Packing Steiner Trees: Separation Algorithms

M. Grötschel A. Martin R. Weismantel

Abstract

In this paper we investigate separation problems for classes of inequalities valid for the polytope associated with the Steiner tree packing problem, a problem that arises, e. g., in VLSI routing. The separation problem for Steiner partition inequalities is $\mathcal{NP}$-hard in general. We show that it can be solved in polynomial time for those instances that come up in switchbox routing. Our algorithm uses dynamic programming techniques. These techniques are also applied to the much more complicated separation problem for alternating cycle inequalities. In this case we can compute in polynomial time, given some point y, a lower bound for the gap $\alpha - a^T y$ over all alternating cycle inequalities $a^T x \geq \alpha$. This gives rise to a very effective separation heuristic. A by-product of our algorithm is the solution of a combinatorial optimization problem that is interesting in its own right: Find a shortest path in a graph where the "length" of a path is its usual length minus the length of its longest edge.

1 Introduction

To introduce the problem we are considering let us begin with a few defintions. We are given a graph $G = (V, E)$. If T is a subset of V, then an edge set $S \subseteq E$ is called a *Steiner tree in G for T* if the subgraph induced by S contains a path from s to t for every pair s, t of nodes in T. We will call the elements of T *terminals* and T *terminal set* or *net*. We are further given a list $\mathcal{N} = \{T_1, \ldots, T_N\}$, $N \geq 1$, of nets, i. e., subsets of V, and moreover, for each edge $e \in E$, a positive *capacity* $c_e \in \mathbb{N}$. A *Steiner tree packing* is an N-tuple $(S_1, \ldots, S_N)$ of edge sets $S_k \subseteq E$ such that each set S_k is a Steiner tree in G for T_k, $k = 1, \ldots, N$, and such that each edge $e \in E$ is contained in at most c_e of these Steiner trees. The *Steiner tree packing problem* is the task to decide whether, for a given graph $G = (V, E)$ with edge capacities $c_e \in \mathbb{N}$ and for a given net list $\mathcal{N}$, a Steiner tree packing exists. The ultimate goal of the investigation is to find a minimum weight Steiner tree packing with respect to some given weight function on the edges.

In [GMW92b] we have shown how the Steiner tree packing problem can be employed to model various versions of the routing problem in VLSI design. We have demonstrated that a cutting plane method based on polyhedral investigations can be successfully utilized for the optimal solution of small real routing problems and that good lower bounds on the optimum solution value can be computed in acceptable running time. The cornerstone of our cutting plane algorithm is an effective implementation of exact and heuristic separation routines for various classes of inequalities, introduced in [GMW92a], that are valid and under mild assumptions facet-defining for the associated Steiner tree packing polyhedron. The design and investigation of these separation algorithms are the subject of this paper.

2 The Polyhedral Approach and Some Basic Results

In this section we define the Steiner tree packing polyhedron and describe some basic polyhedral results. We start by introducing some graphtheoretic notation.

We denote graphs by $G = (V, E)$, where V is the node set and E the edge set. All graphs we consider are undirected, loopless and finite. For a given edge set $F \subseteq E$, we denote by $V(F)$ all nodes that are incident to an edge in F. An edge e with endnodes u and v is also denoted by uv. Given two node sets $U, W \subseteq V$, we denote by $[U : W]$ the set of edges in G with one endnode in U and the other in W. For a node set W, we also use $E(W)$ instead of $[W : W]$. A set of node sets $V_1, \ldots, V_p \subset V$, $p \geq 2$, is called a *partition of* V if all sets V_i are nonempty, the node sets are mutually disjoint and the union of these sets is V. (Note that we use "$\subset$" to denote strict set theoretic containment.) If $V_1, \ldots, V_p$ is a partition of V, then $\delta(V_1, \ldots, V_p)$ denotes the set of edges in G whose endnodes are in different sets. We call $\delta(V_1, \ldots, V_p)$ a *multicut (with p shores) induced by* $V_1, \ldots, V_p$. For $W \subset V$, $W \neq \emptyset$, we write $\delta(W)$ instead of $\delta(W, V \setminus W)$ and call this set the *cut induced by* W. We abbreviate $\delta(\{v\})$ by $\delta(v)$. For an edge set F, we define $d_F(v) := |\delta(v) \cap F|$, which is the *degree* of v in the subgraph (V, F) of G. With a planar graph G we always associate a fixed embedding of G in the plane. The set of edges that are incident to the outer face of a planar graph $G = (V, E)$ will be denoted by $\mathrm{O}_G(E)$. For a subset of edges $S \subseteq E$, we define $\mathrm{O}_G(S) := \mathrm{O}_{(V(S),S)}(S)$, i. e., $\mathrm{O}_G(S)$ denotes the set of outer face edges of the graph induced by S.

Let $K = (v_0, e_1, v_1, e_2, \ldots, v_{l-1}, e_l, v_l)$ be a sequence of nodes and edges, where each edge e_i is incident with the nodes v_{i-1} and v_i for $i = 1, \ldots, l$, and where the edges are pairwise disjoint and the nodes distinct (except possibly v_0 and v_l). K is called a *path* (or a $[v_0, v_l]-$*path*), if $v_0 \neq v_l$, and a *cycle*, if $v_0 = v_l$ and $l \geq 2$. The nodes $v_1, \ldots, v_{l-1}$ of a path K are called the *inner nodes* of K. Each edge that connects two nodes of a cycle (path) K and that is not in K is called a *diagonal of* K. We say that two diagonals uv and $u'v'$ *cross with respect to* K if the corresponding nodes appear in the sequence u, u', v, v' or u, v', v, u' by walking along the cycle (path). Similarly, we call two sets of diagonals F_1 and F_2 *cross free* if, for all $e_1 \in F_1$ and $e_2 \in F_2$, e_1 and e_2 do not cross. Otherwise, F_1 and F_2 are *crossing*. For our purposes, it is convenient to consider a path P or a cycle C, respectively, as a subset of the edge set. We call an edge set B a *tree* if $(V(B), B)$ is connected and contains no cycle. The *leaves* of a tree B are the nodes that are incident to exactly one edge of B.

Note that a Steiner tree is not a tree, in general. (Our Steiner trees are supersets of "ordinary" Steiner trees. We employ this slight change of the more standard definition, since it simplifies a number of technicalities of our polyhedral investigations.) A Steiner tree that is a tree and whose leaves are terminals is called *edge-minimal Steiner tree*. We call an edge e in a graph G, given some net T, a *Steiner bridge*, if every Steiner tree for T in G contains e.

We now introduce a polytope associated with the Steiner tree packing problem. We are given a graph $G = (V, E)$ with capacities $c_e \in \mathbb{N}$ for all $e \in E$ and a net list $\mathcal{N} = \{T_1, \ldots, T_N\}$, $N \geq 1$. We will denote an *instance of the Steiner tree packing problem* by the tripel $(G, \mathcal{N}, c)$. Let $\mathbb{R}^{\mathcal{N} \times E}$ denote the $N \cdot |E|$ – dimensional vector space $\mathbb{R}^E \times \ldots \times \mathbb{R}^E$, where the components of each vector $x \in \mathbb{R}^{\mathcal{N} \times E}$ are indexed by x_e^k for $k \in \{1, \ldots, N\}$, $e \in E$. Moreover, for a vector $x \in \mathbb{R}^{\mathcal{N} \times E}$ and $k \in \{1, \ldots, N\}$, we denote by $x^k \in \mathbb{R}^E$ the vector $(x_e^k)_{e \in E}$, and we simply write $x = (x^1, \ldots, x^N)$ instead of $x = ((x^1)^T, \ldots, (x^N)^T)^T$. For a subset $E' \subseteq E$ and a vector $a \in \mathbb{R}^{\mathcal{N} \times E}$, we define a vector $a|_{E'} \in \mathbb{R}^{\mathcal{N} \times E'}$ by $(a|_{E'})_e^k := a_e^k$ for all $k = 1, \ldots, N$ and $e \in E'$. For an edge set $F \subseteq E$, $\chi^F \in \mathbb{R}^E$ denotes the *incidence vector of* F, i. e., $\chi_e^F := 1$, if $e \in F$, and $\chi_e^F := 0$, otherwise. Conversely, for each 0/1-vector $x \in \mathbb{R}^E$, the set $I_x := \{e \in E \mid x_e = 1\}$ is called the *incidence set of* x. The *incidence vector of a Steiner tree packing* $(S_1, \ldots, S_N)$ is denoted by $(\chi^{S_1}, \ldots, \chi^{S_N})$.

