguished as the root r. Let every vertex v have a nonnegative integer p_v as its priority, and let every vertex v have a nonnegative integer c_v as its capacity. A *priority Steiner* tree is a Steiner tree such that for each terminal $t \in T$ every vertex v on the r, t path has priority $p_v \geq p_t$. In the problem **IUV**-priority (packing undirected vertex-capacitated priority Steiner trees) the goal is to find a maximum set of priority Steiner trees obeying vertex capacities (i.e., for each Steiner vertex $v \in V - T$ the number of trees containing v is $\leq c_v$). In the problem **GUV**-priority, we have ℓ sets of terminals $T_1, \ldots, T_\ell$ and ℓ roots $r_1, \ldots, r_\ell$ (where $r_i \in T_i$, for $i = 1, \ldots, \ell$, and ℓ is polynomial in n), and the goal is to find a maximum set of trees obeying the vertex capacities, where each of these trees must have an $i \in \{1, \ldots, \ell\}$ such that the tree is a priority Steiner tree with root r_i and terminal set T_i (that is, the tree contains all the nodes in T_i and for each $t \in T_i$ every vertex v on the r_i, t path has $p_v \geq p_t$). The algorithm we presented for **GUE** extends to **GUV**-priority, giving roughly the same approximation guarantee.

Theorem 3.12 *Let $\epsilon > 0$ be a constant. There is a polynomial-time algorithm for* **GUV***-priority that finds a set of priority Steiner trees (satisfying the vertex capacity constraints) of size $\Omega(\max\{\varphi_f/n^{\frac{1+\epsilon}{2}},\ \varphi_f^2/n^{1+\epsilon}\})$ if $\varphi_f \leq n$, and of size $\Omega(\varphi_f/n^{\frac{\epsilon}{2}})$ otherwise, where φ_f denotes the optimal value of the instance of fractional* **GUV***-priority.*

Proof: The fractional packing problem for **GUV**-priority is obtained in the usual way (formulate the packing problem as an integer program and then relax the integer variables to be nonnegative reals). First note that in the dual of the LP formulation of fractional **GUV**-priority, the separation problem is the minimum (node-weighted) priority Steiner tree problem, where the node weights correspond to the (nonnegative) variables of the dual. The following lemma, together with the $O(n^\epsilon)$ approximation algorithm of [4] for minimum (arc-weighted) directed Steiner trees, shows that there is an $O(n^\epsilon)$ approximation algorithm for the minimum (node-weighted) priority Steiner tree problem. (In the lemma, we abuse the notation and denote instances of these problems by the associated graphs or digraphs.)

Lemma 3.13 *There is a polynomial-time algorithm that given an instance $G = (V, E)$ of the minimum (node-weighted undirected) priority Steiner tree problem constructs an instance G'' of the minimum (arc-weighted) directed Steiner tree problem such that G has a priority Steiner tree of weight W if and only if G'' has a directed Steiner tree of weight W.*

Proof: Let $G(V, E)$ be the given undirected graph, let r be the root, and let the set of terminals be $T \subseteq V$; each vertex $v \in V$ has a weight w_v and a priority p_v. Without loss of generality, we may assume that the vertex priorities p_v are in the range $1, \ldots, n$, and the root r has priority n. We assume (w.l.o.g.) that each terminal in G has weight zero.

Let G' be the digraph obtained from graph G by first splitting each vertex v into two vertices v^1, v^2 and adding the arc v^1v^2, and then replacing each edge $xy \in E$ by a pair of arcs x^2y^1, y^2x^1; moreover, for each $v \in V$, define the weight and priority of arc v^1v^2 to be w_v and p_v, and define the weight and priority of the other arcs to be 0 and n, respectively. For each possible priority $\ell \in \{1, \ldots, n\}$, we start with a copy G'_ℓ of G' and we remove all arcs that have priority less than ℓ (thus each arc in G'_ℓ has priority $\geq \ell$). Finally, for $\ell = n, n-1, \ldots, 2$, for each node $v \in G$ we add an arc from the copy of v^2 in G'_ℓ to the copy of v^2 in $G'_{\ell-1}$. These new arcs have a weight of 0 and a priority of n (so their weights and priorities will not affect our proof). In the resulting digraph, let the root be the node r^1 in G'_n, and for each terminal vertex $t \in T$ of G let the copy of node t^2 in G'_{p_t} be a terminal (thus t corresponds to t^2 in the copy of G' indexed by p_t). Denote the directed instance by G'', and denote its set of terminals and root by T'' and r'', respectively (see Figure 3 for an illustration).

