Approximation Algorithms and Hardness Results for Packing Element-Disjoint Steiner Trees in Planar Graphs

A. Aazami [*] J. Cheriyan [†] K. R. Jampani [‡]

March 2, 2011

Abstract

We study the problem of packing element-disjoint Steiner trees in graphs. We are given a graph and a designated subset of terminal nodes, and the goal is to find a maximum cardinality set of element-disjoint trees such that each tree contains every terminal node. An *element* means a non-terminal node or an edge. (Thus, each non-terminal node and each edge must be in at most one of the trees.) We show that the problem is APX-hard when there are only three terminal nodes, thus answering an open question.

Our main focus is on the special case when the graph is planar. We show that the problem of finding two element-disjoint Steiner trees in a planar graph is NP-hard. Similarly, the problem of finding two edge-disjoint Steiner trees in a planar graph is NP-hard. We design an algorithm for planar graphs that achieves an approximation guarantee close to 2. In fact, given a planar graph that is k element-connected on the terminals (k is an upper bound on the number of element-disjoint Steiner trees), the algorithm returns $\lfloor \frac{k}{2} \rfloor - 1$ element-disjoint Steiner trees. Using this algorithm, we get an approximation algorithm for the edge-disjoint version of the problem on planar graphs that improves on the previous approximation guarantees. We also show that the natural LP relaxation of the planar problem has an integrality ratio approaching 2.

Keywords: Steiner trees; packing; approximation algorithms; hardness of approximation; NP-hard; planar graphs; element connectivity; edge connectivity; partition connectivity.

1 Introduction

In the Steiner Tree Packing problem we are given an (undirected) graph $G = (V, E)$ and a subset of nodes $R \subseteq V$; each node in R is called a terminal node, and each node in $V - R$ is called a *Steiner* node or a non-terminal node. A Steiner node or an edge is called an *element*. A tree that contains all terminal nodes in R is called an R-Steiner tree (or Steiner tree, for short). The goal is to find a set of element-disjoint R-Steiner trees of maximum cardinality; that is, find as many R-Steiner trees as possible such that each Steiner node and each edge is in at most one of the trees. Our main focus is on approximation algorithms and hardness results for this problem. There is an important version of the problem that we call the

[*]Dept. of Comb. & Opt., U. Waterloo, Waterloo ON Canada N2L 3G1. Email: aaazami@uwaterloo.ca.
[†]Dept. of Comb. & Opt., U. Waterloo, Waterloo ON Canada N2L 3G1. Email: jcheriyan@uwaterloo.ca.
[‡]Dept. of Comp. Sci., U. Waterloo, Waterloo ON Canada N2L 3G1. Email: krjampani@uwaterloo.ca.

EDGE-DISJOINT STEINER TREE PACKING problem; here, the goal is to find a set of edge-disjoint R-Steiner trees of maximum cardinality; that is, find as many R-Steiner trees as possible such that each edge is in at most one of the trees.

1.1 Previous literature

Consider the special case of the STEINER TREE PACKING problem where all of the nodes are terminal nodes (i.e., $R = V$). Then the problem is the same as finding a maximum-cardinality set of edge-disjoint spanning trees. Tutte [31] and Nash-Williams [26] independently proved the following min-max theorem for this special case: An undirected graph G has k edge-disjoint spanning trees if and only if for any partition $\mathcal{P}$ of V into $|\mathcal{P}|$ non-empty subsets we have $e(\mathcal{P}) \geq k(|\mathcal{P}| - 1)$, where $e(\mathcal{P})$ is the number of edges in G with end-nodes in different sets of $\mathcal{P}$. Frank, Kiraly, and Kriesell [8] extended this result to hypergraphs via the notion of partition-connectivity (see Section 2 for details): A hypergraph $\mathcal{H}$ decomposes into k hyperedge-disjoint partition-connected hypergraphs if and only if the partition-connectivity of $\mathcal{H}$ is at least k.

We say that the set of terminals R is *k-element connected* if there exist k element-disjoint paths between every pair of nodes in R; that is, for any two nodes $s, t \in R$, there exist k paths between s and t such that each element occurs in at most one of these k paths. Similarly, we say that the set of terminals R is *k-edge connected* if there exist k edge-disjoint paths between every pair of nodes in R. We use n to denote the number of nodes in the input graph. Also, we call the terminal nodes *black* nodes, and the non-terminal nodes *white* nodes. An edge between two white nodes is called a *white* edge.

Kaski [19] proved that the problem of finding two edge-disjoint Steiner trees is NP-hard, and also showed that the EDGE-DISJOINT STEINER TREE PACKING problem is NP-hard even with 7 terminals. The problem was proved to be APX-hard even with 4 terminals in [3]. Jain, Mahdian, and Salavatipour [17] presented an approximation algorithm with a guarantee of $O(|R|)$. Later, Lau [22, 23], using the result of Frank et al. [8], proved that if the terminals are $24k$-edge connected, then there exist k edge-disjoint Steiner trees, and he gave an approximation algorithm with a guarantee of 24.

Cheriyan and Salavatipour [4] studied the element-disjoint STEINER TREE PACKING problem; they observed that the problem is hard to approximate within a factor of $\Omega(\log n)$, and they designed a randomized approximation algorithm with a guarantee of $O(\log n)$. Subsequently, Calinescu, Chekuri and Vondrak [1] designed a simpler algorithm with a similar approximation guarantee, and also, they derandomized their algorithm.

To the best of our knowledge, the above results gave the best results for planar graphs, before the results in this paper were obtained. In other words, the previous best approximation guarantee for packing edge-disjoint Steiner trees in planar graphs was 24, [23], and for packing element-disjoint Steiner trees in planar graphs was $O(\log n)$, [4, 1].

There have been some recent developments; we were not aware of these results until after we completed our research and submitted our manuscript for journal publication. Chekuri and Korula [2] obtained some related results using different techniques; in particular, they design an $O(1)$-approximation algorithm for the problem of packing element-disjoint Steiner forests in planar graphs. Demaine, Hajiaghayi, and Klein [5] obtained some results on minimum-cost node-weighted Steiner trees in planar graphs, including a 6-approximation algorithm ; this implies a 6-approximation algorithm for *fractionally* packing element-disjoint Steiner trees in planar graphs, even with capacities on the elements. West and Wu [33] have improved on the results of Lau [23] by showing that there exist k edge-disjoint Steiner trees if the terminals are $6.5k$-edge connected.

To the best of our knowledge, the systematic study of problems of this type was started by Grötschel et al., see [12, 10, 13, 11, 14]. They were motivated by applications in VLSI circuit design, see [14, 24]. They focused on a generalization of the Edge-disjoint Steiner Tree Packing problem, where we are given a list of terminal sets, $R_1, R_2, R_3, \ldots, R_q$ and the goal is to find edge-disjoint Steiner trees $T_1, T_2, T_3, \ldots, T_q$ such that T_i contains (and connects) all the terminal nodes in R_i, for $i = 1, \ldots, q$. Their problem is quite different from the problems of interest to us, and is more general; for example, their problem contains the Edge-disjoint Paths problem as a special case, namely, the special case where each terminal set R_i has size two. Consequently, any hardness result that applies to the Edge-disjoint Paths problem applies also to the Generalized edge-disjoint Steiner Tree Packing problem of Grötschel et al., but those hardness results may not apply to the problems of interest to us. Further results and applications of the generalized problem are discussed by Wagner [32] and Korte et al., [20], but note that the NP-hardness results in [20] (where different Steiner trees have different terminal sets) do not apply to the problems of interest to us. The main focus of the work on the generalized problem was to obtain computational procedures for finding an optimal solution, based on mathematical programming. Some algorithmic results on the generalized problem are presented by Wagner [32], but those results are "disjoint" from our results.

The generalized problem has other well-known applications including multicasting in wireless networks [7], and broadcasting large data streams, such as videos, over the Internet [17].

1.2 Results in this paper

Our focus is on approximation algorithms and hardness results for the element-disjoint Steiner Tree Packing problem on planar graphs. We call this the Planar Steiner Tree Packing problem. Our main results are as follows:

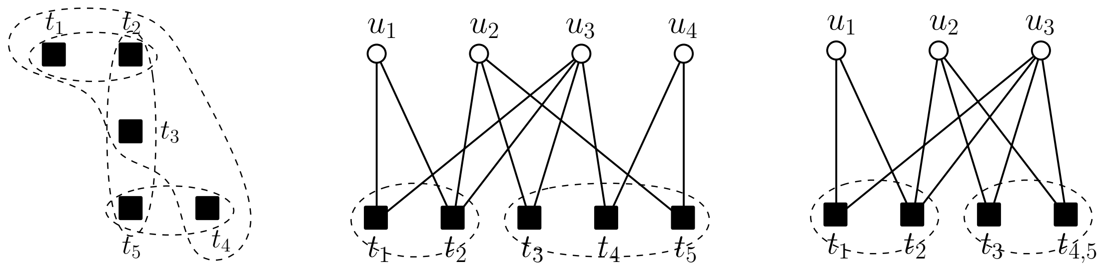
- In Section 2, we present an approximation algorithm with a guarantee of (almost) 2 for the Planar Steiner Tree Packing problem; more precisely, given a planar graph and a set of terminal nodes R such that R is k-element connected, our algorithm finds at least $\min(1, \lfloor \frac{k}{2} \rfloor - 1)$ element-disjoint Steiner trees; here, k is a positive integer. Based on this, we get an approximation algorithm with a guarantee of (almost) 4 for the edge-disjoint version of the problem on planar graphs. To the best of our knowledge, this improves on the known approximation guarantees for the Edge-disjoint Steiner Tree Packing problem on planar graphs. The planarity of the graph is used at only one point in our analysis, and there we use the upperbound on the number of edges in a planar bipartite simple graph. Our methods extend to larger classes of graphs, namely, graphs that exclude a fixed minor, to give approximation guarantees that depend on the order of the forbidden minor.

  We conjecture that a planar graph that is k-element connected on the terminals has at least $\lfloor \frac{k}{2} \rfloor$ element-disjoint Steiner trees.

- In Section 3, we prove that the Steiner Tree Packing problem is APX-hard even with three terminals (i.e., $|R| = 3$). This answers an open question in the literature, see Floréen, et al. [7, Page 119].

  Then, we show that the problem of finding two element-disjoint Steiner trees in a planar graph is NP-hard. An immediate implication is that one cannot improve on the approximation guarantee of 2 for the Planar Steiner Tree Packing problem without further assumptions. We extend our construction and proof to show that the problem of finding two edge-disjoint Steiner trees in a planar graph is NP-hard.

- In Section 4, we show that even on planar graphs the standard LP (linear programming) relaxation of the element-disjoint Steiner Tree Packing problem has an integrality ratio $\geq 2 - \frac{2}{|R|} - \epsilon$, where the additive term ϵ is a function of $|R|$ and the element-connectivity of the terminals, k, and for fixed



(a) Hypergraph (b) Bipartite representation (c) Contracting $\{t_4, t_5\}$

Figure 1: A Hypergraph and its bipartite representation

$|R|$, $\epsilon \to 0$ as $k \to \infty$. Our approximation guarantee of (almost) 2 for planar graphs (mentioned above) implies that the integrality ratio on planar graphs approaches 2 as $k \to \infty$.

The significance of our lower bound on the integrality ratio comes from the fact that the optimal value of this LP relaxation gives the best upper bound known (as far as we know) on the maximum number of element-disjoint Steiner trees. Thus, for planar graphs, our result shows that the approximation guarantee of 2 cannot be improved by any algorithm or analysis that relies on an upper bound that is dominated by the LP bound.

Moreover, we modify our construction to get a similar lower bound on the integrality ratio for the edge-disjoint version of the problem on planar graphs.

1.3 Hypergraphs: notation and definitions

This subsection has a few definitions pertaining to hypergraphs; these are used in Section 2.1.

A hypergraph is a pair $\mathcal{H} = (V, \mathcal{E})$ where V is the node-set of $\mathcal{H}$ and $\mathcal{E}$ is a collection of non-empty subsets of V. A subset $Z \in \mathcal{E}$ is called a *hyperedge* of $\mathcal{H}$. Given a partition $\mathcal{P} = \{V_1, \ldots, V_t\}$ of V into non-empty subsets, a hyperedge $Z \in \mathcal{E}$ is called a *crossing* hyperedge if it intersects at least two subsets of $\mathcal{P}$ and otherwise it is called an *internal* hyperedge. We use $|\mathcal{P}|$ to denote the number of sets V_i in $\mathcal{P}$, and we denote the number of crossing hyperedges corresponding to the partition $\mathcal{P}$ by $e_{\mathcal{H}}(\mathcal{P})$ (or simply, by $e(\mathcal{P})$).

Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, we associate a bipartite graph $G_{\mathcal{H}} = (V, U; E)$ to $\mathcal{H}$ as follows. Corresponding to each hyperedge $Z \in \mathcal{E}$ we have a node $u_Z \in U$. A node $v \in V$ is adjacent to $u_Z \in U$ if $v \in Z$; note that the degree of u_Z in $G_{\mathcal{H}}$ is the size of Z.

Consider the hypergraph $\mathcal{H}$ shown in Figure 1(a). The node-set of $\mathcal{H}$ is $V = \{t_1, t_2, t_3, t_4, t_5\}$, and the hyperedges of $\mathcal{H}$ are $Z_1 = \{t_1, t_2\}$, $Z_2 = \{t_2, t_3, t_5\}$, $Z_3 = \{t_1, t_2, t_3, t_4\}$ and $Z_4 = \{t_4, t_5\}$. Figure 1(b) shows the bipartite graph, $G = (V, U; E)$, associated with $\mathcal{H}$. Consider the partition $\mathcal{P} = \{\{t_1, t_2\}, \{t_3, t_4, t_5\}\}$ of V; this partition is shown in dashed lines in Figure 1(b). The hyperedges Z_2, Z_3 are crossing hyperedges w.r.t. (with respect to) $\mathcal{P}$, and hyperedges Z_1, Z_4 are internal hyperedges w.r.t. $\mathcal{P}$. Thus, $e(\mathcal{P}) = 2$, since there are two crossing hyperedges in $\mathcal{P}$. Given a partition $\mathcal{P}$, a useful operation is to *contract* an internal hyperedge: we identify all nodes in Z into a single node and remove Z from the hypergraph. For example, Figure 1(c) shows the bipartite representation of the hypergraph obtained by contracting the internal hyperedge $Z_4 = \{t_4, t_5\}$. If we further contract $Z_1 = \{t_1, t_2\}$ we get a copy of $K_{2,3}$. If we contract some internal hyperedges (w.r.t. $\mathcal{P}$) of $\mathcal{H}$, then we obtain a "shrunk" hypergraph $\mathcal{H}'$ and a partition $\mathcal{P}'$ of $V(\mathcal{H}')$; note that the crossing hyperedges of $\mathcal{H}$ (w.r.t. $\mathcal{P}$) are the same as the crossing hyperedges of $\mathcal{H}'$ (w.r.t. $\mathcal{P}'$).