The *Steiner tree packing polyhedron* STP $(G, \mathcal{N}, c)$ is the convex hull of all incidence vectors of Steiner tree packings. It is easy to see that the following holds.

$$
\text{(2.1)} \qquad \begin{aligned}
\mathrm{STP}\,(G, \mathcal{N}, c) = \mathrm{conv}\,\{x \in \mathbb{R}^{\mathcal{N} \times E} \mid \\
&\text{(i)} \quad \sum_{e \in \delta(W)} x_e^k \geq 1, \text{ for all } W \subset V,\ W \cap T_k \neq \emptyset, \\
&\qquad\qquad\qquad (V \setminus W) \cap T_k \neq \emptyset,\ k = 1, \ldots, N; \\
&\text{(ii)} \quad \sum_{k=1}^{N} x_e^k \leq c_e, \quad \text{for all } e \in E; \\
&\text{(iii)} \quad 0 \leq x_e^k \leq 1, \quad \text{for all } e \in E,\ k = 1, \ldots, N; \\
&\text{(iv)} \quad x_e^k \in \{0, 1\}, \quad \text{for all } e \in E,\ k = 1, \ldots, N\}.
\end{aligned}
$$

The inequalities (2.1) (i) are called *Steiner cut inequalities*, inequalities (2.1) (ii) are called *capacity inequalities* and the ones in (2.1) (iii) *trivial inequalities*. In case $N = 1$, the Steiner tree packing polyhedron is also called the *Steiner tree polyhedron*. Note that (2.1) (i) – (iv) yields an integer programming formulation of the weighted Steiner tree packing problem.

We close this section by listing some polyhedral results that are of importance for the remainder of the paper. The reader interested in the proofs of these results is referred to [GMW92a].

First of all, the problem of deciding whether, for some given $l \in \mathbb{N}$, the dimension of the Steiner tree packing polyhedron is at least l is $\mathcal{NP}$-complete. This follows from the fact that the Steiner tree packing problem itself is $\mathcal{NP}$-complete (see, for instance, [KL84], [S87]). Due to this fact, we have decided to study the Steiner tree packing polyhedron for problem instances, for which the dimension can easily be determined and to look for facet-defining inequalities for these special instances. The justification of the choice to be described below can be found in [GMW92a].

We restrict ourselves to considering instances $(G, \mathcal{N}, c)$, where the graph G is complete, the net list $\mathcal{N} = \{T_1, \ldots, T_N\}$ is *disjoint* (i. e. $T_i \cap T_j = \emptyset$ for all $i, j \in \{1, \ldots, N\}$, $i \neq j$) and the capacities are equal to one ($c = \mathbb{1}$). It can easily be verified that the corresponding Steiner tree packing polyhedron STP $(G, \mathcal{N}, \mathbb{1})$ is fulldimensional in this case. The subsequent lemma shows how validity results for the Steiner tree packing polyhedron for some graph can be transformed to validity results for the Steiner tree packing polyhedron for the graph obtained by deleting some edge or splitting some node and thus, by repeated application, how validity results for the complete graph can be transformed to the genral case.

Lemma 2.2 *Let $(G, \mathcal{N}, c)$ be an instance of the Steiner tree packing problem.*

(a) (Deleting an edge) *Let $a^T x \geq \alpha$ be a valid inequality of $STP(G, \mathcal{N}, c)$ and suppose $f \in E$ is deleted from G. Then $\hat{a}^T x \geq \alpha$ is a valid inequality of $STP(G \setminus f, \mathcal{N}, c|_{E \setminus \{f\}})$ where $\hat{a}^k_e = a^k_e$ for all $e \in E \setminus \{f\}$, $k \in \{1, \ldots, N\}$ (where $G \setminus f$ denotes the graph that is obtained by deleting edge f).*

(b) (Splitting a node) *Let $f \in E$ and let $\hat{a}^T x \geq \alpha$ be a valid inequality of $STP(G\,/\,f, \hat{\mathcal{N}}, \hat{c})$ ($G\,/\,f$ denotes the graph that is obtained by shrinking edge f, $\hat{\mathcal{N}}$ and $\hat{c}$ denote the corresponding net list and capacity vector defined on $G\,/\,f$). Then, $a^T x \geq \alpha$ defines a valid inequality for $STP(G, \mathcal{N}, c)$ with $a^k_e = \hat{a}^k_e$ for all $e \in E \setminus \{f\}$, $k \in \{1, \ldots, N\}$ and $a^k_f = 0$ for all $k = 1, \ldots, N$.*

The next theorem shows that each nontrivial facet-defining inequality of the Steiner tree polyhedron can be lifted to yield a facet-defining inequality of the Steiner tree packing polyhedron.

Theorem 2.3 *Let $G = (V, E)$ be the complete graph with node set V and let $\mathcal{N} = \{T_1, \ldots, T_N\}$, $N \geq 2$, be a disjoint net list. Let $\bar{a}^T x \geq \alpha$, $\bar{a} \in \mathbb{R}^E$, be a nontrivial facet-defining inequality of $STP(G, \{T_l\}, \mathbb{1})$ for some $l \in \{1, \ldots, N\}$. Then, $a^T x \geq \alpha$ defines a facet of $STP(G, \mathcal{N}, \mathbb{1})$, where $a \in \mathbb{R}^{\mathcal{N} \times E}$ denotes the vector with $a_e^l = \bar{a}_e$, $a_e^k = 0$ for all $k = 1, \ldots, N, k \neq l$, $e \in E$.*

Theorem 2.3 implies that, in order to obtain a complete description of some Steiner tree packing polyhedron STP $(G, \mathcal{N}, c)$, at least all "individual" Steiner tree polyhedra STP $(G, \{T\}, c), T \in \mathcal{N}$, must be known completely. Of course, this knowledge will hardly do. There are many classes of inequalities that combine at least two nets. We call such inequalities *joint*. In [GMW92a] and [GMW93] several classes of joint inequalities are described.

Polyhedral results such as the ones mentioned above are utilized algorithmically by means of separation algorithms in the framework of a cutting plane method. We will discuss separation problems for classes of inequalities valid for the Steiner tree packing polyhedron and separation algorithms for these classes in the subsequent sections.

3 Separation of the Steiner Partition Inequalities

Let a graph $G = (V, E)$ and a set of terminals $T \subseteq V$, $|T| \geq 2$ be given. A partition $V_1, \ldots, V_p$, $p \geq 2$, of V is called a *Steiner partition (with respect to T)* if $V_i \cap T \neq \emptyset$ for $i = 1, \ldots, p$. The inequality

$$x(\delta(V_1, \ldots, V_p)) \geq p - 1$$

induced by a Steiner partition $V_1, \ldots, V_p$ is called a *Steiner partition inequality*. (Note that a Steiner cut inequality is the special case, where $p = 2$.) Obviously, each Steiner partition inequality is valid for STP $(G, \{T\}, \mathbb{1})$. The separation problem for this class of inequalities can be formulated as follows.

Problem 3.1 (Separation problem for the Steiner partition inequalities)
Let a graph $G = (V, E)$, a terminal set $T \subseteq V$, and a vector $y \in \mathbb{R}^E$ with $0 \leq y_e \leq 1$, for $e \in E$, be given. Decide whether y satisfies all Steiner partition inequalities. If not, find a Steiner partition inequality that y violates.

Problem 3.1 is $\mathcal{NP}$-hard in general (cf. [GMS92]). Restricting 3.1 to Steiner cut inequalities, i. e., the case $p = 2$, the separation problem can be solved in polynomial time by min-cut computations using any of the many polynomial time max-flow algorithms, see [AMO93]. We show now that, if we restrict the graph G to be planar and the set of terminals T to lie on the outer face of G, the separation problem 3.1 can be solved in time polynomial in the size of G and the encoding length of y. In the following we describe this algorithm.

It is shown in [GM90] that the following conditions are necessary and sufficient for a Steiner partition inequality induced by $V_1, \ldots, V_p$ to be facet-defining, provided that the graph G is connected and contains no Steiner bridge:

$$
(3.2) \qquad
\begin{array}{ll}
(i) & (V_i, E(V_i)) \text{ is connected for } i = 1, \ldots, p, \\
(ii) & (V_i, E(V_i)) \text{ contains no Steiner bridge with respect} \\
 & \text{to the terminal set } V_i \cap T \ (i = 1, \ldots, p), \text{ and} \\
(iii) & G(V_1, \ldots, V_p) \text{ is 2-node connected;}
\end{array}
$$

where $G(V_1, \ldots, V_p)$ is the graph obtained from G by contracting each node set of the partition to a single node. Moreover, the proof shows that each Steiner partition inequality that does not define a facet of STP $(G, \mathcal{N}, \mathbb{1})$ is the non-negative linear combination of facet-defining Steiner partition inequalities and trivial inequalities. Thus, we can restrict ourselves to solving Problem 3.1 for facet-defining Steiner partition inequalities.

We now describe how each edge set $\delta(V_1, \ldots, V_p)$ of a Steiner partition satisfying (3.2) (i) and (iii) can be viewed as a Steiner tree in a certain "dual" graph.