Consider any terminal $t \in T$ of G, and let t'' denote the corresponding terminal of G''. Clearly, in G'', every arc in any (directed) path from r'' to t'' has priority at least p_t. Moreover, for any path between r and t in G such that every vertex v in the path has $p_v \geq p_t$, G'' has directed paths from r'' to t''. Our construction picks one of these r'', t'' directed paths as follows: Let the r, t path of G be $v_0, v_1, v_2, \ldots, v_q$ (where $v_0 = r, v_q = t$); we assign a number $p''(v)$ to each vertex v in this path such that $p_t \leq p''(v) \leq p_v$, $p''(t) = p_t$, $p''(r) = p_r$, and moreover, these numbers form a non-increasing sequence along the r, t path (i.e., $p''(v_0) \geq p''(v_1) \geq \ldots \geq p''(v_q)$). Then the r'', t'' directed path of G'' consists of the nodes $r'' = r^1$, r^2 in G'_n, followed by a directed path from (the copy of) r^2 in G'_n to (the copy of) r^2 in $G'_{p''(v_1)}$, followed by v_1^1 and v_1^2 in $G'_{p''(v_1)}$, followed by a directed path from (the copy of) v_1^2 in $G'_{p''(v_1)}$ to (the copy of) v_1^2 in $G'_{p''(v_2)}$, followed by v_2^1 and v_2^2 in $G'_{p''(v_2)}$, ..., followed by v_q^1 and v_q^2 in $G'_{p''(v_q)}$.

It can be seen that corresponding to any directed Steiner tree H'' of G'' there is a priority Steiner tree of G whose weight is at most the weight of H''. (To see this, let S be the set of Steiner vertices v of G such that one of the copies of the arc v^1v^2 is in H''; then the subgraph of G induced by $S \cup T$ has an r, t path for each $t \in T$, and moreover, each Steiner vertex v in an r, t path has $p_v \geq p_t$ because the arc v^1v^2 (in H'') occurs in say $G'_{p''(v)}$ where $p_v \geq p''(v) \geq p_t$.)

Moreover, it can be seen that corresponding to any priority Steiner tree H of G there is a directed Steiner tree H'' of G'' with the same weight. (To see this, we associate a number $p''(v)$ with each vertex v of H, where $p''(v)$ is the maximum priority of any terminal in the subtree of H rooted at v; thus $p''(v) \leq p_v$, $\forall v \in V$, and for each terminal t, every vertex v in the r, t path of H has $p''(v) \geq p_t$; then for each terminal vertex $v \in T$ and for each Steiner vertex v in H, we add to H'' (the copy of) nodes v^1, v^2 and arc v^1v^2 in $G'_{p''(v)}$; finally we add to H'' appropriate arcs of weight 0 and priority n.)