2 Approximation algorithms

2.1 Element-disjoint Steiner trees

We present an approximation algorithm for packing element-disjoint Steiner trees in planar graphs that achieves an approximation guarantee close to 2 (details below). Our method consists of two steps. First, we transform to a planar bipartite graph, while preserving the terminals and their element-connectivity. Then, we view the bipartite graph as a hypergraph, and apply a method of Frank et al. [8] to decompose the set of hyperedges $\mathcal{E}$ into a number of disjoint sets $\mathcal{E}_1, \mathcal{E}_2, \ldots$ such that each set $\mathcal{E}_i$ induces a Steiner tree of our bipartite graph. Each of these "bipartite" Steiner trees transforms back to a Steiner tree of the original graph. The planarity of the graph is used at only one point in our analysis, and there we use the upperbound on the number of edges in a planar bipartite simple graph. Our methods extend to larger classes of graphs, namely, graphs that exclude a fixed minor, to give approximation guarantees that depend on the order of the forbidden minor.

The following theorem is the main result of this section.

Theorem 2.1 *Let $G = (V, E)$ be an undirected planar graph, let $R \subseteq V$ be the set of terminals, and assume that R is k-element connected. Then there are at least $\left\lfloor \frac{k}{2} \right\rfloor - 1$ element-disjoint Steiner trees in G. Moreover, there is an algorithm with a running time of $O(|V|^{4.5})$ that finds at least $\left\lfloor \frac{k}{2} \right\rfloor - 1$ element-disjoint Steiner trees in G.*

We define the Bipartite Steiner Tree Packing problem to be a subproblem of the element-disjoint Steiner Tree Packing problem such that the graph is bipartite, all terminal nodes are in one part of the bipartition, and all Steiner nodes are in the other part. Consider a planar instance of the element-disjoint Steiner Tree Packing problem, i.e., the associated graph is planar. We can transform it into a planar instance of Bipartite Steiner Tree Packing by using the following theorem. The theorem is due to Hind and Oellermann, see [16], and a short proof is given in [4].

Theorem 2.2 *[16] Consider a graph $G = (V, E)$ that has a set of terminals R such that R is k-element connected. There is a polynomial-time algorithm that repeatedly deletes or contracts white edges to obtain a bipartite graph G' from G such that R stays k-element connected, and moreover, R forms one part of the bipartition of G'.*

Let $G = (R, U; E)$ be an instance of the Bipartite Steiner Tree Packing problem, where R is the set of terminal nodes and U is the set of Steiner nodes. Recall the definitions and notation for hypergraphs from Section 1.3. We associate a hypergraph $\mathcal{H}_G$ to G as follows. The node set of $\mathcal{H}_G$ is given by the set of terminal nodes of G, that is, $V(\mathcal{H}_G) = R$; moreover, corresponding to each Steiner node $u \in U$ of G, $\mathcal{H}_G$ has a hyperedge that we denote by Z_u, where Z_u is the subset of $V(\mathcal{H}_G)$ consisting of the neighbors of u in G, that is, $Z_u = \{v \in R \mid \{u, v\} \in E(G)\}$; thus, $\mathcal{H}_G = (R, \mathcal{E})$ where $\mathcal{E} = \{Z_u \mid u \in U\}$.

Moreover, given any hypergraph $\mathcal{H}$, we may view its associated bipartite graph $G_{\mathcal{H}}$ as an instance of the Bipartite Steiner Tree Packing problem, where $V(\mathcal{H}_G)$ gives the set of terminal nodes of the latter instance and the nodes of $G_{\mathcal{H}}$ corresponding to the hyperedges of $\mathcal{H}$ give the Steiner nodes of the latter instance.

Recall that we denote a partition of V into non-empty subsets by $\mathcal{P} = \{V_1, \ldots, V_t\}$, and we use $|\mathcal{P}|$ to denote the number of sets V_i in $\mathcal{P}$. A hypergraph $\mathcal{H}$ is called *k-partition connected* if $e_{\mathcal{H}}(\mathcal{P}) \geq k(|\mathcal{P}| - 1)$ for every partition $\mathcal{P}$ of V. A 1-partition connected hypergraph is simply called *partition-connected*. If a hypergraph $\mathcal{H}$ is partition-connected, then it is easy to see that the associated bipartite graph $G_{\mathcal{H}}$ is connected, and so it contains a Steiner tree (with terminal set $V(\mathcal{H})$). But the converse does not hold:

for a connected instance of Bipartite Steiner Tree Packing, the associated hypergraph may not be partition-connected. Frank et al. [8] proved the following generalization of the Tutte–Nash-Williams theorem.

Theorem 2.3 (Theorem 2.8 in [8]) *A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is k-partition connected if and only if $\mathcal{E}$ partitions into k subsets $\mathcal{E}_1, \ldots, \mathcal{E}_k$ such that each of the sub-hypergraphs $\mathcal{H}_i = (V, \mathcal{E}_i)$ is partition-connected.*

Therefore, we can obtain ℓ element-disjoint Steiner trees in G if $\mathcal{H}_G$ is ℓ-partition connected. Now we prove the following lemma that completes the proof of Theorem 2.1.

Lemma 2.4 *Let $G = (R, U; E)$ be a bipartite planar graph such that R is k-element connected. Then the hypergraph $\mathcal{H}_G = (R, \mathcal{E})$ associated with G is $\lfloor \frac{k-2}{2} \rfloor$-partition connected.*

Proof: We may assume that G is connected. Consider the hypergraph $\mathcal{H}$ and define the *fractional* partition-connectivity, λ^*, as follows:

$$\lambda^* = \min_{\mathcal{P}} \frac{e(\mathcal{P})}{|\mathcal{P}| - 1}, \tag{1}$$

where the minimum is over all partitions $\mathcal{P}$ of R with $|\mathcal{P}| \geq 2$. Let λ denote the partition-connectivity of $\mathcal{H}$. It follows from the definition of partition-connectivity that $\lambda = \lfloor \lambda^* \rfloor$. Let $\mathcal{P}^* = \{X_1, X_2, \ldots, X_\ell\}$ be a partition that achieves the minimum ratio λ^*. In the rest of the proof, except where mentioned otherwise, crossing hyperedges and internal hyperedges are w.r.t. $\mathcal{P}^*$.

Consider the Steiner nodes of G that correspond to the internal hyperedges. We contract all the edges of G that are incident to these Steiner nodes, and we call the resulting graph G'. In more detail, consider each internal hyperedge $Z_u \in \mathcal{E}$ and contract all edges in G adjacent to the Steiner node u corresponding to hyperedge Z_u. We may ignore all parallel edges in G' formed by these edge contractions.

Claim 2.5 *The obtained graph G' is a bipartite planar graph and has the following properties:*

1. *All of the remaining Steiner nodes in G' correspond to crossing hyperedges in $\mathcal{H}$, and they form one part of the bipartition,*
2. *The other part of the bipartition has $|\mathcal{P}^*|$ nodes, and each node has degree at least k.*

Proving this claim completes the lemma. This follows because G' has at least $k\,|\mathcal{P}^*|$ edges and at most $2(e(\mathcal{P}^*) + |\mathcal{P}^*|) - 4$ edges since it is a bipartite planar graph and each planar bipartite graph on n nodes has at most $2n - 4$ edges. Hence, we have

$$k\,|\mathcal{P}^*| \leq 2(e(\mathcal{P}^*) + |\mathcal{P}^*|) - 4 \Longrightarrow e(\mathcal{P}^*) \geq \frac{(k-2)\,|\mathcal{P}^*|}{2} + 2 \Longrightarrow \lambda^* > \frac{k-2}{2}.$$

Proof of Claim 2.5: Consider a set $X_i \in \mathcal{P}^*$ of size at least 2 and arbitrarily partition it into two non-empty sets X_i' and X_i'', and let $\mathcal{P}'$ be the obtained partition. Since $\mathcal{P}^*$ is the minimum ratio partition, we have $\lambda' = \frac{e(\mathcal{P}')}{|\mathcal{P}'|-1} \geq \lambda^*$. Hence, $e(\mathcal{P}') \geq \lambda^*(|\mathcal{P}'|-1) > \lambda^*(|\mathcal{P}^*|-1) = e(\mathcal{P}^*)$. Hence, there exists a hyperedge that is crossing w.r.t. $\mathcal{P}'$ but is not crossing w.r.t. $\mathcal{P}^*$; that is, one of the internal hyperedges w.r.t. $\mathcal{P}^*$ intersects both X_i' and X_i''. This reasoning applies to each set $X_i \in \mathcal{P}^*$ and to each 2-partition X_i', X_i'' of X_i; hence, for each $X_i \in \mathcal{P}^*$, the subgraph of G induced by X_i and the Steiner nodes corresponding to the hyperedges internal to X_i is connected. Thus, contracting all edges in G adjacent to the Steiner nodes corresponding to the internal hyperedges (w.r.t. $\mathcal{P}^*$) will shrink each set X_i of $\mathcal{P}^*$ into a single node. The obtained graph G' is planar, and it is easy to see that it is bipartite with all the Steiner nodes corresponding to the crossing hyperedges (w.r.t $\mathcal{P}^*$) in one part of the partition and all of the "contracted" nodes in the other part. Now we prove that the degree of each contracted node is at least k using the fact that the terminals are k-element connected in G. To see this, consider a shrunk node v_i corresponding to a subset $X_i \in \mathcal{P}^*$, and assume that it has less than k neighbors in G'. Let Y' be the set of neighbors of v_i, so $|Y'| < k$. Note that Y' separates v_i from any other contracted node v_j in G', i.e., v_i and v_j are in different connected components of $G' \setminus Y'$. Now focus on the original hypergraph $\mathcal{H}$ and note that Y' (viewed as a subset of $\mathcal{E}(\mathcal{H})$) contains all hyperedges that intersect both X_i and $R \setminus X_i$; thus, in the original graph G, we see that Y' (viewed as a subset of U) separates X_i from the rest of the terminals, because Y' contains all the Steiner nodes that are adjacent to both X_i and $R \setminus X_i$. This is a contradiction because the terminals are k-element connected in G; that is, for any set of white nodes Y whose deletion separates a pair of terminals, we must have $|Y| \geq k$. This shows that each contracted node has degree at least k in G'. □

This completes the proof the theorem. □

We give a formal outline of the algorithm, followed by an analysis of the running time.

Algorithm:

1. In the first step, we reduce the given graph $G = (V, E)$ to an instance G' of the Bipartite Steiner Tree Packing problem using Theorem 2.2; note that G' is obtained from G by removing or contracting white edges.

2. In the second step, we decompose the associated hypergraph $\mathcal{H}$ of G' into the maximum number of hyperedge-disjoint partition-connected sub-hypergraphs, by using results of Frank et al. [8] and Edmonds' matroid partition algorithm [6]. The independence test in this algorithm checks whether a given hypergraph is a hyperforest. (See the running time analysis given below for more discussion.)

3. Each partition-connected sub-hypergraph corresponds to a Steiner tree in G'. By "uncontracting" the edges that were contracted in the first step of the algorithm, we obtain a set of element-disjoint Steiner trees in G. If the given graph G is planar, and k-element connected on the terminals, then the algorithm returns $\geq \lfloor \frac{k}{2} \rfloor - 1$ element-disjoint Steiner trees.

To analyse the running time of the algorithm, consider the first step in detail. We take a white edge e and delete it from G. Then we check if the terminals are still k-element connected. If this holds, then we take another white edge and continue, otherwise, we identify the end-nodes of e and move to the next white edge. We can test for k-element connectivity in planar graphs in time $O(kn\,|R|)$ using the augmenting-paths algorithm for the maximum s-t flow problem. Hence, the total running time of the first step is $O(kn^2\,|R|)$, since the number of white edges is $O(n)$.

In the second step, we decompose $\mathcal{H}$ into the maximum number of hyperedge-disjoint partition-connected sub-hypergraphs via Theorem 2.3. The proof of this theorem is based on Edmonds' matroid partition theorem, and the proof can be "implemented" via the matroid partition algorithm [6]. The running time of the matroid partition algorithm is $O(p^{2.5} f(p))$, where p denotes the size of the ground set of the matroid, and $f(p)$ denotes the running time for testing independence in the given matroid. The test for independence corresponds to testing whether a hypergraph satisfies the conditions for a hyperforest, and this in turn corresponds to testing whether a bipartite graph $(R, U; E)$ has positive surplus, i.e., $\forall S \subseteq U, S \neq \emptyset : |\Gamma(S)| > |S|$; we mention that U corresponds to a subset of the ground set of the matroid. The positive surplus condition holds if and only if the size of a maximum matching is $|U|$ and every node $v \in R$ is non-critical, i.e., there is a maximum matching avoiding v; we can test for this via the Gallai-Edmonds decomposition in a running time of $O(|E|\,(|U|+|R|))$. Thus $f(p) = O(p^3)$ for an arbitrary bipartite graph, and for planar bipartite graphs this improves to $f(p) = O(p^2)$. Hence, using the fact that $p \leq n$, the running time of our algorithm on planar graphs is $O(kn^2\,|R| + n^{4.5}) = O(n^{4.5})$.