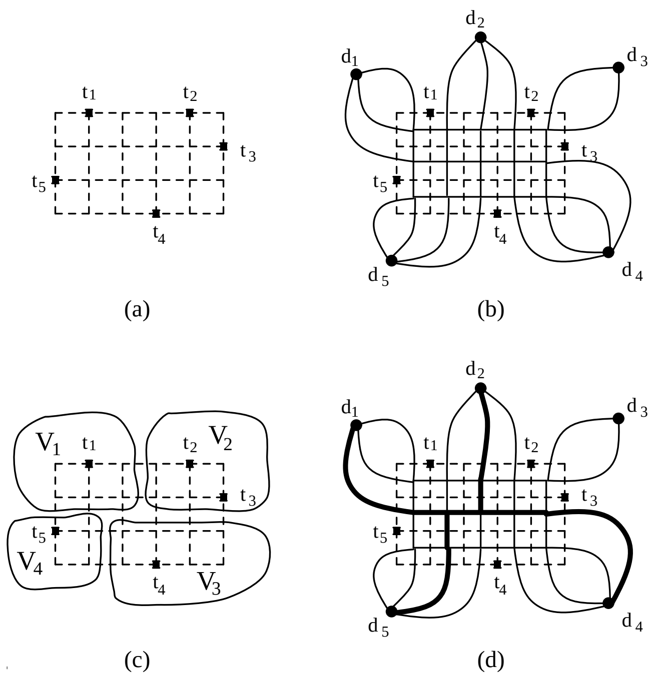

Figure 1:

For the remainder of this section we assume that the graph G is 2-node connected. We can do this without loss of generality, because otherwise the overall problem can be decomposed in an obvious way into subproblems where the corresponding graphs are 2-node connected. Thus, the edge set $\mathrm{O}_G(E)$ that encloses the outer face of G is a cycle. We may assume that the terminal set $T = \{t_1, \ldots, t_z\}$ is numbered in a clockwise fashion along this cycle. Let us consider the dual planar graph $G^* = (V^*, E)$ of G. We subdivide the node representing the outer face into z nodes $d_1, \ldots, d_z$ such that every edge of $\mathrm{O}_G(E)$ that is passed by walking from t_i to t_{i+1} on $\mathrm{O}_G(E)$ in clockwise order is now incident to d_{i+1} for $i = 1, \ldots, z$. (We identify an index $i > z$ with $((i-1) \text{ modulo } z) + 1$.) Let $G_D = (V_D, E)$ denote the resulting graph and set $T_D := \{d_1, \ldots, d_z\}$. We call T_D the *set of dual terminals*. Observe that, instead of working with a bijective mapping, we denote both the edge set of the original graph G and its "dual" G_D by the same symbol E. We make sure that this notational simplification will not lead to confusion. Figures 1 (a), showing a 4×6 grid with five terminals, and Figures 1 (b), where the edges of G_D are displayed by solid lines, illustrate this construction.

Let us now define the following set of Steiner trees in G_D.

$$\begin{array}{ll} \mathcal{D} := \{S \subseteq E \mid & S \text{ is an edge-minimal Steiner tree in } G_D \\ & \text{for some } J \subseteq T_D,\ |J| \geq 2, \text{ such that} \\ & d_S(j) = 1 \text{ for all } j \in J, \\ & d_S(t) = 0 \text{ for all } t \in T_D \setminus J\}. \end{array}$$

Clearly, every Steiner tree S in $\mathcal{D}$ determines the set $J \subseteq T_D$ of its terminals uniquely. For notational ease we will thus often write S_J to denote a Steiner tree S in $\mathcal{D}$ and its associated set J of dual terminals.

Lemma 3.3 *Let $G = (V, E)$ be a planar graph and T a set of terminals located on the outer face of G. Then, the following statements are true:*

1. *If $V_1, \ldots, V_p$ is a Steiner partition of V with respect to T satisfying (3.2) (i) and (iii), then the multicut $\delta(V_1, \ldots, V_p)$ viewed as an edge set of the dual graph G_D is a Steiner tree S_J contained in $\mathcal{D}$ with $|J| = p$.*

2. *If S_J is a Steiner tree in G_D contained in $\mathcal{D}$, then there exists a unique Steiner partition $V_1, \ldots, V_{|J|}$ of V with respect to T satisfying (3.2) (i) and (iii) such that $S_J = \delta(V_1, \ldots, V_{|J|})$.*

We will prove a similar statement in Lemma 4.4, so we omit the proof here. Lemma 3.3 shows that the Steiner partitions of V satisfying (3.2) (i) and (iii) are in one-to-one correspondence to the edge-minimal Steiner trees in G_D that are in $\mathcal{D}$. To illustrate this on an example, consider Figures 1 (c) and (d): the multicut $\delta(V_1, V_2, V_3, V_4)$ induced by the Steiner partition V_1, V_2, V_3, V_4 of V depicted in Figure 1 (c) is a Steiner tree in G_D for the subset $\{d_1, d_2, d_4, d_5\}$ of the dual terminals, see the thick solid lines in Figure 1 (d).

To check whether a given vector $y \in \mathbb{R}^E$, $y \geq 0$, satisfies all Steiner partition inequalities $x(\delta(V_1, \ldots, V_p)) \geq p - 1$, we determine the value

$$\alpha := \min_{S_J \in \mathcal{D}} \Big(y(S_J) - |J| \Big). \tag{3.4}$$

If $\alpha \geq -1$, Lemma 3.3 implies that there exists no violated Steiner partition inequality. Otherwise, the corresponding Steiner tree S_J yields the violated Steiner partition inequality $y(S_J) < |J| - 1$.

Observe that the objective function of the minimization problem in (3.4) is not linear. One way to linearize it is to consider the following 2-stage process. First, for every $J \subseteq T_D$ with $|J| \geq 2$ we determine a Steiner tree S_J^* for J in G_D such that the weight $y(S_J^*)$ is minimum, where only those Steiner trees S_J are considered that satisfy $d_{S_J}(j) = 1$ for all $j \in J$ and $d_{S_J}(j) = 0$ for all $j \in T_D \setminus J$. Then we determine, among all these Steiner trees S_J^*, $J \subseteq T_D$ with $|J| \geq 2$, a Steiner tree $S_{J^*}^*$ such that the value $y(S_{J^*}^*) - |J^*|$ is as small as possible. In other words (3.4) can be written in the following way:

$$\alpha = \min_{\substack{J \subseteq T_D \\ |J| \geq 2}} \Big(\min_{\substack{S \text{ Steiner tree for } J \\ S \in \mathcal{D}}} y(S) \Big) - |J|. \tag{3.5}$$

However, this does not lead to a polynomial time algorithm. Our approach for the computation of α is based on ideas of [DW71] and [EMV87] who have presented a dynamic programming algorithm for the solution of the following problem.

Suppose, we are given a graph $G = (V, E)$ and a set of terminals Z, and we want to compute a minimal (with respect to some weighting $w : E \to \mathbb{R}_+$) Steiner tree for Z. The idea of the algorithm is based on the observation that, for every minimal Steiner tree S and every node $v \in V(S)$ that is not a leaf of S, there exists a subset $J \subseteq Z$ such that S can be split into two subtrees S_1 and S_2, where S_1 is an optimal Steiner tree with respect to $J \cup \{v\}$ and S_2 is an optimal Steiner tree with respect to $(Z \setminus J) \cup \{v\}$. This observation leads to the following recursion formula.

For $J \subseteq Z$ and $v \in V$, let $S(J \cup \{v\})$ denote a minimal Steiner tree in G for $J \cup \{v\}$. Moreover, let $S_v(J \cup \{v\})$ be a minimal Steiner tree in G with respect to $J \cup \{v\}$, where we require in addition $d_{S_v(J \cup \{v\})}(v) \geq 2$, if $v \notin Z$. Then, we obtain (see [DW71])

(i) $w(S_v(J \cup \{v\})) = \min_{\emptyset \subset I \subset J} w(S(I \cup \{v\})) + w(S((J \setminus I) \cup \{v\}))$,

(ii) $w(S(J \cup \{v\})) = \min_{u \in V} w(W(v, u)) + w(S_u(J \cup \{u\}))$;

where $W(u, v)$, $u, v \in V$ denotes a shortest path from u to v in G. Of course, for arbitrary graphs G and terminal sets Z, the running time of the dynamic program based on this recursion is exponential in the number of terminals. However, in the particular case, where G is planar and all terminals lie on the outer face of G, Erickson, Monma and Veinott (cf. [EMV87]) showed that it suffices to consider only subsets of Z whose elements are located consecutively on the outer face. Since the number of these subsets is quadratic in the number of terminals, a minimal Steiner tree can be computed in polynomial time using this recursion.