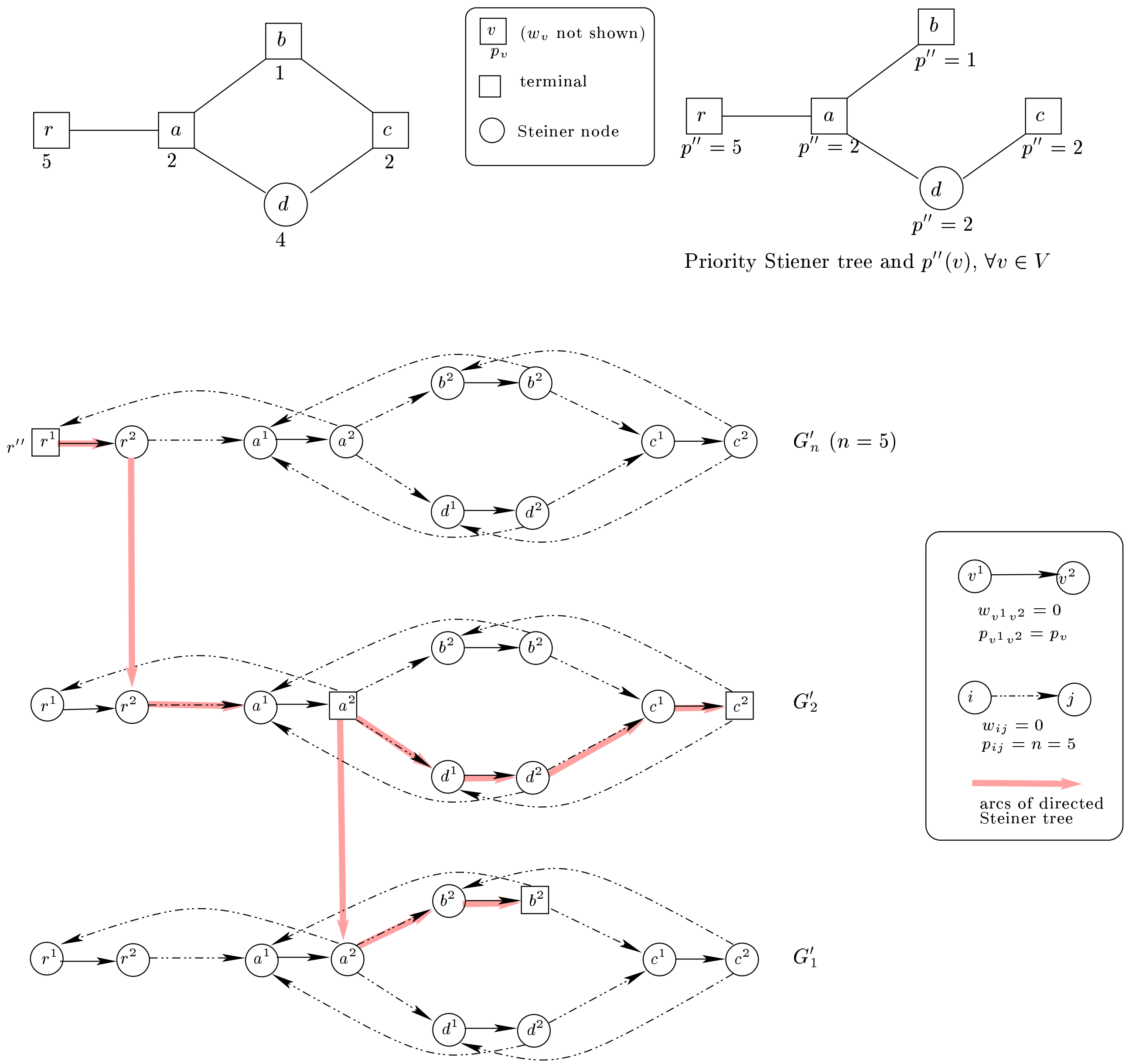

Figure 3: Graph $G(V, E)$, and directed graph G'', showing only G'_5, G'_2, and G'_1; also, only some of the arcs from v^2 in G'_l to v^2 in G'_{l-1} are shown.

Consequently, an α-approximate solution to the minimum arc-weighted directed Steiner tree problem in G'' gives an α-approximate solution to the minimum node-weighted priority Steiner tree problem in G. This proves Lemma 3.13 ■

The next result follows from Lemma 3.13 and another analog of Theorem 2.9 (or Theorem 4.1 in [16]) that relates the fractional **GUV**-priority problem to the minimum (node weighted) priority Steiner tree problem.