We have planar examples showing that the analysis in Lemma 2.4 is tight. Construct a planar bipartite graph $G = (R, U; E)$ as follows: Let $R = \{t_1, \cdots, t_d\}$ be the set of terminal nodes. For each $i \in \{1, \cdots, d-1\}$, the node set U contains k' nodes of degree 2, where each one is adjacent to t_i and t_{i+1}. In addition, the set U contains k' nodes of degree 2 where each one is adjacent to t_d and t_1. Finally, we add two nodes of degree d to U, each of which is adjacent to all the terminal nodes in R.

Note that G has k' element-disjoint cycles, such that each cycle contains all the terminal nodes, and moreover, the nodes on each cycle are alternately Steiner nodes and terminal nodes. Each one of these k' cycles corresponds to a 1-partition connected subhypergraph in $\mathcal{H}_G$. Note that these subhypergraphs are hyper-edge disjoint. Hence, by Theorem 2.3, which is due to Frank et al. [8], the partition-connectivity of $\mathcal{H}_G$ is at least k'. Let $\mathcal{P}$ be the partition of R into singletons, i.e., $\mathcal{P}$ has a set for each node in R. Also let λ be the partition-connectivity of $\mathcal{H}_G$. Then, assuming that $k' < d - 3$, we have

$$\lambda \leq \left\lfloor \frac{e(\mathcal{P})}{|\mathcal{P}| - 1} \right\rfloor = \left\lfloor \frac{k'd + 2}{d - 1} \right\rfloor = k' + \left\lfloor \frac{k' + 2}{d - 1} \right\rfloor = k'$$

It is clear from the construction that the element-connectivity of G is $2k'+2$, while the partition-connectivity of its corresponding hypergraph is k'. This shows the tightness of Lemma 2.4.

2.2 Edge-disjoint Steiner trees

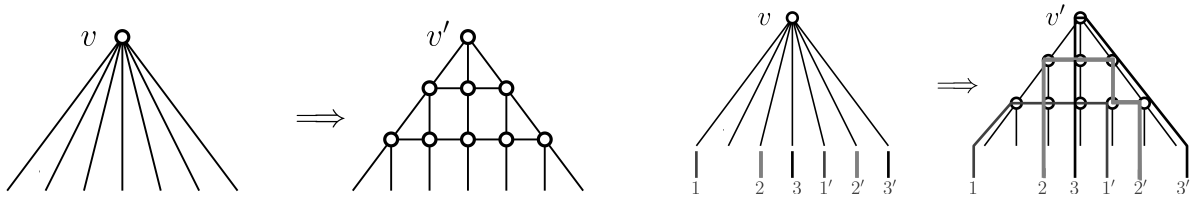
The above result extends to the packing of edge-disjoint Steiner trees in planar graphs, to give the following result.

Theorem 2.6 *Let $G = (V, E)$ be an undirected planar graphs, let $R \subseteq V$ be the set of terminals, and assume that R is k-edge connected. Then there are at least $\lfloor \frac{k}{4} \rfloor - 1$ edge-disjoint Steiner trees in G. Moreover, there is an algorithm with a running time of $O(|V|^{4.5})$ that finds at least $\lfloor \frac{k}{4} \rfloor - 1$ edge-disjoint Steiner trees in G.*

Proof: We first reduce G to a planar graph G' with Steiner nodes of degree at most 4. This is done by repeatedly replacing a Steiner node of degree more than 4 by a gadget that preserves the edge-connectivity and planarity, but doesn't introduce any new Steiner nodes of degree more than 4. Let v be a Steiner node of degree $d > 4$ in G. We replace node v by the gadget shown in Figure 2(a).

The gadget has $\lceil \frac{d}{2} \rceil - 1$ rows including the last row containing a single node v'. Clearly, the obtained graph has one less Steiner node of degree more than 4; also, the set of terminal nodes and their degrees stay the same. Moreover, R is k edge-connected in the obtained graph. To show this, we claim that any set of edge-disjoint paths using edges incident to v can be rerouted via the gadget. We sketch a proof of this claim, although the gadget and its properties are well known, see [25, 28]. This can be proved by induction on the number of rows in the gadget. The base case is when there is only one row containing the single node v', and note that v' has degree at most 4. Clearly for this case the claim is true. Given a paring of edges used in the paths going through v, we first route one of the extreme pairs (i.e., the pair using the left most edge or the pair using the right most edge), say the left most pair, using the first horizontal row of the gadget. Next, we send the other pairs using the vertical edges to the next horizontal row. Some of the paths may need to be "shifted" to the left; for example, the path labeled $(2, 2')$ is shifted on the first row to the left in Figure 2(b). Finally, we inductively reroute the remaining paths.

We replace all Steiner nodes of degree more than 4 to get a planar graph G' with no Steiner node of degree more than 4 such that the terminal nodes are k-edge connected in G'. Also observe that edge-disjoint Steiner trees in G' can be transformed to edge-disjoint Steiner trees in G.



(a) Gadget for high degree Steiner nodes

(b) Routing the paths via the gadget

Figure 2: Gadget for reducing degree

We can assume that G' has no edge connecting two terminals; such an edge can be subdivided by introducing a Steiner node. Now we show that the terminal nodes in G' are $\lceil \frac{k}{2} \rceil$ element-connected. Let X be a set of white nodes of minimum cardinality whose deletion separates some two terminals, i.e., $G' \setminus X$ has at least two connected components $C_1, C_2, \ldots, C_\ell$ that each contains a terminal. There are at most $4\,|X|$ edges going out of X since each white node in X has degree ≤ 4. Hence, there is a component C_j that has at most $2\,|X|$ edges entering it, and consequently there is an edge-cut of size at most $2\,|X|$ that separates a pair of terminals. Since such a cut has size $\geq k$, we have $|X| \geq \frac{k}{2}$. This shows that terminal nodes are $\lceil \frac{k}{2} \rceil$ element-connected in G'. Now we apply Theorem 2.1 to G' and obtain $\lfloor \frac{k}{4} \rfloor - 1$ element-disjoint Steiner trees in G'. These Steiner trees are clearly edge-disjoint. Thus, G has at least $\lfloor \frac{k}{4} \rfloor - 1$ edge-disjoint Steiner trees. □

2.3 Element-disjoint Steiner trees in H-minor-free graphs

In the proof of Lemma 2.4, we used the fact that the number of edges in a planar graph is linear in the number of nodes; moreover, the proof uses edge contractions and edge deletions, and the fact that these operations preserve the planarity of the graph. These two facts about planar graphs are valid in graphs that exclude a fixed minor H. Kostochka [21] and Thomason [30] showed that every graph of average degree at least $cr\sqrt{\log_2 r}$ has a K_r minor, where $c \leq 324$. It follows from the Kostochka–Thomason result that an H-minor-free graph G has at most $c_H \cdot |V(G)|$ edges, where $c_H = \frac{c}{2}\,|V(H)|\,\sqrt{\log_2 |V(H)|}$. Thus, our analysis for planar graphs extends to give the following result.

Theorem 2.7 *Let H be a fixed graph. Let $G = (V, E)$ be an undirected graph that has no H minor, let $R \subseteq V$ be the set of terminals, and assume that R is k-element connected. Then there are at least $\left\lfloor \frac{k}{c_H} \right\rfloor - 1$ element-disjoint Steiner trees in G. Moreover, there is an algorithm with a running time of $O(n^{4.5} + k\,|R|\,c_H^2 n^2)$ that finds this number of element-disjoint Steiner trees in G.*

3 Hardness results

This section has four main results. In the first two subsections, we focus on instances with only 3 terminal nodes. We show that the edge-disjoint Steiner Tree Packing problem is NP-hard in planar graphs, even when there are 3 terminal nodes. Next, we focus on graphs in general (without the planarity restriction), and we show that the edge-disjoint Steiner Tree Packing problem with 3 terminal nodes is APX-hard; then we extend this to prove APX-hardness for the element-disjoint Steiner Tree Packing problem on three terminal nodes. This settles an open question in the literature, see [7, Page 119 second column]. In the third subsection, we show that the problem of finding two element-disjoint Steiner trees in a planar graph is NP-hard. In the last subsection, we extend this NP-hardness result to the setting of edge-disjoint Steiner trees.

3.1 Planar graphs with three terminal nodes

This section has a proof of the following result; our proof is based on a major recent result of Naves [27].

Theorem 3.1 *The edge-disjoint* Steiner Tree Packing *problem in planar graphs with* 3 *terminals is NP-hard even with all terminals on the outer face.*

We focus on planar graphs, and reduce the Edge-Disjoint Path problem with 2 demand pairs to the Steiner Tree Packing problem with 3 terminals. Let $\mathcal{I} = (G; x_1, y_1, d_1; x_2, y_2, d_2)$ be an instance of the planar Edge-Disjoint Path problem, where $G = (V, E)$ is a planar graph and $(x_1, y_1), (x_2, y_2)$ are demand pairs. Moreover, the end nodes of the demand pairs are incident to the outer face in a given planar embedding of G, and they occur in the order x_1, x_2, y_1, y_2 in the face. The goal is to find d_1 paths from x_1 to y_1 and d_2 paths from x_2 to y_2 such that all of the $d_1 + d_2$ paths are edge disjoint. Our result is based on the following major recent result of Naves [27, 28].

Theorem 3.2 (Theorem 9 in [27]) *The planar* Edge-Disjoint Path *problem with two demand pairs on the outer face is NP-hard.*

Reduction:

1. Start from a copy of G and add 3 terminal nodes $\{t, t_1, t_2\}$ and two non-terminal nodes s_1, s_2 to the outer face of G.
2. Add d_1 parallel edges from s_1 to each of t, t_2, x_1, and similarly we add d_2 parallel edges from s_2 to each of t, t_1, x_2.
3. Finally, we add d_1 parallel edges from y_1 to t_1, and d_2 parallel edges from y_2 to t_2. Note that the obtained graph is planar.
4. Let H be the obtained graph, and let $R = \{t, t_1, t_2\}$ (see Figure 3 for an illustration).

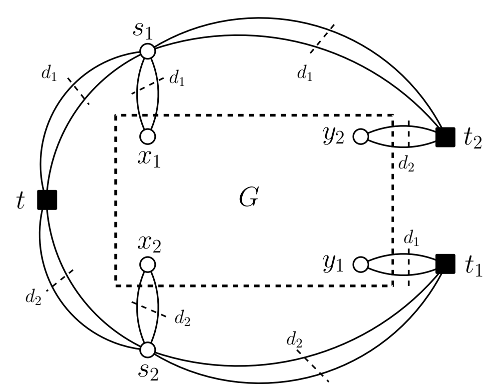


Figure 3: Planar graph with 3 terminals on the outer face

The following lemma completes the proof of Theorem 3.1.

Lemma 3.3 *The planar graph H has $d_1 + d_2$ edge-disjoint Steiner trees if and only if the* Edge-Disjoint Path *problem in G has a solution.*

Proof: Assume that the Edge-Disjoint Path problem in G has a solution; that is, there are d_1 edge-disjoint x_1-y_1 paths, and there are d_2 edge-disjoint x_2-y_2 paths, and all these d_1+d_2 paths are edge-disjoint. Observe that by adding edges $\{s_1, t\}$, $\{s_1, x_1\}$, $\{s_1, t_2\}$, $\{y_1, t_1\}$ to any x_1-y_1 path in G we get a Steiner tree. Hence, using d_1 edge-disjoint x_1-y_1 paths, we get d_1 edge-disjoint Steiner trees. Similarly, we can get d_2 edge-disjoint Steiner trees using d_2 edge-disjoint x_2-y_2 paths. Moreover, any two of these $d_1 + d_2$ Steiner trees are edge-disjoint. Thus, graph H has $d_1 + d_2$ edge-disjoint R-Steiner trees.

Now we prove the other direction. Suppose that H has $d_1 + d_2$ edge-disjoint Steiner trees. Let $\mathcal{F}$ denote this set of Steiner trees. The degree of each terminal in any one of the Steiner trees in $\mathcal{F}$ is one since each terminal has degree $d_1 + d_2$ in H. Therefore, each Steiner tree in $\mathcal{F}$ has a Steiner node of degree three. We claim that this node of degree three is either s_1 or s_2, and moreover, only one of s_1 or s_2 can be in any Steiner tree. Focus on the edges with exactly one end-node in $V(G)$; these are the edges crossing the dashed box in Figure 3. First we show that each Steiner tree T in $\mathcal{F}$ has exactly two of these crossing edges. If T has no crossing edges, then terminal t is forced to have degree two in T, which is not possible. Also note that T cannot have only one crossing edge since there are no terminal nodes in $V(G)$. This shows that T has at least two crossing edges. This implies that each Steiner tree in $\mathcal{F}$ has exactly two crossing edges, since there are $(d_1 + d_2)$ Steiner trees in $\mathcal{F}$ and $2(d_1 + d_2)$ crossing edges in total. Hence, for any Steiner tree in $\mathcal{F}$, the node of degree three cannot be in $V(G)$ (i.e., it cannot be inside the dashed box in Figure 3), so the node of degree three is either s_1 or s_2. Also since there are $3(d_1 + d_2)$ edges adjacent to s_1 and s_2 no tree contains both s_1 and s_2. Now consider a Steiner tree T in $\mathcal{F}$ containing s_1. Tree T uses the edge $\{s_1, x_1\}$ to enter G and it has no edges adjacent to s_2, so it must contain $\{y_1, t_1\}$. Hence, T contains an x_1-y_1 path inside G. Similarly, if T contains s_2, then it has an x_2-y_2 path inside G. Therefore, the $d_1 + d_2$ Steiner trees in $\mathcal{F}$ give us a solution to the instance $\mathcal{I}$ of the Edge-Disjoint Path problem in G. □

3.2 APX-hardness for general graphs with 3 terminal nodes

In this subsection, we focus on graphs in general, without the planarity restriction. We prove that the edge-disjoint Steiner Tree Packing problem with 3 terminals is APX-hard. Our result is obtained by a reduction from the Integer2Commodity problem that is known to be APX-hard [15, Corollary 4.1]. We also show that the element-disjoint Steiner Tree Packing problem with 3 terminals is APX-hard, by using a simple reduction from the edge-disjoint version.

Theorem 3.4 *The edge-disjoint* Steiner Tree Packing *problem with* 3 *terminals is APX-hard.*

The Integer2Commodity problem is as follows: We are given an undirected graph $G = (V, E)$ and distinct nodes $x_1, y_1, x_2, y_2 \in V$; the goal is to find a maximum-size collection of edge-disjoint paths, each joining either x_1 to y_1 or x_2 to y_2.