Let us return to our problem of determining α. We can clearly use the polynomial time algorithm described above to compute a minimal Steiner tree for every $J \subseteq T_D$ in G_D, because G_D is planar and the dual terminal set T_D (and thus J) lies on the outer face of G_D. We can also take the additional condition that every Steiner tree S for J has to satisfy $S \in \mathcal{D}$ into account by some slight modifications of the recursion formula. Moreover, by running the recursion appropriately we can simultanously determine the optimal subset J^* of T_D (and thus solve (3.5)) as follows.

First, from the minimum weight of a Steiner tree for J we subtract the number of its terminals. This can easily be taken into account in the recursion formula, since each terminal is a leaf of the Steiner tree (see properties of $\mathcal{D}$). Second, the minimum in (3.5) is taken over all subsets of T_D with at least two elements. The number of these subsets is exponential in the size of the terminals. However, it is possible to decide locally which dual terminal belongs to the optimal solution. Namely, a shortest path $P(v,d), v \in V_D \setminus T_D, d \in T_D$, is a branch of a minimal Steiner tree only if $y(P(v,d)) \leq 1$ holds. This is due to the fact that, if such a branch is added to a minimal Steiner tree, the left hand side of the corresponding Steiner partition inequality increases by the weight of the path, whereas the right hand side is incremented by one. Summing up we obtain the following recursion:

$$
(3.6)\quad
\begin{array}{lll}
\text{(i)} & y^v_{i,0} := \min\{y(W(v,d_i)) - 1,\ 0\}, & \text{for all } v \in V_D \setminus T_D, i = 1,\dots,z; \\
\text{(ii)} & \psi^v_{i,j} := \min\limits_{1 \leq l \leq j} (y^v_{i,l-1} + y^v_{i+l,j-l}), & \text{for all } v \in V_D \setminus T_D, \\
 & & i = 1,\dots,z,\ j = 1,\dots,z-1; \\
\text{(iii)} & y^v_{i,j} := \min\limits_{u \in V_D \setminus T_D} (y(W(v,u)) + \psi^u_{i,j}), & \text{for all } v \in V_D \setminus T_D, \\
 & & i = 1,\dots,z,\ j = 1,\dots,z-1;
\end{array}
$$

where $W(u,v)$, $u,v \in V_D$ denotes a shortest path in G_D from u to v such that $(T_D \cap V_D(W(u,v))) \setminus \{u,v\} = \emptyset$. This additional restriction is necessary to guarantee that the solution belongs to $\mathcal{D}$.

In the following we show that Recurision (3.6) works correctly. Unfortunately, the proof requires many technicalities that we do not see how to avoid.

For $i = 1,\dots,z$, $j = 0,\dots,z-1$, let $P_{i,j}$ denote the unique path from d_i to d_{i+j} by walking along the outer face of G_D in clockwise order. We define the interval $[d_i, d_{i+j}] := T_D \cap V_D(P_{i,j})$. Consider a Steiner tree S in G_D for some subset $J \subseteq [d_i, d_{i+j}]$, $J \neq \emptyset$. We denote by l_S the index of the "left most" dual terminal and by $l_S + r_S$ the index of the "right most" dual terminal of S, i. e., element of J; in formulas:

$$l_S := i + h^*, \text{ with } h^* := \min\{h \mid h \geq 0,\, d_{i+h} \in V_D(S)\},$$

and

$$r_S := \max\{h \mid h \leq j,\, d_{l_S+h} \in V_D(S)\}.$$

Moreover, for $i = 1, \ldots, z$, $j = 1, \ldots, z-1$, we introduce the symbol $e_{i,j}$ to denote the edge that is incident to d_i and d_{i+j}. Set $G_{i,j} := (V_D, E \cup \{e_{i,j}\})$. In the planar representation of $G_{i,j}$ we embed the edge $e_{i,j}$ in the outer face of G_D such that it is homotopic to the path $P_{i,j}$. Figure 2 illustrates this construction. It will turn out to be useful to employ the symbol $e_{i,0}$ in some recursion formula in order to avoid the treatment of additional special cases. We will interpret $e_{i,0}$ as a nonexisting edge and, accordingly, $G_{i,0}$ as the graph G_D.

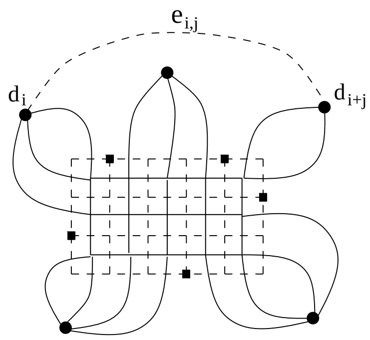

Figure 2:

Lemma 3.7 *For $i = 1, \ldots, z$, $j = 0, \ldots, z-1$ and $v \in V_D \setminus T_D$, define $\mathcal{D}^v_{i,j} := \{S \subseteq E \mid S$ is an edge-minimal Steiner tree for $J \cup \{v\}$ with $J \subseteq [d_i, d_{i+j}]$ such that $d_S(j) = 1$ for $j \in J$ and $d_S(j) = 0$ for $j \in T_D \setminus J$ and such that, if $J \neq \emptyset$, in addition $v \in V_D(O_{G_{l_S,r_S}}(S \cup \{e_{l_S,r_S}\}))\}$. Then, for the values that are computed using Recursion (3.6), the following property holds:*

$$y^v_{i,j} \leq \min_{\emptyset \subseteq J \subseteq [d_i, d_{i+j}]} \Big(\min_{\substack{S \text{ Steiner tree for } J\cup\{v\} \\ S \in \mathcal{D}^v_{i,j}}} y(S) \Big) - |J|,$$

for all $v \in V_D \setminus T_D$, $i = 1, \ldots, z, j = 0, \ldots, z-1$.

Proof.
For $r = 1, \ldots, z$, $s = 0, \ldots, z-1$ and $v \in V_D \setminus T_D$, let $D^v_{r,s} \subseteq [d_r, d_{r+s}]$ and $S^v_{r,s} \in \mathcal{D}^v_{r,s}$ be a Steiner tree for $D^v_{r,s} \cup \{v\}$ such that

$$y(S^v_{r,s}) - |D^v_{r,s}| = \min_{\emptyset \subseteq J \subseteq [d_r, d_{r+s}]} \Big(\min_{\substack{S \text{ Steiner tree for } J\cup\{v\} \\ S \in \mathcal{D}^v_{r,s}}} y(S) \Big) - |J|.$$

We must show that $y^v_{i,j} \leq y(S^v_{i,j}) - |D^v_{i,j}|$ for all $v \in V_D \setminus T_D$, $i = 1, \ldots, z$, $j = 0, \ldots, z-1$. We prove the statement by induction over j. For $j = 0$ Lemma 3.7 is obviously true. Suppose the statement also holds for all $k = 0, \ldots, j-1$. Let $v \in V_D \setminus T_D$ be any arbitrary node and $i \in \{1, \ldots, z\}$. For ease of notation let $l := l_{S^v_{i,j}}$, $r := r_{S^v_{i,j}}$ and $F := \mathrm{O}_{G_{l,r}}(S^v_{i,j} \cup \{e_{l,r}\})$. We distinguish two cases:

(1) $d_{S^v_{i,j}}(v) \geq 2$.

Since G_D is planar, since all dual terminals (and thus $D^v_{i,j}$) lie on the outer face of G_D, and since $v \in V_D(F)$, there exists an index $q \in \{1, \ldots, j\}$ and two nonempty disjoint subtrees $S_1, S_2 \subseteq S^v_{i,j}$ with $S_1 \cup S_2 = S^v_{i,j}$ such that S_1 is an edge-minimal Steiner tree for $(D^v_{i,j} \cap [d_i, d_{i+q-1}]) \cup \{v\}$ and S_2 is an edge-minimal Steiner tree for $(D^v_{i,j} \cap [d_{i+q}, d_{i+j}]) \cup \{v\}$ (see Figure 3).