Lemma 3.14 *There is an $O(n^{\epsilon})$-approximation algorithm for the fractional* **GUV***-priority problem.*

The above lemma can be used to obtain an $O(n^{\frac{1}{2}+\epsilon})$-approximation algorithm for the **GUV**-priority problem, similar to the approximation algorithm for **GUE**; we find an approximately optimal solution to the fractional packing problem and then apply randomized rounding; see the analysis in Section 2.2. This proves Theorem 3.12. ■

On the other hand, we prove an $\Omega(n^{\frac{1}{3}-\epsilon})$ hardness result for **IUV**-priority by adapting the proof of Theorem 2.4, thus improving on our logarithmic hardness result for **IUV**. The main difference from the proof of Theorem 2.4 is that we use instances of the Undir-Node-USF problem (*Undirected Node capacitated Unsplittable Flow*) – which is shown to be NP-complete in [11] – instead of instances of 2DIRPATH as the modules that are placed on the "gray boxes" in Figure 1.

Theorem 3.15 *Given an instance of* **IUV***-priority, it is NP-hard to approximate the solution within $O(n^{\frac{1}{3}-\epsilon})$ for any $\epsilon > 0$.*

Proof: The construction is the same as in the proof of Theorem 2.4, except that the edges are undirected and we replace the "modules" (gray box intersections) that consist of the same instance of the 2DIRPATH problem by parameterized instances of the Undir-Node-USF problem, described below. An instance of the Undir-Node-USF problem is an undirected graph $G(V, E)$ with distinct vertices $x_1, y_1, x_2, y_2 \in V$, plus two integers $p_2 > p_1 \geq 0$. Furthermore, each node v of G has a priority p_v. We may assume that G has an x_2, y_2 path such that each vertex v on this path has $p_v \geq p_2$ (this follows from the construction of Guruswami et al. [11]). The question is whether or not there exist two vertex-disjoint paths Q_1, Q_2, such that Q_i (for $i = 1, 2$) starts at x_i, ends at y_i, and every node v of Q_i has priority $p_v \geq p_i$. Guruswami et al. [11] (see Theorem 3 in their paper) proved the following result (by giving a reduction from the satisfiability problem): Given an instance of Undir-Node-USF, it is NP-complete to decide whether the answer is "Yes" or "No". Moreover, this holds for any two distinct integers p_2, p_1. (We remark that our notation differs from that of [11]; they use the terms "node capacities c_v", and "source-sink pairs (s_i, t_i)" with "demands d_i" whereas we use "node priorities p_v", and we have two pairs (x_1, y_1) with priority p_1 and (x_2, y_2) with priority p_2; there is no other difference.)

In the proof of Theorem 2.4 we make the following changes. We fix $N = |V(G)|^{\frac{1}{\epsilon}}$, where G is the "module" graph (instance of Undir-Node-USF). The terminals $b_1, \ldots, b_N$ are given distinct priorities, say, $1, \ldots, N$. We remove all the edges $a_i b_i$, $i = 1, \ldots, N$. For each gray box intersection with vertices $s^i_{\alpha\beta}, t^i_{\alpha\beta}, p^{\alpha}_{\beta i}, q^{\alpha}_{\beta i}$ we identify vertices x_2, y_2 with $s^i_{\alpha\beta}, t^i_{\alpha\beta}$ (horizontal line) and fix priority $p_2 = N$ (to connect b_N to the root), and we identify vertices x_1, y_1 with $p^{\alpha}_{\beta i}, q^{\alpha}_{\beta i}$ (vertical line) and fix priority $p_1 = \beta$

(to connect b_β to the root). All the nodes v that are not in the interior of any "module" get priority $p_v = N$.