Theorem 3.5 ([15, Corollary 4.1]) *The* Integer2Commodity *problem is APX-hard.*

Reduction: In the hardness construction in the proof of the above theorem (see [15, Section 4.1.1]), the nodes x_1 and y_1 both have degree d_1, and the node y_2 has degree d_2; so there are at most d_i edge-disjoint paths between x_i and y_i for each $i \in \{1, 2\}$. In the "yes" instances of the problem the objective value is $d_1 + d_2$, whereas in the "no" instance the objective value is at most $(d_1 + d_2)(1 - \epsilon)$, for some $\epsilon > 0$.

(Refer to the appendix for more details.) We denote this instance of the INTEGER2COMMODITY problem by $\mathcal{I} = (G; x_1, y_1, d_1; x_2, y_2, d_2)$.

We use the same construction as the construction in Theorem 3.1. Let H be the undirected graph with three terminal nodes t, t_1, t_2 that is obtained from G (see Figure 3 for an illustration).

Proof of Theorem 3.4 :

Completeness: Assume that the instance $\mathcal{I}$ has objective value of $d_1 + d_2$. Hence, there are d_i edge-disjoint paths from x_i to y_i for each $i \in \{1, 2\}$. Observe that by adding edges $\{s_1, t\}$, $\{s_1, x_1\}$, $\{s_1, t_2\}$, $\{y_1, t_1\}$ to any x_1-y_1 path in G we get a Steiner tree. Hence, using d_1 edge-disjoint x_1-y_1 paths, we get d_1 edge-disjoint Steiner trees. Similarly, we can get d_2 edge-disjoint Steiner trees using d_2 edge-disjoint x_2-y_2 paths. Moreover, any two of these $d_1 + d_2$ Steiner trees are edge-disjoint. Thus, H has $d_1 + d_2$ edge-disjoint Steiner trees.

Soundness: Suppose H has a set $\mathcal{T}$ of at least $(1-\epsilon)(d_1+d_2)$ edge-disjoint Steiner trees in H. First, we claim that for each terminal node there are at most $\epsilon(d_1 + d_2)$ Steiner trees from $\mathcal{T}$ where that terminal node is not a leaf. Consider a terminal node $r \in \{t, t_1, t_2\}$. Let n_1 denote the number of Steiner trees from $\mathcal{T}$ where r is a leaf node, and let n_2 be the number of remaining Steiner trees. The degree of each terminal node is $d_1 + d_2$, so $n_1 + 2n_2 \leq d_1 + d_2$; note that each terminal node has either degree 1 or degree 2 in any Steiner tree. Also we know that $n_1 + n_2 = |\mathcal{T}| \geq (1-\epsilon)(d_1 + d_2)$. These two inequalities imply that $n_2 \leq \epsilon(d_1 + d_2)$. Applying this argument for all three terminal nodes, we see that the following holds: There is a set of edge-disjoint Steiner trees $\mathcal{T}' \subseteq \mathcal{T}$ of size at least $(1 - 4\epsilon)(d_1 + d_2)$ such that in each Steiner tree in $\mathcal{T}'$ each terminal node has degree one.

Note that each Steiner tree $T \in \mathcal{T}'$ has a non-terminal node of degree 3, since all terminal nodes in T have degree 1. The node of degree 3 in each Steiner tree is either a node from G (a node inside the dashed box in Figure 3) or a node from $\{s_1, s_2\}$. Let $\mathcal{T}'' \subseteq \mathcal{T}'$ be the set of Steiner trees where the node of degree 3 is either s_1 or s_2. We claim that there are at least $(1 - 12\epsilon)(d_1 + d_2)$ Steiner trees in $\mathcal{T}''$. To see this, note that each Steiner tree from $\mathcal{T}' \setminus \mathcal{T}''$ uses at least 3 edges crossing the dashed box in Figure 3, and each Steiner tree from $\mathcal{T}''$ uses exactly two edges crossing the dashed box. Let $q_2 = |\mathcal{T}''|$, and let $q_3 = |\mathcal{T}' \setminus \mathcal{T}''|$. There are $2(d_1 + d_2)$ edges crossing the dashed box, so we have $2q_2 + 3q_3 \leq 2(d_1 + d_2)$. We also have $q_2 + q_3 = |\mathcal{T}'| \geq (1 - 4\epsilon)(d_1 + d_2)$. Using these two inequalities, we get $|\mathcal{T}' \setminus \mathcal{T}''| = q_3 \leq 8\epsilon(d_1 + d_2)$, and this proves our claim. Note that a Steiner tree in $\mathcal{T}''$ may contain both s_1 and s_2. Using similar arguments as above, we can show that there is a set of edge-disjoint Steiner trees $\mathcal{T}''' \subseteq \mathcal{T}''$ of size at least $(1 - 30\epsilon)(d_1 + d_2)$ such that each Steiner tree $T \in \mathcal{T}'''$ satisfies the following properties: (1) each terminal node in T has degree 1, (2) the degree 3 node in T is either s_1 or s_2, and (3) T does not contain both s_1, s_2. To see this, note that each tree in $\mathcal{T}'''$ uses exactly 3 of the edges incident to both s_1, s_2 and each tree in $\mathcal{T}'' \setminus \mathcal{T}'''$ uses at least 5 of the edges incident to both s_1, s_2, hence, we get $|\mathcal{T}'' \setminus \mathcal{T}'''| \leq 18\epsilon(d_1 + d_2)$.

Now consider a Steiner tree $T \in \mathcal{T}'''$ containing s_1. Tree T uses the edge $\{s_1, x_1\}$ to enter G and it has no edges incident to s_2, so it must contain the edge $\{y_1, t_1\}$. Hence, T contains an x_1-y_1 path inside G. Similarly, if T contains s_2, then it contains an x_2-y_2 path inside G. Therefore, the Steiner trees in $\mathcal{T}'''$ give us a solution to the instance $\mathcal{I}$ with objective value at least $|\mathcal{T}'''| \geq (1 - 30\epsilon)(d_1 + d_2)$. This completes the proof of soundness. □

Corollary 3.6 *The element-disjoint* STEINER TREE PACKING *problem with* 3 *terminals is APX-hard.*

Proof: The proof is by a reduction from the edge-disjoint STEINER TREE PACKING problem. Let G be an instance of the edge-disjoint STEINER TREE PACKING problem with 3 terminals. Let G' be a graph obtained from G by sequentially replacing each Steiner node v of degree d by a clique of size d and connecting each neighbor of v to a distinct node in the clique. It can be checked that any set of element-disjoint Steiner trees in G' corresponds to a set of edge-disjoint Steiner trees in G, and there is a bijection

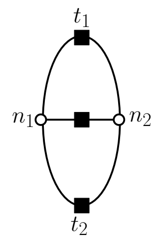


Figure 4: Basic Gadget

between the two sets, i.e., each tree in the first set corresponds to a tree in the second set. Therefore, the maximum number of element-disjoint Steiner trees in G' is equal to the maximum number of edge-disjoint Steiner trees in G. Hence, by Theorem 3.4, the element-disjoint STEINER TREE PACKING problem with 3 terminals is APX-hard. □

3.3 NP-hardness of packing 2 element-disjoint Steiner trees in planar graphs

In this subsection, we prove the following Theorem.

Theorem 3.7 *The problem of finding two element-disjoint Steiner trees in planar graphs is NP-hard.*

Our proof is based on two previous results, namely, Kaski's proof [19] that the problem of finding two edge-disjoint Steiner trees in general graphs is NP-hard, and Plesník's proof [29] that the Hamiltonian cycle problem in planar digraphs with degree bound two is NP-hard. We give a reduction from the NAE-3SAT problem, see the NOT-ALL-EQUAL 3SAT problem in [9]. An instance of this problem consists of a set of boolean variables $\{x_1, \dots, x_n\}$ and a collection of clauses $\{Q_1, \dots, Q_m\}$, where each clause consists of three literals; the question is whether there exists a truth assignment to the variables such that each clause has at least one true literal and at least one false literal. This problem is NP-complete.

A *Basic Gadget* or BG is a complete bipartite graph with 3-terminals and 2-Steiner nodes (see Figure 4). In any planar embedding of this graph, the outer face consists of two terminals and the two Steiner nodes. If H is a BG we use $H(t_1)$ and $H(t_2)$ to denote its terminals on the outer face and $H(n_1)$ and $H(n_2)$ to denote its Steiner nodes (also on the outer face). Note that any solution to the (planar) 2-element disjoint trees problem on H, contains $H(n_1)$ and $H(n_2)$ in different trees.

Reduction: Let $\mathcal{I} = Q_1 \wedge Q_2 \wedge \cdots Q_m$, be an instance of NAE-3SAT where clause $Q_j = P_{j_1} \vee P_{j_2} \vee P_{j_3}$, with literals $P_{j_k} \in \{x_1, \bar{x}_1, x_2, \bar{x}_2, \cdots, x_n, \bar{x}_n\}$.

Given $\mathcal{I}$, we describe how to create an instance $G = (U \cup R, E)$ of the planar 2-element disjoint Steiner trees problem, such that $\mathcal{I}$ has two complementary satisfying assignments if and only if G has two element-disjoint Steiner trees. We first describe the construction of a partial planar graph G_p along with its embedding; this will aid us in constructing G. We define G_p and its embedding as follows. (See Figure 5(a) for an example).

1. We add a sequence of "clause" BG's $C_1, C_2, \cdots, C_{3m}$ such that adjacent BG's share their outer terminals: i.e. for $i \in \{1, \cdots, 3m-1\}$, $C_i(t_2) = C_{i+1}(t_1)$ and all nodes $C_i(n_2)$ are on the same side in the embedding (see Figure 5(a)).

2. For each clause Q_i, we add a terminal q_i and connect it to Steiner nodes $C_{3i-2}(n_1)$, $C_{3i-1}(n_1)$ and $C_{3i}(n_1)$.

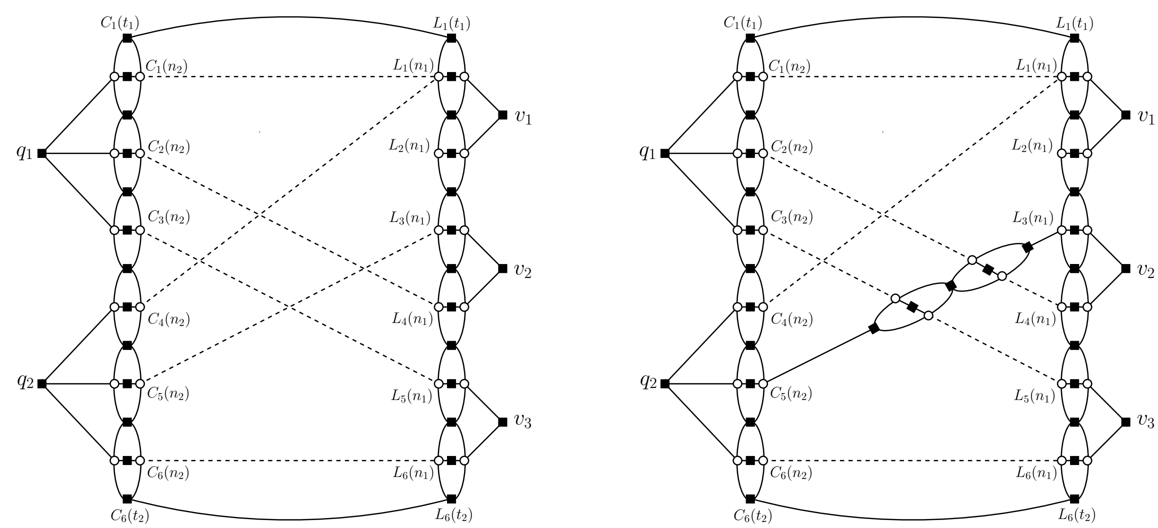
3. We add a sequence of "literal" BG's $L_1, L_2, \cdots, L_{2n}$ such that adjacent BG's share their outer terminals: i.e for $i \in \{1, \cdots, 2n-1\}$, $L_i(t_2) = L_{i+1}(t_1)$ and all nodes $L_i(n_1)$ are on the same side in the embedding.

4. For each pair of BG's L_{2i-1} and L_{2i} (where $1 \in \{1, \cdots, n\}$), we add a new terminal v_i and connect it to $L_{2i-1}(n_2)$ and $L_{2i}(n_2)$.

5. We add the edges $\{C_1(t_1), L_1(t_1)\}$ and $\{C_{3m}(t_2), L_{2n}(t_2)\}$.

6. Finally, we add certain constraints between the Steiner nodes of G_p called *switching lines*. A switching line (s_1, s_2) between Steiner nodes s_1 and s_2 ensures the two nodes are in different Steiner trees in any solution to G. Later in the section, we give a procedure to replace the switching lines with certain gadgets that "implement" them. For each clause $Q_i = P_{i_1} \vee P_{i_2} \vee P_{i_3}$, let L_{j_1}, L_{j_2} and L_{j_3} be the literal BG's corresponding to P_{i_1}, P_{i_2} and P_{i_3}. (e.g., If $P_{i_1} = x_k$ then $j_1 = 2k-1$ and if $P_{i_1} = \bar{x_k}$, then $j_1 = 2k$). We add the following switching lines to G_p: $(C_{3i-2}(n_2), L_{j_1}(n_1)), (C_{3i-1}(n_2), L_{j_2}(n_1)), (C_{3i}(n_2), L_{j_3}(n_1))$.

7. **Embedding of** G_p: Let H be the graph G_p without the switching lines. We embed H in the plane such that the clause BG's are aligned vertically to the left, the literal BG's are aligned vertically to the right and the cycle $B_p = C_1(t_1), C_1(n_2), C_2(t_1), C_2(n_2), \cdots, C_{3m}(t_1), C_{3m}(n_2), C_{3m}(t_2), L_{2n}(t_2), L_{2n}(n_1), L_{2n}(t_1), L_{2n-1}(n_1), L_{2n-1}(t_1), \cdots, L_1(n_1), L_1(t_1)$ forms an (internal) face of H (see Figure 5(a)). We define a *boundary* to be a cycle of the embedded graph whose interior contains no nodes or edges but may contain switching lines. Thus, we call B_p the boundary of G_p. Now, we represent each switching line of G_p with a straight (dashed) line joining its end nodes. Note that these line segments would all be present inside (the embedding of) B_p. Also the line segments may cross each other. But without loss of generality, we assume that no three switching lines cross at the same point.