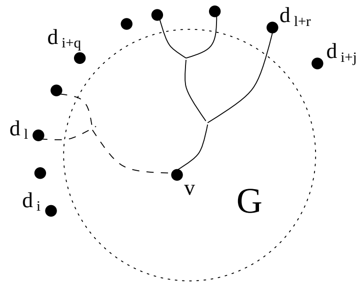

Figure 3:

Moreover, $S^v_{i,j} \in D^v_{i,j}$ implies that $d_{S_1}(d) = 1$ for all $d \in D^v_{i,j} \cap [d_i, d_{i+q-1}]$ and $d_{S_1}(d) = 0$ for all $d \in T_D \setminus (D^v_{i,j} \cap [d_i, d_{i+q-1}])$. The same holds for S_2, i. e., $d_{S_2}(d) = 1$ for all $d \in D^v_{i,j} \cap [d_{i+q}, d_{i+j}]$ and $d_{S_2}(d) = 0$ for all $d \in T_D \setminus (D^v_{i,j} \cap [d_{i+q}, d_{i+j}])$. Let $F_1 := \mathrm{O}_{G_{l_{S_1}, r_{S_1}}}(S_1 \cup \{e_{l_{S_1}, r_{S_1}}\})$ and $F_2 := \mathrm{O}_{G_{l_{S_2}, r_{S_2}}}(S_2 \cup \{e_{l_{S_2}, r_{S_2}}\})$. It is clear that $V_D(F) \subseteq V_D(F_1) \cup V_D(F_2)$ and $\{v\} = V_D(F_1) \cap V_D(F_2)$. Therefore, $S_1 \in \mathcal{D}^v_{i,q-1}$ and $S_2 \in \mathcal{D}^v_{i+q,j-q}$. This yields

$$\begin{aligned} y(S^v_{i,j}) - |D^v_{i,j}| &= y(S_1) - |D^v_{i,j} \cap [d_i, d_{i+q-1}]| + \\ &\quad\; y(S_2) - |D^v_{i,j} \cap [d_{i+q}, d_{i+j}]| \\ &\geq y(S^v_{i,q-1}) - |D^v_{i,q-1}| + y(S^v_{i+q,j-q}) - |D^v_{i+q,j-q}| \\ &\geq y^v_{i,q-1} + y^v_{i+q,j-q} \\ &\geq \psi^v_{i,j} \geq y^v_{i,j}. \end{aligned}$$

(2) $d_{S^v_{i,j}}(v) = 1$.

We consider three subcases:

(a) $|D^v_{i,j}| \geq 2$.

Then, since $S^v_{i,j}$ is edge-minimal, there exists a node $u \in V_D \setminus T_D$ with $d_{S^v_{i,j}}(u) \geq 2$ such that $S^v_{i,j} = W(v,u) \cup S'$, where $W(v,u) \cap S' = \emptyset$ (see Figure 4). Obviously, $l_{S'} = l$, $r_{S'} = r$ and, since $d_{S^v_{i,j}}(v) = 1$ and $v \in V_D(F)$ we have that $F = W(v,u) \cup \mathrm{O}_{G_{l,r}}(S' \cup \{e_{l,r}\})$. Moreover, it is easy to check that S' is

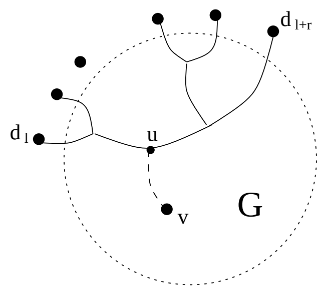

Figure 4:

an edge-minimal Steiner tree for $D^v_{i,j} \cup \{u\}$ satisfying all further properties in $\mathcal{D}^u_{i,j}$. This, together with (1) (note that $d_{S'}(u) \geq 2$) yields

$$\begin{aligned} y(S^v_{i,j}) - |D^v_{i,j}| &= y(W(v,u)) + y(S') - |D^v_{i,j}| \\ &\geq y(W(v,u)) + \psi^u_{i,j} \\ &\geq y^v_{i,j}. \end{aligned}$$

(b) $D^v_{i,j} = \{d_u\}$ for some $u \in \{i, \ldots, i+j-1\}$.
Then, we know that $S^v_{i,j} \in \mathcal{D}^v_{i,u-i}$, and we obtain

$$\begin{aligned} y(S^v_{i,j}) - |D^v_{i,j}| &= y(W(v,d_u)) - 1 \\ &\geq y(S^v_{i,u-i}) - |D^v_{i,u-i}| \\ &\geq y^v_{i,u-i} \geq y^v_{i,u-i} + y^v_{u+1,i+j-u-1} \\ &\geq \psi^v_{i,j} \geq y^v_{i,j}. \end{aligned}$$

(c) $D^v_{i,j} = \emptyset$.
Here, we have that $y(S^v_{i,j}) - |D^v_{i,j}| = 0 \geq \psi^v_{i,j} \geq y^v_{i,j}$.

This completes the proof. $\square$

Let $\beta := \min_{v \in V_D \setminus T_D} y^v_{1,z-1}$ and let S^* be the corresponding edge set. Obviously, $(V_D(S^*), S^*)$ is connected and Lemma 3.7 implies $\beta \leq \alpha$. If $\beta \geq -1$, then there does not exist a violated Steiner partiton inequality. If $\beta < -1$, we get $p^* := |V_D(S^*) \cap T_D| \geq 2$, since $y \geq 0$ holds. Thus, $S^* \in \mathcal{D}$ and $\alpha = \beta$. Due to Lemma 3.3 there exists a Steiner partiton $V_1, \ldots, V_{p^*}$ with $\delta(V_1, \ldots, V_{p^*}) = S^*$ and $0 > \beta + 1 = y(\delta(V_1, \ldots, V_{p^*})) - p^* + 1$. Therefore, $V_1, \ldots, V_{p^*}$ defines a violated Steiner partiton inequality.
This gives rise to the following algorithm.

Algorithm 3.8 (Separation algorithm for the Steiner partition inequalities)

Input:

A planar graph $G = (V, E)$, a set of terminals $T \subseteq V$ that are located on the outer face and a vector $y \in \mathbb{R}^E$, $y \geq 0$.

Output:

One of the following possibilities:

- a violated Steiner partiton inequality,

- the message "there does not exist a violated Steiner partiton inequality".

(1) Construct the graph $G_D = (V_D, E)$ with $T_D = \{d_1, \dots, d_z\}$.

(2) Compute shortest paths $W(u, v)$ for all $u, v \in V_D$ such that no inner node of the corresponding paths is an element of T_D.

(3) Determine $y^v_{1,z-1}$ for all $v \in V_D \setminus T_D$ using Recursion (3.6).

(4) Set $\beta := \min_{v \in V_D \setminus T_D} y^v_{1,z-1}$.

(5) If $\beta \geq -1$, print the message "there does not exist a violated Steiner partiton inequality", STOP.

(6) Determine the edge set S^* corresponding to β.

(7) Return the violated inequality $(\chi^{S^*})^T x \geq |V_D(S^*) \cap T_D| - 1$.

(8) STOP.

The running time for the execution of steps (3) and (4) of Algorithm 3.8 is bounded by $O(|V_D|^2|T|^2)$. Also note that (3.2) (ii) can easily be taken into account in step (i) of Recursion (3.6). Taking all together, we obtain the following theorem.

Theorem 3.9 *Let $G = (V, E)$ be a planar graph and let $T \subseteq V$ be a set of terminals located on the outer face of G. Then, the separation problem for the Steiner partition inequalities can be solved in time $O(|V_D|^2|T|^2 + \gamma)$, where γ is the running time for the computation of the shortest paths between all pairs of nodes.*

Let us close this section by two remarks.

From Lemma 3.3 we know that each Steiner tree in G_D for some subset J of T_D corresponds to a Steiner partition inequality. This observation gives rise to several heuristic algorithms for finding violated Steiner partition inequalities. Namely, instead of calculating an optimal Steiner tree in G_D, we determine a Steiner tree also heuristically. Many heuristics are known for the solution of the minimum Steiner tree problem (see, for instance, [HRW92] for a survey). We have implemented one such algorithm that is based on the ideas described in [TM80]. This heuristic starts with a terminal $d \in T_D$. Then, a terminal $d' \in T_D \setminus \{d\}$ is chosen such that the weight of a shortest path from d' to d is minimal. Finally, d' and d are connected via a shortest path. This scheme is iterated until all terminals are connected. For our purposes this procedure is slightly modified. First, we have to make sure that no inner node on the corresponding shortest paths is an element of T_D. Second, in order to generate as many inequalites as possible, we compute a Steiner tree starting with all pairs of nodes d_i, d_j, where $d_i, d_j \in T_D$. The advantage of this heuristic is that not only the final Steiner trees define Steiner partition inequalities, but also any of its iteratively computed subtrees defines a Steiner partition inequality (cf. Lemma 3.3). Working in this scheme we obtain plenty of inequalities. For each of them we check whether it is violated. We will see in the last section that this heuristic works very well for our problem instances.

Finally, let us point out that Algorithm 3.8 can also be used to solve certain multicut problems. Suppose, there is given a planar graph G, a set of nodes $T \subseteq V$ located on the outer face of G and nonnegative edge weights $w_e, e \in E$, and we want to determine $\min\{-\lambda, \min\{w(\delta(V_1, \ldots, V_p)) - \lambda p \mid V_1, \ldots, V_p, p \geq 2$ is a Steiner partition of V with respect to T such that $G(V_1, \ldots, V_p)$ is 2-node connected$\}\}$, where λ is the gain for each element of the partition. By applying some modifications to Algorithm 3.8 this problem can be solved in polynomial time as well.