It can be seen that Lemma 2.5 in the proof of Theorem 2.4 applies to the new setting (for **IUV**-priority). Now, consider any priority Steiner tree T_q. Every vertex in the r, b_N path (of T_q) must have priority N, thus this path cannot contain any "vertical line segments" (paths containing edges of the form $p^\alpha_{\beta i} q^\alpha_{\beta(i+1)}$); that is, this path corresponds to one of the horizontal lines H^i. Using this, it can be seen that Lemma 2.6 in the proof of Theorem 2.4 applies to the new setting (for **IUV**-priority). This proves Theorem 3.15. ∎

References

[1] S. Arora and C. Lund, *Hardness of approximations*, in *Approximation Algorithms for NP-hard Problems*, Dorit Hochbaum Ed., PWS Publishing, 1996.

[2] A. Baveja and A. Srinivasan, *Approximation algorithms for disjoint paths and related routing and packing problems*, Mathematics of Operations Research 25:255-280, 2000. Earlier version in FOCS1997.

[3] R. Carr, and S. Vempala, *Randomized metarounding*, Random Structures and Algorithms, 20:343-352, 2002. Earlier version in STOC2000.

[4] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, *Approximation algorithms for directed Steiner problem*, J.Algorithms 33(1):73-91, 1999. Earlier version in SODA1998.

[5] M. Charikar, J. Naor, and B. Schieber, *Resource optimization in QoS multicast routing of real-time multimedia*, Proc. 19th Annual IEEE INFOCOM (2000).

[6] J. Cheriyan and M.R. Salavatipour, *Packing Steiner-node-disjoint Steiner trees*, manuscript, 2004.

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

[8] U. Feige, M. Halldorsson, G. Kortsarz, and A. Srinivasan, *Approximating the domatic number*, Siam J.Computing 32(1):172-195, 2002. Earlier version in STOC 2000.

[9] P. Floréen, P. Kaski, J. Kohonen, and P. Orponen, *Multicast time maximization in energy constrained wireless networks*, in Proc. 2003 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC2003), San Diego, CA, 2003.

[10] A. Frank, T. Király, M. Kriesell, *On decomposing a hypergraph into k connected sub-hypergraphs*, Discrete Applied Mathematics 131(2):373-383, 2003.

[11] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, *Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems*, J.Computer and System Sciences 67(3):473-496, 2003. Earlier version in STOC1999.

[12] S. Fortune, J. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, Theoretical Computer Science 10(2):111-121, 1980.

[13] M. Grötschel, A. Martin, and R. Weismantel, *The Steiner tree packing problem in VLSI design*, Mathematical Programming 78:265-281, 1997.

[14] S. Guha and S. Khuller, *Improved methods for approximating node weighted Steiner trees and connected dominating sets*, Information and Computation 150:57-74, 1999. Earlier version in FST&TCS1998.

[15] E. Hazan, S. Safra, and O. Schwartz, *On the hardness of approximating k-dimensional matching*, Electronic Colloqium on Computational Complexity, Rep.No.20, 2003.

[16] K. Jain, M. Mahdian, M.R. Salavatipour, *Packing Steiner trees*, Proc. SODA2003.

[17] V. Kann, *Maximum bounded 3-dimensional matching is MAX SNP-complete*, Information Processing Letters 37:27-35, 1991.

[18] P. Kaski, *Packing Steiner trees with identical terminal sets*, Information Processing Letters 91(1):1-5, 2004.

[19] S.G. Kolliopoulos and C. Stein, *Approximating disjoint-path problems using packing integer programs*, Mathematical Programming 99:63-87, 2004. Earlier version in Proc. IPCO1998.

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

[21] L. Lau, *An approximate max-Steiner-tree-packing min-Steiner-cut theorem*, Proc. 45th IEEE FOCS2004.

[22] A. Martin and R. Weismantel, *Packing paths and Steiner trees: Routing of electronic circuits*, CWI Quarterly 6:185-204, 1993.

[23] S. Vempala, and B. Vöcking, *Approximating multicast congestion*, Proc. 10th ISAAC, Chennai, Springer LNCS, 367-372, 1999.