(a) Graph G_p for the instance $\mathcal{I} = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

(b) The graph obtained by applying the Uncross operation on the switching line $(C_5(n_2), L_3(n_1))$

Figure 5: Planar construction

We now describe the procedure for obtaining G from G_p. Given a boundary B and a switching line e in B, the following operation replaces e with a subgraph $\hat{S}_e$ and adjusts the interior of B, splitting it into two boundaries.

Uncross (B, e):

1. If e does not cross any other switching line, then define $\hat{S}_e$ to be a path of length two connecting the end nodes of e and having a (new) terminal in the middle. Delete e and embed $\hat{S}_e$ along the straight line corresponding to e.

2. Otherwise let e cross switching lines $e_1, e_2, \cdots, e_k$. In this case, define $\hat{S}_e$ as a sequence of `BG` gadgets $R_1, R_2, \cdots, R_k$, such that adjacent `BG`'s share an outer terminal (i.e., $R_i(t_2) = R_{i+1}(t_1)$ for $i \in \{1, \cdots, k-1\}$). Now connect the terminal nodes $R_1(t_1)$ and $R_k(t_2)$ to the end nodes of e.

   Delete e and embed $\hat{S}_e$ such that all terminals in $\hat{S}_e$ lie along the straight line corresponding to e and for each R_i, $i \in \{1, \cdots, k\}$, the Steiner nodes $R_i(n_1)$ and $R_i(n_2)$ lie along the straight line corresponding to e_i. Also, for each i $(\in \{1, \cdots, k\})$, replace e_i with two switching lines, from $R_i(n_1)$ and $R_i(n_2)$ to the end nodes of e_i. These switching lines are embedded as (disjoint) line segments that are contained in the line segment corresponding to e_i. Figure 5(b) illustrates this with an example.

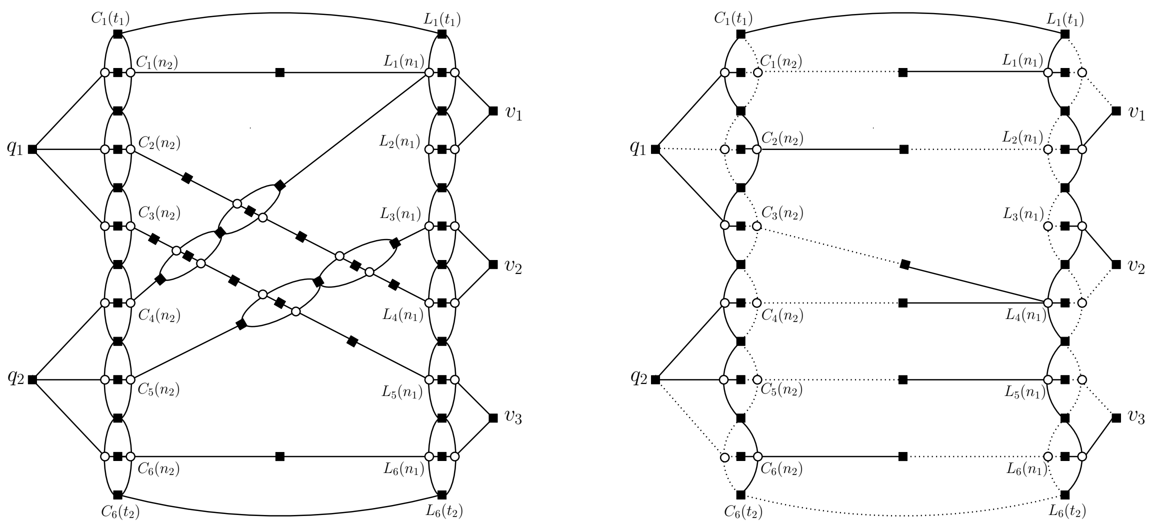
Note that $\hat{S}_e$ divides the boundary B into two boundaries B^1 and B^2 such that B^1 and B^2 share the end nodes of e and the outer terminals of $\hat{S}_e$. To construct G from G_p, we start by applying the above operation on the boundary B_p and an arbitrary switching line e_p in B_p. This splits B_p into boundaries B_p^1 and B_p^2 such that any new or remaining switching line is present in either B_p^1 or B_p^2. We use the above operation recursively to eliminate all the switching lines in B_p^1 and B_p^2. (Figure 6(a) shows the construction for a sample instance). Let $SLT(G_p)$ be the recursion tree obtained by this procedure. $SLT(G_p)$ is a binary tree in which each node is represented by a pair (B, e), where B is a boundary and e is a switching line in B. A node (B', e') is a child of (B, e) if boundary B' is one of the two boundaries obtained by applying the uncross operation at (B, e). If (B, e) is a leaf node then B doesn't contain any switching lines inside it, and we define e to be empty. We assign an integer number called the level to each pair (B, e) in the above construction. The level of the pair (B_p, e_p) is defined to be 0. If (B', e') is a child of a pair (B, e) at level i, then we define the level of (B', e') to be $i + 1$. Let h denote the maximum level over all pairs; i.e., h is the height of the recursion tree $SLT(G_p)$. In the construction of G, we assign a level to each of the following objects: terminal nodes, Steiner nodes, boundaries, and switching lines. The objects in G_p (before applying any uncross operation) are defined to be at level 0. When we apply the uncross operation to a pair at level i in $SLT(G_p)$, we define the level of the new objects (i.e., terminal nodes, Steiner nodes, switching lines, and the two new boundaries) to be $i + 1$.

Lemma 3.8 *The instance $\mathcal{I}$ of* NAE-3SAT *is satisfiable if and only if G has two element-disjoint Steiner trees.*

Proof: We first prove the lemma for the simple case when no two switching lines in G_p cross each other, that is, each switching line in G is realized by a path of length two with a unique terminal in the middle. Subsequently, we extend the proof to the general case.

Simple Case (switching lines of G_p do not cross each other):

Consider an assignment that satisfies $\mathcal{I}$. We show that elements of G can be colored with red and blue colors such that each color class corresponds to a Steiner tree, and the two Steiner trees are element-disjoint. We first color the subgraph G_p of G as follows.



(a) The final graph G obtained by recursive applications of the uncross operation.

(b) Two element-disjoint Steiner trees corresponding to $\mathcal{I} = (x \vee \bar{x} \vee \bar{y}) \wedge (\bar{y} \vee z \vee \bar{z})$ and the assignment $(x, y, z) = (T, F, T)$

Figure 6: The final construction

1. For each clause Q_i and each literal P_{i_k} in Q_i, if P_{i_k} is true, then we assign red color to $C_{3(i-1)+k}(n_1)$, forcing $C_{3(i-1)+k}(n_2)$ to take blue color. If P_{i_k} is false, then we assign blue color to $C_{3(i-1)+k}(n_1)$, forcing $C_{3(i-1)+k}(n_2)$ to take red color.

2. For each variable x_i, if x_i is set to true then we assign red to $L_{2i-1}(n_1)$ and blue to $L_{2i}(n_1)$, forcing $L_{2i-1}(n_2)$ and $L_{2i}(n_2)$ to take blue and red colors respectively. If x_i is set to false then we assign blue to $L_{2i-1}(n_1)$ and red to $L_{2i}(n_1)$, forcing $L_{2i-1}(n_2)$ and $L_{2i}(n_2)$ to take red and blue colors respectively.

3. We assign red color to the edge $\{C_1(t_1), L_1(t_1)\}$ and blue color to the edge $\{C_{3m}(t_2), L_{2n}(t_2)\}$.

It is easy to verify that the above coloring respects the switching lines of G_p (i.e., for any switching line e in G_p, the two end nodes of e have different colors). Note that the subgraphs induced by each color class together with all terminal nodes is connected in the graph G_p without the switching lines; see figure 6(b) for an illustration. In G, each remaining terminal (i.e., terminal not in G_p) is adjacent to the ends of a switching line and hence connected to both color classes. Thus, each color class is connected and contains all the terminals. This shows that if $\mathcal{I}$ is satisfiable then G has two element-disjoint Steiner trees.

For the other direction, assume that G has two element-disjoint Steiner trees. We color all the Steiner nodes of G that appear in one of the trees with red and the Steiner nodes of the other tree are colored with blue. Consider the subgraph G_p of G. For any switching line e of G_p, G contains a terminal (of degree 2) adjacent to the end nodes of e. Hence, the end nodes of each switching line should have different colors. Now, for any variable x_i ($i \in 1, \cdots, n$), the nodes $L_{2i-1}(n_1)$ and $L_{2i}(n_1)$ have different colors (because of terminal v_i). If these two nodes are colored red and blue respectively, then we assign true to x_i. Otherwise $L_{2i-1}(n_1)$ and $L_{2i}(n_1)$ are colored blue and red (respectively) and we assign false to x_i. We claim that this is a satisfying assignment for $\mathcal{I}$. For the sake of contradiction, assume that there is a clause Q_j, $j \in 1, \cdots, m$, such that every literal P_{j_k}, $\forall k \in \{1, 2, 3\}$, is false. Now, consider any $k \in \{1, 2, 3\}$. If P_{j_k}

appears uncomplemented (i.e., $P_{j_k} = x_l$ for some variable x_l), then the switching line between $C_{3(j-1)+k}(n_2)$ and $L_{2l-1}(n_1)$ implies that $C_{3(j-1)+k}(n_2)$ is forced to be red and hence $C_{3(j-1)+k}(n_1)$ is forced to be blue. If P_{j_k} is of the form $P_{j_k} = \bar{x}_l$ for some variable x_l, then the switching line between $C_{3(j-1)+k}(n_2)$ and $L_{2i}(n_1)$ implies that $C_{3(j-1)+k}(n_2)$ is forced to be red and hence $C_{3(j-1)+k}(n_1)$ is forced to be blue. Thus, for each $k \in 1, 2, 3$, $C_{3(j-1)+k}(n_1)$ is colored blue. But this disconnects the terminal q_j in the red tree. Therefore, all three literals cannot be false in any clause and hence the assignment satisfies every clause. By a symmetric argument, we can show that the complement of the assignment also satisfies every clause. Therefore, $\mathcal{I}$ is a satisfiable instance of NAE-3SAT.

General Case (switching lines of G_p may cross each other):

We now prove the lemma for the general case when switching lines of G_p may cross each other.

Consider an assignment that satisfies the NAE-3SAT instance $\mathcal{I}$. We show that the Steiner nodes in G can be colored with blue and red colors such that each color class together with all the terminal nodes induces a connected subgraph. We first color all Steiner nodes in the graph G_p (i.e., all Steiner nodes that are at level 0 in the construction) in the same way as explained in the simple case, and next we "extend" this coloring to all the remaining Steiner nodes (at other levels) in G. Recall that in the coloring of G_p the end nodes of each switching line in B_p have different colors. This is an important property in our coloring, and in our extended coloring, at any level of the construction, each switching line has this property. Let us denote the set of all Steiner nodes of blue (or red) color at levels $\leq i$ by U_i^b (or U_i^r). Also let T_i denote the set of all terminal nodes at levels $\leq i$.

We claim that the coloring of G_p can be extended in such a way that the subgraphs induced by $T_i \cup U_i^b$ and $T_i \cup U_i^r$ form two element-disjoint connected subgraphs, and moreover, the end nodes of each switching line at level $\leq i$ have different colors. We prove this by induction on i.

Base case: The coloring of Steiner nodes of G_p explained in the easy case proves the claim for $i = 0$.

Induction step: Now, assume that the claim is correct for all $j \leq i$. Let (B, e) be a pair of face boundary and its corresponding switching line at level $i + 1$ where the uncross operation is applied, and suppose that $e_1, \ldots, e_k$ are the switching lines crossing e (in this order when we move from one end node of e to the other end node). In the uncross operation, we realized the switching line e by a sequence of k new `BG` gadgets, and replaced each e_i by two switching lines (e_i^1 and e_i^2). This uncrossing operation split the boundary B into boundaries B^1 and B^2. Note that one end node of each new switching line is connected to a Steiner node on B and the other end node is a non-colored Steiner node on a copy of `BG` gadget (along e). Now color the non-colored end node of each new switching line by the color "opposite" to the color of the end node that is on B; this shows the first property in the induction hypothesis. In this coloring, the two Steiner nodes in each new copy of the `BG` gadget get different colors. It is easy to check that in this coloring all new terminal nodes (at level $i + 1$) are connected to the blue end node of e via blue Steiner nodes, and similarly all new terminal nodes are connected via red Steiner nodes to the red end node of e; all these paths are going through the k new copies of the `BG` gadget along e. Note that the end nodes of e belong to different connected subgraphs induced by $T_i \cup U_i^b$ and $T_i \cup U_i^r$ (by the induction hypothesis), since each end node of e is a Steiner node of level $\leq i$. Applying the above procedure to all pairs (B, e) at level $i + 1$, we can color all Steiner nodes at level $i + 1$; this shows that the subgraphs induced by $T_{i+1} \cup U_{i+1}^b$ and $T_{i+1} \cup U_{i+1}^r$ are connected. This completes the induction step.

Note that the above claim for $i = h$ proves that G has two element-disjoint Steiner trees, since $T_h = R$. Thus, we showed that if $\mathcal{I}$ is satisfiable, then G has two element-disjoint Steiner trees.

For the other direction, assume that G has two element-disjoint Steiner trees. We color all the Steiner nodes of G that appear in one of the trees with red and the Steiner nodes of the other tree with blue.

We prove that the end nodes of each switching line in G_p have different colors. This completes the proof by the argument given in the simple case. We prove that this property is satisfied at each level of the construction; i.e., end nodes of each switching line at any level have different colors.

Given a boundary B, a path P is called a red B-path if all the Steiner nodes in P are red, P meets the boundary of B only at its end nodes, P has at least one other node besides the two end nodes, and P lies in the interior of B. A blue B-path is defined similarly. Now, we claim the following. In each boundary B at level $\geq i$ the end nodes of each switching line, e in B, have different colors, and in B there is no red or blue B-path. We prove this claim by induction on i.