4 Separation of the Alternating Cycle Inequalities and Extensions

We first introduce the so-called alternating cycle inequalities. Let $G = (V, E)$ be a graph and $\mathcal{N} = \{T_1, T_2\}$ a net list. We call a cycle F in G an *alternating cycle with respect to* T_1, T_2, if $F \subseteq [T_1 : T_2]$ and $V(F) \cap T_1 \cap T_2 = \emptyset$ (see Figure 5). Moreover, let $F_1 \subseteq E(T_2)$ and $F_2 \subseteq E(T_1)$ be two sets of diagonals of the alternating cycle F with respect to T_1, T_2. The inequality

$$(\chi^{E\setminus(F\cup F_1)}, \chi^{E\setminus(F\cup F_2)})^T x \geq \tfrac{1}{2}|F| - 1$$

is called an *alternating cycle inequality.*

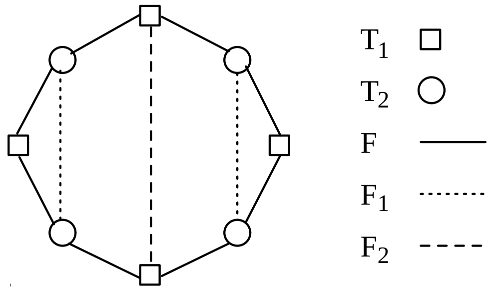

Figure 5:

It is not difficult to see that the basic form of an alternating cycle inequality, i. e., $F_1 = F_2 = \emptyset$, is valid for STP $(G, \mathcal{N}, \mathbb{1})$, but in general, it is not facet-defining. The sets F_1 and F_2 are used to strengthen the basic form; in fact, choosing them appropriately we can obtain facet-defining inequalities.

The sets of diagonals $F_1 \subseteq E(T_2)$ and $F_2 \subseteq E(T_1)$ are called *maximal cross free with respect to* F, if F_1 and F_2 are cross free, and each diagonal $e_1 \in E(T_1) \setminus F_2$

crosses F_1 and each diagonal $e_2 \in E(T_2) \setminus F_1$ crosses F_2 (see Figure 5). Then, the following theorem holds.

Theorem 4.1 *Let $G = (V, E)$ be a graph that contains the complete graph on node set V as a subgraph and let $\mathcal{N} = \{T_1, T_2\}$ be a disjoint net list with $T_1 \cup T_2 = V$ and $|T_1| = |T_2| = l$, $l \geq 2$. Furthermore let F be an alternating cycle with respect to T_1, T_2 with $V(F) = V$ and $F_1 \subseteq E(T_2)$, $F_2 \subseteq E(T_1)$. Then, the alternating cycle inequality*

$$(\chi^{E\setminus(F\cup F_1)}, \chi^{E\setminus(F\cup F_2)})^T x \geq l - 1$$

defines a facet of $STP(G, \mathcal{N}, \mathbb{1})$ if and only if F_1 and F_2 are maximal cross free.

There is a natural way to extend the alternating cycle inequalities as follows. Let $G = (V, E)$ be a graph and $\mathcal{N} = \{T_1, T_2\}$ be a net list. Let $V_1, \ldots, V_k$ be a partition of V with $k \geq 4$ and k even such that the following properties are satisfied:

$$(4.2) \qquad \begin{array}{ll} (i) & (V_i, E(V_i)) \text{ is connected for } i = 1, \ldots, k, \\ (ii) & V_{2i+1} \cap T_1 \neq \emptyset,\ V_{2i+1} \cap T_2 = \emptyset, \text{ for } i = 0, \ldots, \frac{k}{2} - 1, \\ & V_{2i} \cap T_1 = \emptyset,\ V_{2i} \cap T_2 \neq \emptyset, \text{ for } i = 1, \ldots, \frac{k}{2}, \\ (iii) & [V_i : V_{i+1}] \neq \emptyset, \text{ for } i = 1, \ldots, k. \end{array}$$

(An index $i > k$ is identified with the index $((i-1)$ modulo $k)+1$.) Condition (iii) guarantees that the contracted graph $G(V_1, \ldots, V_k)$ (i. e. the graph obtained by contracting every element of the partition to a single node) contains at least one hamiltonian cycle. We choose an edge set $F \subseteq \cup_{i=1}^{k} [V_i : V_{i+1}]$ in G that forms a hamiltonian cycle in $G(V_1, \ldots, V_k)$. Note that, due to (ii), F is alternating. Furthermore, let $F_1 \subseteq \bigcup_{\substack{i,j \text{ even} \\ i \neq j}} [V_i : V_j]$, $F_2 \subseteq \bigcup_{\substack{i,j \text{ odd} \\ i \neq j}} [V_i : V_j]$ be two edge sets such that F_1 and F_2, viewed as edge sets in the contracted graph $G(V_1, \ldots, V_k)$, are cross free with respect to the alternating cycle F. Then we call the following inequality

$$(\chi^{E\setminus(F\cup F_1)}, \chi^{E\setminus(F\cup F_2)})^T x \geq \frac{k}{2} - 1$$

extended alternating cycle inequality. This inequality is valid with respect to STP $(G, \mathcal{N}, \mathbb{1})$ due to Lemma 2.2. Let us give an example.

Example 4.3 *Consider the graph G in Figure 6 (a) with $T_1 = \{1, 3, 5, 10\}$ and $T_2 = \{4, 9, 12\}$. It can easily be checked that the partition V_1, V_2, V_3, V_4 satisfies (4.2); the corresponding contracted graph $G(V_1, V_2, V_3, V_4)$ is depicted in Figure 6 (a). Obviously, $F := \{\{3, 4\}, \{10, 11\}, \{9, 10\}, \{5, 9\}\}$ is an hamiltonian alternating cycle in $G(V_1, V_2, V_3, V_4)$ and $F_1 := \emptyset$, $F_2 := \{\{2, 6\}, \{5, 6\}\}$ are cross free sets of diagonals. Thus, the inequality $x^1_{26} + x^1_{56} + x^1_{37} + x^1_{67} + x^2_{37} + x^2_{67} \geq 1$ is an extended alternating cycle inequality.*

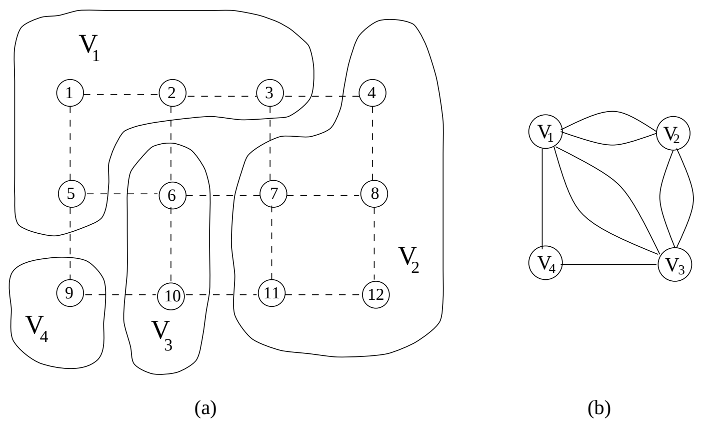

Figure 6:

Observe that, when $V_1, \ldots, V_k$ are chosen we have the freedom to pick F, F_1 and F_2 among many possible alternatives. We call any tripel (F, F_1, F_2) that satisfies the additional requirements defined above a *feasible tripel (for $V_1, \ldots, V_k$)*.

We do not know under which conditions the extended alternating cycle inequalities define facets of the Steiner tree packing polyhedron and we do not know how to separate these inequalities in the general case. Our aim here is to outline a separation routine for extended alternating cycle inequalities in the (practically relevant) case where a planar graph G is given and all terminals of T_1 and T_2 are on the outer face.

We proceed in a similar way as for the Steiner partition inequalities. We show that, for each partition $V_1, \ldots, V_k$ satisfying (4.2), the multicut $\delta(V_1, \ldots, V_k)$ corresponds to a certain Steiner tree in a graph that remains to be defined. Here, an additional difficulty comes up, since the edges of $\delta(V_1, \ldots, V_k)$ must be evaluated differently. The coefficients depend on the choice of the alternating cycle F in $G(V_1, \ldots, V_k)$ and on the sets F_1 and F_2. Thus, for the corresponding Steiner tree, a vector must be defined that "sifts" the edges that correspond to F, F_1 or F_2, respectively.