Base case: The claim is trivial for $i = h$, since each boundary B at level h (leaf of the construction tree) has no nodes in its interior; note that B has no switching line.

Induction step: Assume that the claim is true for all $j \geq i$. Now, consider a boundary B at level $i - 1$, and suppose the uncross operation is applied on a switching line e at this step (in the construction of G); i.e., (B, e) is a uncross operation at this level. Let B^1 and B^2 be the boundaries obtained after this uncross operation. First, we show that the end nodes of e have different colors. For the sake of the contradiction, assume that both end nodes of e are colored blue; the other case where both end nodes are red is similar to this case. Consider a terminal node t that lies on boundaries of both B^1 and B^2 in the subgraph $\hat{S}_e$ that realizes the switching line e; recall that if e has no switching line crossing it, then e is replaced by a terminal, otherwise, e is replaced by a sequence of `BG` gadgets. Let t' be a terminal node in the outer face of G. The terminal t is connected to t' in the red tree. Both end nodes of e are blue, so such a red path from t to t' needs to go through the interior of B^1 or B^2. But, this is a contradiction, since by induction hypothesis B^i (for $i = 1, 2$) has no red B^i-path. Now consider a switching line f (other than e) in B. If f lies in B^1 or B^2, then obviously its end nodes have different colors by the induction hypothesis. Otherwise, f crosses e, so f is replaced by two new switching lines f^1 and f^2, where one of them, say f^1, lies in B^1 and the other one lies in B^2. Note that each one of them is connected from one end node to a boundary of B and from the other end node to a Steiner node of the same copy of a `BG` gadget in $\hat{S}_e$. By the induction hypothesis end nodes of f^1 and f^2 have different colors, and also the end nodes of f^1 and f^2 in the corresponding `BG` gadget have different colors. Hence, the end nodes of f are forced to be different. This proves the first property of the induction hypothesis.

Now we show that there is no monochromatic B-path. Let P be a B-path, and note that by definition it is going through the interior of B. The path P either contains a B^1-path or a B^2 path or it completely goes through the subgraph $\hat{S}_e$ that realizes the switching line e. In the first two cases P cannot be monochromatic by the induction hypothesis. In the last case, note that P starts from one end node of e and ends at the other end node of e. Hence, it cannot be monochromatic either, since both end nodes of e have different colors. This completes the induction step and proves the claim.

The above claim for $i = 0$ shows that the end nodes of each switching line in B_p have different colors. Therefore, by the argument in the easy case the NAE-3SAT instance $\mathcal{I}$ is satisfiable. □

3.4 NP-hardness of packing 2 edge-disjoint Steiner trees in planar graphs

In this subsection, we extend Theorem 3.7 to the setting of edge-disjoint Steiner trees, and we sketch a proof of the following theorem.

Theorem 3.9 *The problem of finding two edge-disjoint Steiner trees in planar graphs is NP-hard.*

We use the notation from the construction used for Theorem 3.7. Our plan is to start with the planar graph G constructed in the proof of Theorem 3.7, and then to apply some modifications to get the result on edge-disjoint Steiner trees; informally speaking, these modifications are successful because G has a

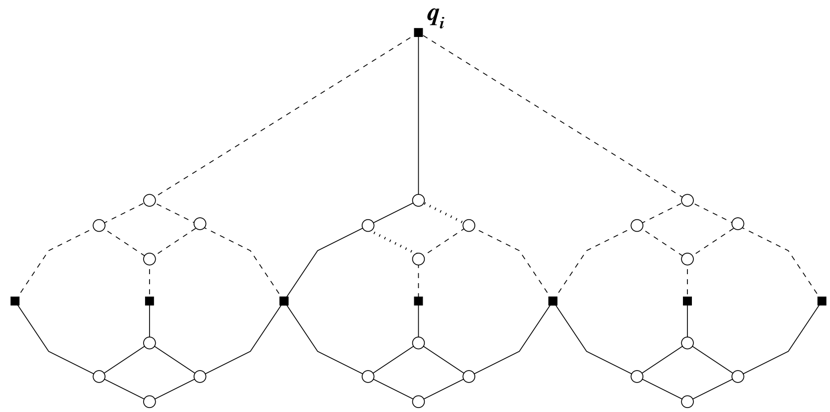
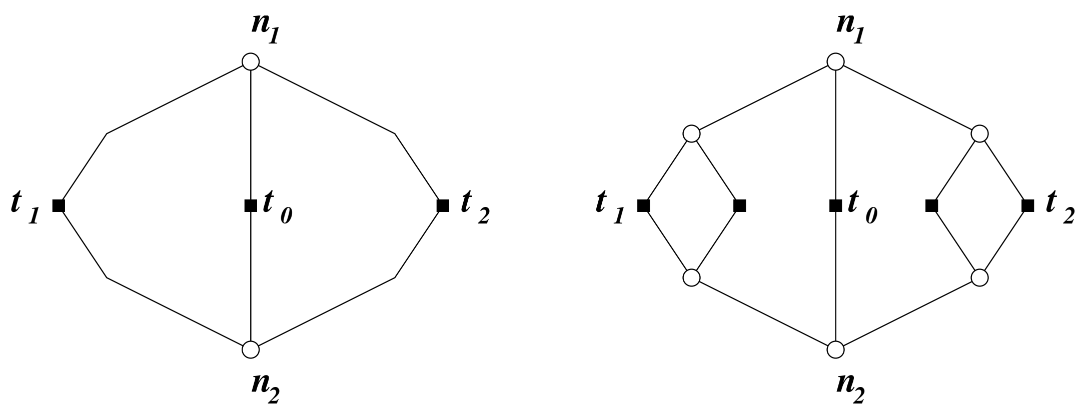


Figure 7: Two edge-disjoint Steiner trees do not imply satisfiability of the NAE-3SAT formula. The sequence of three `BG`s adjacent to a "clause terminal" q_i "allows" two edge-disjoint Steiner trees: one contained in the set of solid edges, the other one contained in the set of dashed edges; the dotted edges are in neither tree; but the clause is *not* satisfied because all three literals get the same truth value.

special structure. (The modifications were suggested by Guyslain Naves.) Suppose that we replace each Steiner node of G of degree ≥ 4 by a gadget such that the resulting graph has each Steiner node of degree ≤ 3, while preserving the key property that there exist two element-disjoint Steiner trees iff the instance of NAE-3SAT is satisfiable. Then, we would have the desired result, since an upper bound of three on the degree of every Steiner node implies that a pair of Steiner trees is disjoint on the edges iff the pair is disjoint on the elements. Thus, we replace each Steiner node v of G of degree $\deg(v) \geq 4$ by a cycle on $\deg(v)$ new Steiner nodes, and we replace the edges of G incident to v by edges incident to distinct nodes of the cycle. (See Figure 10 for an illustration for a Steiner node of degree 4.) Unfortunately, this does not suffice to preserve the key property. (We remark that the terminals q_i for the clauses are adjacent to three basic gadgets, and this property may be exploited to construct edge-disjoint Steiner trees that do not correspond to element-disjoint Steiner trees.) See Figure 7 for an illustration. We also need to replace the basic gadget `BG`. We may view `BG` as consisting of two "sides", where each "side" is a $K_{1,3}$ (a star with 3 leaves), and the three terminals are common to the two "sides". The replacing gadget that we denote by `BGE` has five terminals, and these are common to the two "sides"; each "side" consists of a tree with three Steiner nodes (internal nodes of the tree) and the five terminals (leaves of the tree); let the terminals be denoted $t_0, t_1, t_2, t'_1, t'_2$, where t_1, t_2 are "external terminals" (used for attaching a basic gadget to its neighboring basic gadgets, for example, in the sequence of "clause" gadgets $C_1, \ldots, C_{3m}$) and the others are "internal terminals". The root of each tree corresponds to one of the Steiner nodes n_1 or n_2 of `BG`, and it has three children: two Steiner nodes, and the terminal t_0; moreover, each of the non-root Steiner nodes has two of the terminals as children. See Figure 8 for an illustration.

Let G' denote the graph obtained from G (the graph constructed for Theorem 3.7) by replacing all `BG` gadgets by `BGE` gadgets, and then replacing all Steiner nodes of degree ≥ 4. Similarly to the proof of Theorem 3.7, the easy part of the proof is to show that G' has two edge-disjoint Steiner trees if the instance of NAE-3SAT is satisfiable. Let us focus on the other part of the proof, and let us assume that G' has two edge-disjoint Steiner trees.

In our construction, in every copy of `BGE`, each internal terminal has degree two, hence, their two incident edges and two neighbors (which are Steiner nodes of degree 3) are placed in two different Steiner trees. Moreover, we claim that all the nodes of the same "side" of the gadget may be placed in the same Steiner tree. (Observe that the middle gadget in Figure 7 violates this claim.) It can be seen that



(a) BG (b) BGE

Figure 8: The Basic Gadgets for element-disjoint Steiner trees, BG, and for edge-disjoint Steiner trees, BGE

Theorem 3.9 follows from this claim. Informally speaking, the claim implies that the functioning of each BGE gadget is the same as the functioning of a BG gadget; hence, the proof of Lemma 3.8 applies to G' and guarantees that the instance of NAE-3SAT is satisfiable (since G' has two element-disjoint Steiner trees).

This claim can be verified if the switching lines do not cross each other, as in the proof of the simple case of Lemma 3.8. First, observe that every terminal of degree 2 is a leaf of each Steiner tree; therefore, we may ignore all terminals of degree 2. Now, consider the subgraph formed by the sequence of "literal" BGEs $L_1, \ldots, L_{2n}$ and the two edges $\{C_1(t_1), L_1(t_1)\}$ and $\{C_{3m}(t_2), L_{2n}(t_2)\}$. Clearly, each of the two Steiner trees contains exactly one of the latter two edges. Hence, each Steiner tree must induce a connected subgraph of the subgraph formed by the sequence of "literal" BGEs $L_1, \ldots, L_{2n}$. It follows that our claim holds for these "literal" BGEs: all the nodes of the same "side" of the gadget may be placed in the same Steiner tree. The claim also holds for the "clause" BGEs $C_1, \ldots, C_{3m}$. This can be verified by detailed checking; we omit the details, but mention the importance of the three internal terminals of degree 2 in each BGE gadget.

Finally, observe that a switching line e with crossings is replaced by a subgraph, call it $G'(e)$, that consists of a sequence of BGEs $R_1, \ldots, R_k$ together with edges between these BGEs and some other terminals of degree 2; also, there are two other edges between $G'(e)$ and the rest of G', namely, the edge between R_1 and one end node of the switching line, and the edge between R_k and the other end node of the switching line. Thus, we may view $G'(e)$ as a terminal of degree 2, and $G'(e)$ is essentially a leaf of each of the two Steiner trees. In other words, deleting $G'(e)$ from G' should not disconnect either of the two Steiner trees, hence, we cannot "route" one of the Steiner trees "through" $G'(e)$. Thus, the claim holds in general, even when the switching lines cross each other.

4 Integrality ratio for packing Steiner trees in planar graphs

In this section, we show that the "standard" linear programming relaxation of the element-disjoint Steiner Tree Packing problem has integrality ratio approaching 2, even on planar graphs. This result extends to give the same lower bound on the integrality ratio for the edge-disjoint Steiner Tree Packing problem.

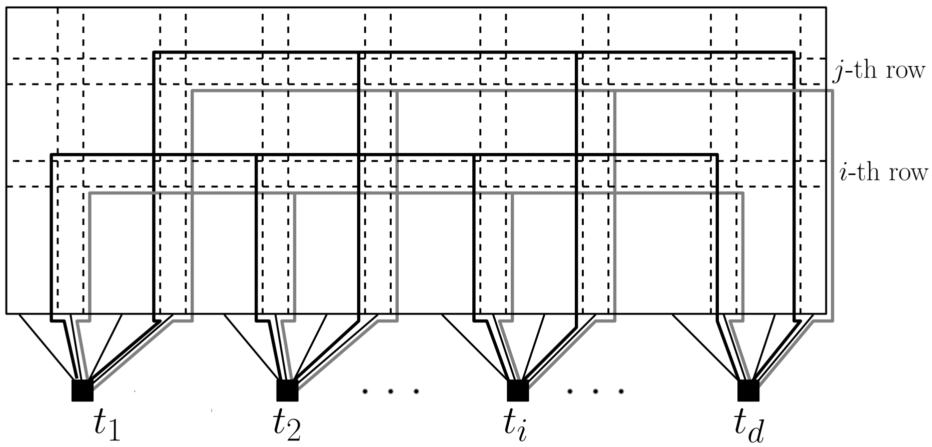


Figure 9: Integrality ratio example for the element-disjoint problem

4.1 Integrality ratio example for packing element-disjoint Steiner trees

Consider the following linear programming relaxation of the element-disjoint Steiner Tree Packing problem. For notational convenience, we assume there are no edges between terminals, by subdividing edges if needed.

$$\text{(LP-element)} \quad z_{LP}(G) = \max \sum_{T \in \mathcal{T}} x_T$$

$$\text{subject to}$$

$$\sum_{T \in \mathcal{T}: v \in T} x_T \leq 1 \qquad \forall v \in V \setminus R$$

$$x_T \geq 0 \qquad \forall T \in \mathcal{T}$$

Theorem 4.1 *The LP relaxation of the element-disjoint* Steiner Tree Packing *problem has an integrality ratio* $\geq 2 - \frac{2}{|R|} - \epsilon$ *even on planar graphs, where the additive term* ϵ *is a function of* k *and* $|R|$ *and for fixed* $|R|$, $\epsilon \to 0$ *as* $k \to \infty$ *(here,* k *denotes the element-connectivity of the terminals).*

Construction: Start from a $2k \times 2kd$ grid (with $2k+1$ horizontal lines and $2kd+1$ vertical lines); then subdivide the alternate edges of the last row of the grid; thus, the number of subdividing nodes is kd; moreover, within the last row, there is a path of length three between consecutive subdividing nodes. Next, add d terminal nodes $R = \{t_1, \ldots, t_d\}$ to the outer face of the grid. Finally, connect each terminal node t_i to k consecutive subdividing nodes, such that each subdividing node is connected to one terminal node. Let G be the obtained graph. The construction is illustrated in Figure 9; the figure does *not* show all of the graph G; in particular, many vertical lines and horizontal lines of the grid are omitted; the purpose of the figure is to illustrate half-integral Steiner trees and the way they cross each other (see the proof of Claim 4.5).

First, we prove that G has at most $\frac{kd}{2(d-1)} + d - 2$ element-disjoint Steiner trees. Next, we show that `LP-element` has optimal value of $z_{LP} = k$. Proving these two claims completes the proof of Theorem 4.1. The following result holds for *any* planar graph that satisfies the conditions given in the result; the graph

G satisfies these conditions, so the result applies to G. The result will be also used in the analysis of the integrality ratio example for the edge-disjoint Steiner Tree Packing problem.

Lemma 4.2 *Let $\hat{G} = (\hat{V}, \hat{E})$ be any planar graph, and let R be a subset of $\hat{V}$; call R the set of terminal nodes. Let d denote $|R|$, where $d \geq 3$. Suppose that there is no edge between terminal nodes, the degree of each terminal node is k, and $\hat{G}$ has a planar embedding such that all terminal nodes are on the outer face. Then $\hat{G}$ has at most $\frac{kd}{2(d-1)} + d - 2$ element-disjoint Steiner trees.*

Proof: Let $\mathcal{S}$ be a maximum-size set of element-disjoint R-Steiner trees in $\hat{G}$. We may assume that none of the Steiner trees in $\mathcal{S}$ has a Steiner node of degree one. By a *terminal edge* we mean an edge that is incident to a terminal. Let $\mathcal{S}_1$ be the set of Steiner trees from $\mathcal{S}$ with less than $2(d-1)$ terminal edges, and let $\mathcal{S}_2$ be the remaining subset of $\mathcal{S}$ (i.e., $\mathcal{S}_2 = \mathcal{S} \setminus \mathcal{S}_1$). Note that $|\mathcal{S}_2| \leq \frac{kd}{2(d-1)}$ since any Steiner tree in $\mathcal{S}_2$ has at least $2(d-1)$ terminal edges and there are kd terminal edges in total. We complete the proof of the lemma by showing that $\mathcal{S}_1 \leq d-2$.

First, observe that each Steiner tree T' in $\mathcal{S}_1$ has a Steiner node of degree ≥ 3; otherwise, if each Steiner node in T' has degree two, then $\sum_{i=1}^{d} \deg_{T'}(t_i)$, the sum of the degrees of the terminals in T', would equal $2(d-1)$; this would contradict the fact that T' has $\leq 2d-3$ terminal edges since it is in $\mathcal{S}_1$. Consider the union of all Steiner trees in $\mathcal{S}_1$, and contract all white edges in the union. In the obtained graph, each terminal node is on the outer face, and each modified Steiner tree has a Steiner node of degree at least 3. Note that the modified Steiner trees are still element-disjoint, so the number of Steiner nodes of degree at least 3 is an upper bound on $|\mathcal{S}_1|$. Also note that the obtained graph is a bipartite planar graph with all the terminal nodes on the outer face. Let n_2 be the number of Steiner nodes of degree 2 and let n_3 be the number of Steiner nodes of degree at least 3 in this graph. Now add a new node to the outer face of this graph and connect it to all d terminal nodes. The obtained graph is a bipartite planar graph with at least $2n_2 + 3n_3 + d$ edges and at most $2(n_2 + n_3 + 1 + d) - 4$ edges. Thus, we have:

$$2n_2 + 3n_3 + d \leq 2(n_2 + n_3 + 1 + d) - 4 \Longrightarrow |\mathcal{S}_1| \leq n_3 \leq d - 2.$$

□

Claim 4.3 *The linear program* `LP-element` *has objective value $z_{LP}(G) = k$.*

Proof: To show this we construct k pairs of half-integral Steiner trees. Each pair is obtained from two consecutive rows by connecting each terminal to these two rows using two consecutive columns. Figure 9 shows two pairs of half-integral Steiner trees and shows how they cross each other. It is easy to check that each Steiner node is contained in at most 2 half-integral Steiner trees. Hence, these $2k$ Steiner trees form a feasible solution to `LP-element`. Thus, we have $z_{LP} \geq k$; moreover, $z_{LP} \leq k$ since each terminal has degree k. □

4.2 Integrality ratio example for packing edge-disjoint Steiner trees

In this subsection, we show that the following linear programming relaxation of the edge-disjoint version of the problem has integrality ratio approaching 2. Let $\mathcal{T}$ be the set of all R-Steiner trees in $G = (V, E)$.

$$(\texttt{LP-edge}) \quad z_{LP}(G) = \max \sum_{T \in \mathcal{T}} x_T$$

subject to

$$\sum_{T \in \mathcal{T}: e \in T} x_T \leq 1 \qquad \forall e \in E$$

$$x_T \geq 0 \qquad \forall T \in \mathcal{T}$$

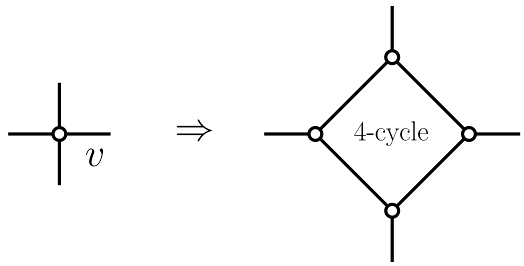


Figure 10: Gadget for degree 4 nodes

Theorem 4.4 *The standard linear programming relaxation of the edge-disjoint* Steiner Tree Packing *problem has an integrality ratio* $\geq 2 - \frac{2}{|R|} - \epsilon$ *even on planar graphs, where the additive term* ϵ *is a function of* k *and* $|R|$ *and for fixed* $|R|$, $\epsilon \to 0$ *as* $k \to \infty$ *(here,* k *denotes the edge-connectivity of the terminals).*

Construction: Start from a $k \times kd$ grid (with $k+1$ horizontal lines and $kd+1$ vertical lines); then add a set of d terminal nodes, $R = \{t_1, \ldots, t_d\}$, to the outer face of the grid. Connect terminal t_1 to the first k consecutive nodes on the bottom-most row of the grid, next, connect terminal t_2 to the second k consecutive nodes on the bottom-most row, and continue in this way to connect each terminal to a set of k consecutive nodes. Note that the terminals have disjoint sets of neighbors. Now, replace each Steiner node of degree 4 in the obtained graph by the gadget shown in Figure 10. Let G be the obtained graph. Note that G is a planar graph. There are k edge-disjoint paths between any two terminal nodes; each path is formed by using one row and two columns of the grid. Hence, the terminal set R is k-edge-connected. Observe that each Steiner node has degree ≤ 3 in G. Therefore, edge-disjoint Steiner trees in G are also

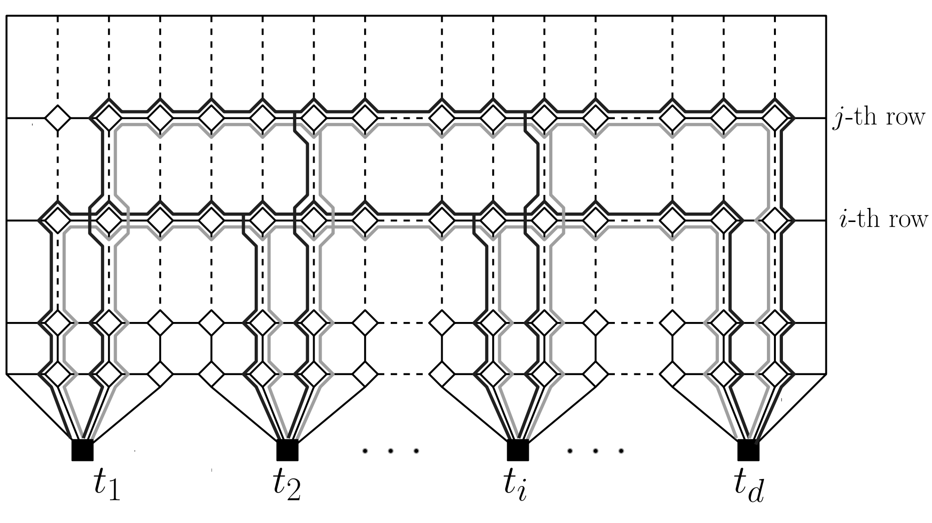


Figure 11: Integrality ratio example for the edge-disjoint problem

element-disjoint. Applying Lemma 4.2, we see that G has at most $\frac{kd}{2(d-1)} + d - 2$ element-disjoint (or, edge-disjoint) Steiner trees. Now, we only need to show that `LP-edge` has optimal value of $z_{LP}(G) = k$. Proving this claim will complete the proof of Theorem 4.4.

Claim 4.5 *The linear program* `LP-edge` *has objective value* $z_{LP}(G) = k$.

Proof: The optimal value $z_{LP}(G)$ is at most k since each terminal has degree k. Now, we create $2k$ half-integral Steiner trees, and this shows that $z_{LP}(G) \geq k$. Corresponding to each row of the original grid we construct a pair of half-integral Steiner trees. The ℓth pair uses the ℓth row and columns $\ell, k+\ell, \ldots, ik+\ell, \ldots, dk+\ell$ of the grid. One of the trees in this pair uses the upper two edges of the 4-cycles and the other tree uses the lower two edges of the 4-cycles of the ℓth row. Similarly one of them uses the right two edges of the 4-cycles and the other one uses the left two edges of the 4-cycles of the columns. In Figure 11 Steiner trees for the ith row and the jth row are shown. It is easy to check that each edge of the modified grid is contained in at most two Steiner trees; Figure 11 shows how two Steiner trees cross each other. Hence, the k pairs of half-integral Steiner trees form a solution to `LP-edge` of value k. □

Acknowledgements: We thank Guyslain Naves for suggesting the modifications to the construction for Theorem 3.7 that give the construction for Theorem 3.9.

References

[1] G. Calinescu, C. Chekuri, and J. Vondrak. Disjoint bases in a polymatroid. *Random Structures and Algorithms*, 35(4):418–430, 2009.

[2] C. Chekuri and N. Korula, A graph reduction step preserving element-connectivity and applications, *Proc. ICALP* 2009, Springer LNCS 5555, 254–265.

[3] J. Cheriyan and M.R. Salavatipour. Hardness and approximation results for packing Steiner trees. *Algorithmica*, 45(1):21–43, 2006.

[4] J. Cheriyan and M.R. Salavatipour. Packing element-disjoint Steiner trees. *ACM Transactions on Algorithms*, 3(4), 2007.

[5] E. D. Demaine, M.T. Hajiaghayi and P. N. Klein, Node-weighted Steiner tree and group Steiner tree in planar graphs, *Proc. ICALP* 2009, Springer LNCS 5555, 328–340.

[6] J. Edmonds, Minimum partition of a matroid into independent subsets. *Journal of Research National Bureau of Standards Section B*, 69:67–72, 1965.

[7] P. Floréen, P. Kaski, J. Kohonen, and P. Orponen. Lifetime maximization for multicasting in energy-constrained wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):117–126, 2005.

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

[9] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

[10] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: a cutting plane algorithm and computational results. *Mathematical Programming*, 72(2):125–145, 1996.

[11] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: further facets. *European J. Combinatorics*, 17(1):39–52, 1996.

[12] M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: polyhedral investigations. *Mathematical Programming A*, 72(2):101–123, 1996.

[13] M. Grotschel, A. Martin, and R. Weismantel. Packing Steiner trees: Separation algorithms. *SIAM J. Discret. Math.*, 9(2):233–257, 1996.

[14] M. Grötschel, A. Martin, and R. Weismantel. The Steiner tree packing problem in VLSI design. *Mathematical Programming*, 78(2):265–281, 1997.

[15] 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. Comput. Syst. Sci.*, 67(3):473–496, 2003.

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

[17] K. Jain, M. Mahdian, and M.R. Salavatipour. Packing Steiner trees. In *SODA*, pages 266–274, 2003.

[18] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991.

[19] P. Kaski. Packing Steiner trees with identical terminal sets. *Inf. Process. Lett.*, 91(1):1–5, 2004.

[20] B. Korte, H. J. Prömel, and A. Steger. Steiner trees in VLSI-layout. In B. Korte, L. Lovász, H. J. Prmel, and A. Schrijver, editors, *Paths, Flows, and VLSI-Layout*, pages 185–214. Springer-Verlag, Berlin, Germany, 1990.

[21] A.V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

[22] L.C. Lau. *On approximate min-max theorems for graph connectivity problems*. PhD thesis, University of Toronto, 2006.

[23] L.C. Lau. An approximate max-Steiner-tree-packing min-Steiner-cut theorem. *Combinatorica*, 27(1):71–90, 2007.

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

[25] M. Middendorf and F. Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica*, 13(1):97–107, 1993.

[26] C. S. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathemathical Society*, 36:445–450, 1961.

[27] G. Naves. The hardness of routing two pairs on one face. *Mathematical Programming*, 1–21, 2010.

[28] G. Naves and A. Sebő. Multiflow feasibility: an annotated tableau. In *Research Trends in Combinatorial Optimization*, pages 261–283. Springer, Berlin Heidelberg, 2008.

[29] J. Plesník. The NP-completeness of the Hamiltonian cycle problem in planar digraphs with degree bound two. *Inf. Process. Lett.*, 8(4):199–201, 1979.

[30] A. Thomason. An extremal function for contractions of graphs. *Math. Proc. Cambridge Philos. Soc.*, 95:261–265, 1984.

[31] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathemathical Society*, 36:221–230, 1961.

[32] D. Wagner. Simple algorithms for Steiner trees and paths packing problems in planar graphs. *CWI Quarterly*, 6(3):219–240, 1993.

[33] D. B. West and H. Wu. Packing of Steiner trees and S-connectors in graphs. *Accepted for journal publication*, 2010.