Without loss of generality we suppose the planar graph G to be 2-node connected so that the edge set $\mathrm{O}_G(E)$ that encloses the outer face is a cycle. Let $T = T_1 \cup T_2$ and we may assume that $T := \{t_1, \ldots, t_z\}$ is numbered in a clockwise fashion along this cycle. Let us consider the dual graph $G^* = (V^*, E)$ of G. We split the node representing the outer face into z nodes $d_1, \ldots, d_z$ such that every edge of $\mathrm{O}_G(E)$ that is passed by walking from t_i to t_{i+1} on $\mathrm{O}_G(E)$ in clockwise order is incident to d_{i+1} for $i = 1, \ldots, z$. Let $G_D = (V_D, E)$ denote the resulting graph and set $T_D = \{d_1, \ldots, d_z\}$. Figure 7 illustrates this construction for the graph of

Example 4.3. Set

$$\begin{array}{rl} M := & \{d_i \in T_D \mid \{t_{i-1}, t_i\} \cap T_k \neq \emptyset \text{ for } k = 1, 2\}, \\ \mathcal{S} := & \{S \subseteq E \mid S \text{ is an edge-minimal Steiner tree for } M \text{ in } G_D \text{ such that} \\ & \qquad d_S(d) = 1 \text{ for all } d \in M \text{ and } d_S(d) = 0 \text{ for all } d \in T_D \setminus M\}. \end{array}$$

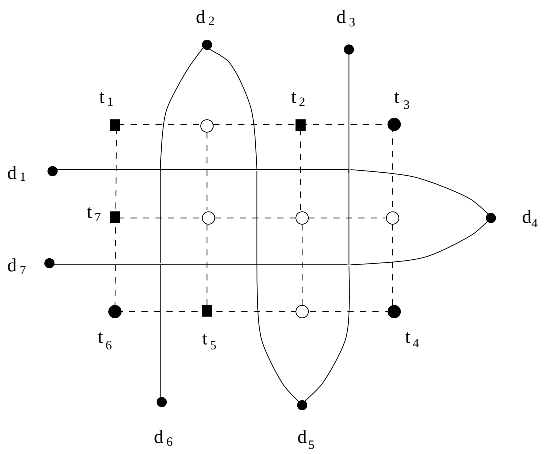

Figure 7:

Then, the following relation holds.

Lemma 4.4 *Let $G = (V, E)$ be a planar graph and $\mathcal{N} = \{T_1, T_2\}$ where all terminals are located on the outer face. Then, the following statements are true:*

1. *If $V_1, \ldots, V_k$ is a partition of V satisfying (4.2), then the corresponding multicut $\delta(V_1, \ldots, V_k)$ viewed as an edge set of the dual graph G_D is a Steiner tree contained in $\mathcal{S}$.*

2. *If S is a Steiner tree in G_D contained in $\mathcal{S}$ and $|M| \geq 4$ holds, then there exists a partition $V_1, \ldots, V_k$ of V satisfying (4.2) such that $S = \delta(V_1, \ldots, V_k)$.*

Proof. Let us prove the first statement. Suppose $V_1, \ldots, V_k$ with $k \geq 4$ even is a partition satisfying (4.2). Since G is planar and all terminals are located on the outer face, properties (4.2) guarantee that the numbering of the partition is clockwise (or anticlockwise) along the outer face of G. Without loss of generality we suppose that the elements of the partition are numbered in clockwise fashion. For every $i \in \{1, \ldots, k\}$, the graph $(V_i, E(V_i))$ is connected and $V_i \cap T \neq \emptyset$. Hence, there exist $d_{i_1}, d_{i_2} \in T_D$ such that $\delta(V_i)$ defines a path from d_{i_1} to d_{i_2}. Without loss of generality d_{i_1}, d_{i_2} are chosen such that terminal $t_{i_1} \in V_i$. From property (iii) of (4.2) and the fact that $V_1, \ldots, V_k$ is a partition it follows that $d_{i_2} = d_{(i+1)_1}$ for $i = 1, \ldots, k$. Thus, $S := \cup_{i=1}^k \delta(V_i) = \delta(V_1, \ldots, V_k)$ is a Steiner tree for $M' := \cup_{i=1}^k \{d_{i_1}\}$. Since $V_1, \ldots, V_k$ is a partition and since (4.2) (ii), (iii) hold, S is edge-minimal. Properties (4.2) (ii) and (iii) together with k even imply that $M' = M$ and thus $d_S(d) = 0$ for all $d \in T_D \setminus M$. Finally, $d_S(d) = 1$ for all $d \in M'$, because $V_1, \ldots, V_k$ is a partition of V with $V_i \cap T \neq \emptyset\,(i = 1, \ldots, k)$ and property (4.2) (iii) holds. Thus, $S \in \mathcal{S}$.

Conversely, let $S \in \mathcal{S}$ and suppose $|M| \geq 4$. We number the elements in $M = \{d_{s_1}, \ldots, d_{s_m}\}$ with $m := |M|$ in clockwise order around the outer face such that $t_{s_1 - 1} \in T_1$. Every unique path P_i in S from d_{s_i} to $d_{s_{i+1}}$ is a cut in G, i. e., there exists a node set V_i such that $\delta(V_i) = P_i$. Moreover, we can assume that $t_{s_i - 1} \in V_i$, for $i = 1, \ldots, m$. Since $d_S(d) = 0$ for all $d \in T_D \setminus M$, $(V_i, E(V_i))$ is connected for all $i = 1, \ldots, m$. Furthermore, $V_1, \ldots, V_m$ is a partition of V, because S is edge-minimal. Property (iii) of (4.2) follows from $d_S(d) = 1$ for all $d \in M$. By construction, $V_1 \cap T_1 \neq \emptyset$. So, we obtain Property (4.2) (ii) from the fact that S is a Steiner tree for M with $V_D(S) \cap (T_D \setminus M) = \emptyset$. Obviously, $m \geq 4$ and even, since $|M| \geq 4$ and even. $\square$

In Figure 8 the Steiner tree $S \in \mathcal{S}$ which corresponds to the partition V_1, V_2, V_3, V_4 shown in Figure 6 (a) is depicted in thick solid lines. From the proof of Lemma 4.4 we see that the cardinality k of a partition $V_1, \ldots, V_k$ satisfying (4.2) equals $|M|$. So, we suppose from now on that $k = |M| \geq 4$, otherwise, there does not exist any extended alternating cycle inequality.

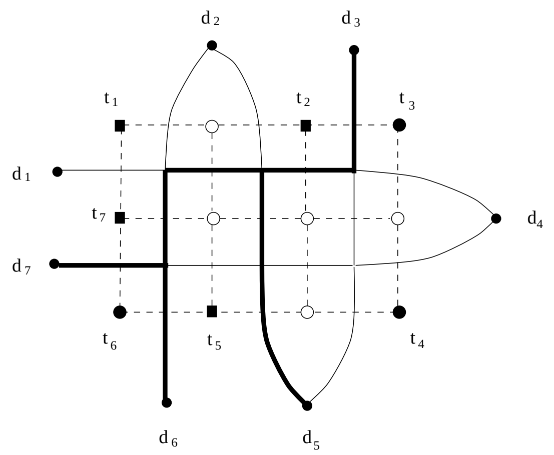

Figure 8:

Next, we define a "sifting function" for each $S \in \mathcal{S}$. Let $V_1^S, \ldots, V_k^S$ be the corresponding partition satisfying (4.2) according to Lemma 4.4. We call a vector $a \in \{0,1\}^{\{T_1, T_2\} \times E}$ a *sifting for* S if there exists a feasible tripel (F, F_1, F_2) for $V_1^S, \ldots, V_k^S$ such that $a = (\chi^{E \setminus (F \cup F_1)}, \chi^{E \setminus (F \cup F_2)})$. Moreover, set $\eta(S) := \frac{k}{2}$ and let $\mathcal{F}(S)$ denote the set of all siftings for S. Consider now the following minimization problem

$$(4.5) \qquad \min_{S\in\mathcal{S}} \min_{a\in\mathcal{F}(S)} y^1(S\cap I_{a^1}) + y^2(S\cap I_{a^2}) - \eta(S),$$

where $y \in \mathbb{R}^{\{T_1,T_2\}\times E}, y \geq 0$, is the vector to be cut off and $I_b = \{e \in E \mid b_e = 1\}$ for $b \in \{0,1\}^E$. Let μ denote the minimum value of (4.5). From Lemma 4.4 and the definition of a sifting we know that we can solve the separation problem for the extended alternating cycle inequalities via computing (4.5). If $\mu \geq -1$, there obviously exists no violated inequality. If $\mu < -1$, let $\tilde{S} \in \mathcal{S}$ and $\tilde{a} \in \mathcal{F}(S)$ be such that $\mu = y^1(\tilde{S}\cap I_{\tilde{a}^1}) + y^2(\tilde{S}\cap I_{\tilde{a}^2}) - \eta(\tilde{S})$. In this case $(\chi^{\tilde{S}\cap I_{\tilde{a}^1}}, \chi^{\tilde{S}\cap I_{\tilde{a}^2}}) \geq \eta(\tilde{S}) - 1$ is an extended alternating cycle inequality that is violated by y. Thus, it remains to solve Problem (4.5).