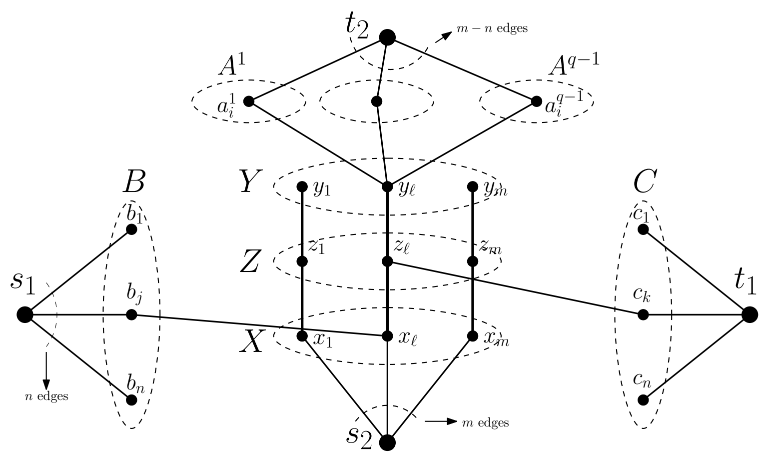


Figure 12: hardness construction

A APX-hardness of the Integer2Commodity problem

In this appendix, we give a full proof of the following result due to Guruswami et al. [15, Corollary 4.1]. The construction in our proof uses a minor modification of their construction; the modification appears to be essential for the proof.

Theorem A.1 *The* INTEGER2COMMODITY *problem is APX-hard.*

Proof: The proof is by a reduction from the BOUNDED 3-DIMENSIONAL MATCHING (B3DM) problem. In the B3DM problem we are given three disjoint sets $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_n\}$, $C = \{c_1, \dots, c_n\}$ and a set $\mathcal{E} = \{e_1, \dots, e_m\} \subseteq A \times B \times C$ of m triples. Moreover, each element in $A \cup B \cup C$ belongs to q triples. It is known [18] that there is a constant $\epsilon_0 > 0$ such that it is NP-hard to distinguish between instances of B3DM where there exists a *perfect* matching (i.e., n disjoint triples) and those instances where the number of disjoint triples is at most $(1 - \epsilon_0)n$. We start from an instance G of the B3DM problem, and construct an instance $\mathcal{I} = (H, s_1, t_1; s_2, t_2)$ of the INTEGER2COMMODITY problem as follows. For any positive integer p, we will use $[p]$ to denote the set $\{1, 2, \dots, p-1, p\}$.

$$V(H) = \{s_1, s_2, t_1, t_2\} \bigcup \{b_i, c_i : i \in [n]\} \bigcup \{a_i^p : i \in [n], p \in [q-1]\} \bigcup \{x_i, y_i, z_i : i \in [m]\}$$

$$\begin{aligned} E(H) &= \{\{s_1, b_i\}, \{t_1, c_i\} : i \in [n]\} \bigcup \{\{t_2, a_i^p\} : i \in [n], p \in [q-1]\} \\ &\bigcup \{\{s_2, x_\ell\}, \{x_\ell, z_\ell\}, \{z_\ell, y_\ell\}, \{x_\ell, b_{j_\ell}\}, \{z_\ell, c_{k_\ell}\} : e_\ell = (a_{i_\ell}, b_{j_\ell}, c_{k_\ell}) \in \mathcal{E}\} \\ &\bigcup \left\{\left\{y_\ell, a_{i_\ell}^p\right\} : e_\ell = (a_{i_\ell}, b_{j_\ell}, c_{k_\ell}) \in \mathcal{E}, p \in [q-1]\right\} \end{aligned}$$

The graph H has four nodes s_1, s_2, t_1, t_2, where (s_1, t_1) and (s_2, t_2) form the two demand pairs in H. Corresponding to each element $b_i \in B$ or $c_i \in C$, we have a node in H, and corresponding to each element $a_i \in A$ we have $q - 1$ nodes $a_i^1, \dots, a_i^{q-1}$ in H. In addition, there are three nodes x_ℓ, y_ℓ, z_ℓ corresponding to each triple $e_\ell \in \mathcal{E}$. Let $X = \{x_\ell : \ell \in [m]\}, Y = \{y_\ell : \ell \in [m]\}, Z = \{z_\ell : \ell \in [m]\}$. Node s_1 is connected to the nodes corresponding to the elements in B, and t_1 is connected to the nodes corresponding to the elements in C. We connect t_2 to all $(q-1)n$ nodes corresponding to the elements in A. For each triple

$e_\ell = (a_{i_\ell}, b_{j_\ell}, c_{k_\ell}) \in \mathcal{E}$, we connect x_ℓ to s_2, b_{j_ℓ} and z_ℓ, we connect z_ℓ to y_ℓ and c_{k_ℓ}, and we connect y_ℓ to all $(q-1)$ copies of the element a_{i_ℓ}. In other words, for each triple $e_\ell \in \mathcal{E}$, H has a subpath x_ℓ, y_ℓ, z_ℓ, and also H has edges between x_ℓ and the B-node of e_ℓ, between z_ℓ and the C-node of e_ℓ, and between y_ℓ and all $q-1$ nodes corresponding to the A-element of e_ℓ. (See Figure 12 for an illustration.) The goal is to find a maximum-cardinality set of edge-disjoint paths such that each path has end nodes s_1, t_1 or has end nodes s_2, t_2. Note that the number of such paths is $\leq m$, because both s_1 and t_1 have degree n, and t_2 has degree $n(q-1) = m-n$. Note that the induced subgraph on $\{a_i^p : i \in [n], p \in [q-1]\} \cup Y$ is a collection of $n = |A|$ copies of the complete bipartite graph $K_{q-1,q}$; each copy of the complete bipartite graph corresponds to an element $a_i \in A$, where one part of the bipartition contains all $q-1$ nodes corresponding to a_i and the other part contains the nodes y_ℓ corresponding to the q triples e_ℓ that contain a_i.

Now, we prove:

(a) [completeness] if $G = (A \cup B \cup C, \mathcal{E})$ has a perfect matching, then there are m edge-disjoint s_i-t_i paths for $i = 1, 2$ in total, and

(b) [soundness] if every matching in G has at most $(1-\epsilon_0)n$ triples, then there are at most $(1-\epsilon)m$ edge-disjoint s_i-t_i paths for $i = 1, 2$ in total, where $\epsilon = \frac{\epsilon_0}{4q}$.

Lemma A.2 (Completeness) *If G has a perfect matching $\mathcal{M}$, then the instance $\mathcal{I}$ of the* INTEGER2COMMODITY *problem has objective value m.*

Proof: There are n disjoint triples in $\mathcal{M}$. For each triple $e_\ell = (a_{i_\ell}, b_{j_\ell}, c_{k_\ell}) \in \mathcal{M}$, we define a path $P_\ell = s_1, b_{j_\ell}, x_\ell, z_\ell, c_{k_\ell}, t_1$. This gives n edge-disjoint s_1-t_1 paths. Now, corresponding to the $m-n$ triples in $\mathcal{E} \setminus \mathcal{M}$ we define $m-n$ paths between s_2 and t_2. Consider an element $a_i \in A$, and note that it appears in exactly $q-1$ triples in $\mathcal{E} \setminus \mathcal{M}$, because each element appears in q triples in $\mathcal{E}$ and $\mathcal{M}$ is a perfect matching. Let $e_{\ell_1}, \ldots e_{\ell_{q-1}}$ be these $q-1$ triples. For each triple e_{ℓ_j}, let Q_j^i be the path $s_2, x_{\ell_j}, z_{\ell_j}, y_{\ell_j}, a_i^j, t_2$. Thus, the path for e_{ℓ_j} uses the jth node corresponding to the element a_i. Considering these $q-1$ paths for each element in A gives us $(q-1)n = m-n$ edge-disjoint paths between s_2 and t_2. We have m paths in total, since we have n s_1-t_1 paths P_ℓ and $m-n$ s_2-t_2 paths Q_j^i; observe that each of these m paths corresponds to a distinct triple in $\mathcal{E}$. It can be seen that all of these m paths are edge disjoint. Hence, the instance $\mathcal{I}$ has a collection of m edge-disjoint s_i-t_i paths for $i = 1, 2$ in total. □

Lemma A.3 (Soundness) *If the instance $\mathcal{I}$ has a collection of $(1-\epsilon)m$ edge-disjoint s_i-t_i paths, then G has a perfect matching of size at least $(1-\epsilon_0)n$, where $\epsilon = \frac{\epsilon_0}{4q}$.*

Proof: Let $\mathcal{P}$ be a collection of $(1-\epsilon)m$ edge-disjoint s_i-t_i paths. First, we claim that there is a subcollection $\mathcal{P}' \subseteq \mathcal{P}$ of size $(1-3\epsilon)m$, such that each path in this subcollection uses exactly one node from X and exactly one node from Z. To see this, note that each node in X has degree 3, so the edge-disjoint paths in $\mathcal{P}$ are node-disjoint on X; moreover, $|X| = m$ and each s_i-t_i path uses at least one node from X; since $|\mathcal{P}| = (1-\epsilon)m$, the number of paths in $\mathcal{P}$ that use more than one node from X is $\leq \epsilon m$. Similarly, there are at most ϵm paths in $\mathcal{P}$ that use more than one node from the set Z. After removing these paths from $\mathcal{P}$, we get a subcollection of paths $\mathcal{P}'$ of cardinality at least $(1-3\epsilon)m$ such that each path uses exactly one node from X and exactly one node from Z. Also, note that the number of s_1-t_1 paths in $\mathcal{P}'$ is at least $(1-3\epsilon)m - (m-n) = n - 3\epsilon m$.

Each path in $\mathcal{P}'$ has the form $s_1, b_{j_\ell}, x_\ell, z_\ell, c_{j_\ell}, t_1$ (without loss of generality), and so contains exactly one edge of the form $\{x_\ell, z_\ell\}$; note that each such edge corresponds to a triple $e_\ell \in \mathcal{E}$. Thus, $\mathcal{P}'$ corresponds to a set of triples $\mathcal{T}$, and we have $|\mathcal{T}| = |\mathcal{P}'| \geq n - 3\epsilon m$. Observe that each element in $B \cup C$ appears in at most one triple in $\mathcal{T}$; otherwise, there will be two paths in $\mathcal{P}'$ that both contain either the same node b_j or the same node c_k. But each element in A may appear in any number of triples in $\mathcal{T}$. Let α_i denote the number of triples in $\mathcal{T}$ that contain the element a_i. Now we pick a subset $\mathcal{T}'$ of disjoint triples from $\mathcal{T}$ as follows. For each $a_i \in A$ with $\alpha_i \geq 1$, we add to $\mathcal{T}'$ an arbitrary triple from $\mathcal{T}$ that contains a_i, and we discard the other $\alpha_i - 1$ triples containing a_i. We claim that $\mathcal{T}'$ has at least $(1-\epsilon_0)n$ disjoint triples. Let us partition A into A_0, A_1, A_2, where A_0 is the set of a_i's with $\alpha_i = 0$, A_1 is the set of a_i's with $\alpha_i = 1$, and A_2 is the set of a_i's with $\alpha_i \geq 2$. Now focus on the s_2-t_2 paths in $\mathcal{P}$, and note that there are at least $(1-\epsilon)m-n = (q-1)n-\epsilon m$ such paths. We claim that there are at most $(q-1)(|A_0|+|A_1|)+\sum_{a_i\in A_2}(q-\alpha_i)$ s_2-t_2 paths in $\mathcal{P}$; we defer the proof of this claim. The claim implies that

$$\begin{aligned}(q-1)n - \epsilon m &\leq (q-1)(|A_0|+|A_1|) + \sum_{a_i\in A_2}(q-\alpha_i) \\ &= (q-1)(|A_0|+|A_1|+|A_2|) - \sum_{a_i\in A_2}(\alpha_i - 1).\end{aligned}$$

Hence, we have:

$$\sum_{a_i\in A_2}(\alpha_i - 1) = \sum_{a_i\in A,\ \alpha_i\geq 1}(\alpha_i - 1) \leq \epsilon m.$$

Observe that the number of triples in $\mathcal{T}\setminus\mathcal{T}'$ is $\sum_{a_i\in A,\ \alpha_i\geq 1}(\alpha_i - 1)$, because we discard $(\alpha_i - 1)$ triples from $\mathcal{T}$ for each element $a_i \in A$ with $\alpha_i \geq 1$. Thus, $|\mathcal{T}\setminus\mathcal{T}'|$ is at most ϵm, and recall that $|\mathcal{T}| \geq n - 3\epsilon m$. Therefore, $\mathcal{T}'$ has at least $n - 4\epsilon m = (1-\epsilon_0)n$ disjoint triples. This completes the proof of the lemma.

To prove the above claim (on the number of s_2-t_2 paths in $\mathcal{P}$), we first remove from H all edges in the union of the s_1-t_1 paths in $\mathcal{P}'$, and next, we find an s_2-t_2 edge cut in the remaining graph of an appropriate size. Denote by H' the graph obtained after removing the edges of all the s_1-t_1 paths in $\mathcal{P}'$. Define $F \subseteq E(H')$ to contain all edges between t_2 and the nodes corresponding to the elements in $A_0 \cup A_1$, and moreover, for each element $a_i \in A_2$, let F contain every edge $\{y_\ell, z_\ell\}$ in H' that corresponds to a triple $e_\ell \in \mathcal{E}\setminus\mathcal{T}$ such that e_ℓ contains a_i. Observe that for each $a_i \in A_2$, H has q subpaths x_ℓ, y_ℓ, z_ℓ (corresponding to the q triples that contain a_i), and α_i of these subpaths are used by the s_1-t_1 paths in $\mathcal{P}'$ (each of these s_1-t_1 paths uses an edge of the form $\{x_\ell, y_\ell\}$), and the remaining $q - \alpha_i$ subpaths each contribute an edge of the form $\{y_\ell, z_\ell\}$ to F. We claim that F is an s_2-t_2 edge cut in H'. To see this, remove F from H' and note that the resulting graph has no s_2-t_2 path; we omit the details for this verification. Therefore, F forms an s_2-t_2 edge cut in H' of size $(q-1)(|A_0|+|A_1|)+\sum_{a_i\in A_2}(q-\alpha_i)$, hence, this gives an upper bound on the number of s_2-t_2 paths in $\mathcal{P}$. $\square$

Thus, the theorem follows. $\square$