As in the case of the Steiner partition inequalities we develop a dynamic program in order to compute the optimal Steiner tree $\tilde{S} \in \mathcal{S}$ and the optimal sifting $\tilde{a} \in \mathcal{F}(\tilde{S})$. Consider the following recursion:

Recursion 4.6 *Let $y^k(u,v)$ denote the value of a shortest path from u to v with respect to the weighting y^k whose inner nodes have empty intersection with T_D. Moreover, let $y(u,v)$ correspond to the value of a shortest path from u to v with respect to the weight function y^1+y^2 whose inner nodes have empty intersection with T_D. Finally, define $y_{-1}(u,v) := \min\{\hat{y}(W) - \max_{e\in W}\hat{y}_e \mid W$ is a path from u to v whose inner nodes have empty intersection with $T_D\}$ where $\hat{y} = y^1+y^2$. Then, for all $i = 1,\ldots,z$, $j = 0,\ldots,z-1$, $v \in V_D \setminus T_D$ and $k = 1,2$ (with $\bar{k} = 1$, if $k = 2$, and $\bar{k} = 2$, if $k = 1$) set*

$$\begin{array}{lll}
(1) & l_1(v,i,0) := y_{-1}(v,d_i) - \frac{1}{2}, & \text{if } d_i \in M; \\
 & l_1(v,i,0) := 0, & \text{if } d_i \in T_D \setminus M; \\
(2) & l_2(v,i,j) := \min\limits_{1\leq r\leq j} l_1(v,i,r-1) + l_1(v,i+r,j-r); & \\
(3) & l_1(v,i,j) := \min\limits_{u\in V_D\setminus T_D} y^k(v,u) + l_2(u,i,j), & \text{if } t_{i-1}, t_{i+j} \in T_k; \\
 & l_1(v,i,j) := \min\limits_{u\in V_D\setminus T_D} y(v,u) + l_2(u,i,j), & \text{if } t_{i-1} \in T_k,\ t_{i+j} \in T_{\bar{k}} \\
 & & \text{or } t_{i-1} \in T_{\bar{k}},\ t_{i+j} \in T_k.
\end{array}$$

Then, the following theorem holds.

Theorem 4.7 *For the value $l_{min} := \min_{v\in V_D\setminus T_D} l_1(v,1,z-1)$ computed via Recursion 4.6, we have*

$$l_{min} \leq \min_{S\in\mathcal{S}} \min_{a\in\mathcal{F}(S)} y^1(S\cap I_{a^1}) + y^2(S\cap I_{a^2}) - \eta(S).$$

Proof.

We will proceed as in the last chapter and prove a relation between solutions for subproblems in the spirit of Lemma 3.7. Hereto we need some further notation.

First of all, we have to define subtrees of Steiner trees in $\mathcal{S}$. For all $i = 1, \ldots, z$, $j = 0, \ldots, z-1$, we use the same notation and definitions as in the previos section, i. e., the interval $[d_i, d_{i+j}]$, the edge $e_{i,j}$ embedded in the outer face of G_D, the graph $G_{i,j} := (V_D, E \cup \{e_{i,j}\})$, and, for a Steiner tree in G_D for $J \subseteq [d_i, d_{i+j}]$, $J \neq \emptyset$, the symbols l_S and r_S. For $v \in V_D \setminus T_D$ and $i = 1, \ldots, z$, $j = 0, \ldots, z-1$, we define $\mathcal{S}^v_{i,j} := \{S \subseteq E \mid S$ is an edge-minimal Steiner tree in G_D for $([d_i, d_{i+j}] \cap M) \cup \{v\}$ such that $d_S(d) = 1$ for all $d \in [d_i, d_{i+j}] \cap M$ and $d_S(d) = 0$ for all $d \in T_D \setminus ([d_i, d_{i+j}] \cap M)$ and such that, if $[d_i, d_{i+j}] \cap M \neq \emptyset$, in addition $v \in V_D(\mathrm{O}_{G_{l_S, r_S}}(S \cup \{e_{l_s, r_S}\}))\}$. Furthermore, we call a vector $a \in \{0,1\}^{\{T_1, T_2\} \times E}$ a sifting for $S \in \mathcal{S}^v_{i,j}$ if there exists a Steiner tree $\tilde{S} \in \mathcal{S}$ and a sifting $\tilde{a}$ for $\tilde{S}$ such that $S \subseteq \tilde{S}$, $a_e = \tilde{a}_e$ for all $e \in S$ and $a_e = 0$, otherwise. Let $\mathcal{F}(S)$ denote the set of siftings for $S \in \mathcal{S}^v_{i,j}$ and set $\eta(S) := \frac{1}{2}|[d_i, d_{i+j}] \cap M|$.

What we want to show is that $l_1(v, i, j)$ is a lower bound for $y^1(S \cap I_{a^1}) + y^2(S \cap I_{a^2}) - \eta(S)$ for all Steiner trees $S \in \mathcal{S}^v_{i,j}$ and all siftings $a \in \mathcal{F}(S)$. Unfortunately, this is not true, in general. It turns out that depending on the node v certain siftings must be excluded.

Let $S \in \mathcal{S}^v_{i,j}$. Define $V^3_S := \{w \in V_D(S) \mid d_S(w) \geq 3\}$. Let $L \subseteq M \cap [d_i, d_{i+j}]$ be the set of nodes d such that, for the unique path P in S from v to d, $V_D(P) \cap V^3_S = \emptyset$ holds. If $L = \emptyset$, we set $\mathcal{F}_v(S) := \mathcal{F}(S)$. Otherwise, let P_d, $d \in L$, denote the unique path from v to d. We set $\mathcal{F}_v(S) := \{a \in \mathcal{F}(S) \mid$ for all $d \in L$ there exists an edge $e \in S \cap P_d$ with $a^1_e = a^2_e = 0\}$, that means that we allow only siftings where the corresponding alternating cycle has nonempty intersection with all paths P_d, $d \in L$.

Now, we show that

$$(*) \qquad l_1(v, i, j) \leq \min_{S \in \mathcal{S}^v_{i,j}} \min_{a \in \mathcal{F}_v(S)} y^1(S \cap I_{a^1}) + y^2(S \cap I_{a^2}) - \eta(S)$$

for all $v \in V_D \setminus T_D$, $i = 1, \ldots, z$, $j = 0, \ldots, z-1$.

We prove this by induction on j. $(*)$ is obviously true for $j = 0$. Now, suppose $(*)$ is also true for all $l = 0, \ldots, j-1$. Consider any arbitrary $v \in V_D \setminus T_D$ and $i \in \{1, \ldots, z\}$. Let $\tilde{S} \in \mathcal{S}^v_{i,j}$ and $\tilde{a} \in \mathcal{F}_v(\tilde{S})$ such that

$$y^1(\tilde{S} \cap I_{\tilde{a}^1}) + y^2(\tilde{S} \cap I_{\tilde{a}^2}) - \eta(\tilde{S}) = \min_{S \in \mathcal{S}^v_{i,j}} \min_{a \in \mathcal{F}_v(S)} y^1(S \cap I_{a^1}) + y^2(S \cap I_{a^2}) - \eta(S).$$

We have to show that $l_1(v, i, j) \leq y^1(\tilde{S} \cap I_{\tilde{a}^1}) + y^2(\tilde{S} \cap I_{\tilde{a}^2}) - \eta(\tilde{S})$. We distinguish two cases:

(1) $d_{\tilde{S}}(v) \geq 2$. Since G_D is planar, all terminals of T_D are located on the outer face of G_D and $v \in V_D(\mathrm{O}_{G_{l_{\tilde{S}}, r_{\tilde{S}}}}(\tilde{S} \cup \{e_{l_{\tilde{S}}, r_{\tilde{S}}}\}))$, there exists an index $r \in \{1, \ldots, j\}$ and two disjoint subtrees S_1, S_2 of $\tilde{S}$ such that $S_1 \cup S_2 = \tilde{S}$, $v \in V_D(S_1)$, $v \in V_D(S_2)$ and such that $S_1 \in \mathcal{S}^v_{i,r-1}$ and $S_2 \in \mathcal{S}^v_{i+r, j-r}$ (see also proof of Lemma 3.7). Set

$a_e^k := \tilde{a}_e^k$ for all $e \in E \setminus S_2$ and $a_e^k := 0$ for all $e \in S_2$, $k = 1,2$. Furthermore, choose $b_e^k := \tilde{a}_e^k$ for all $e \in E \setminus S_1$ and $b_e^k := 0$ for all $e \in S_1$, $k = 1,2$. Next, we show that $a \in \mathcal{F}_v(S_1)$ and $b \in \mathcal{F}_v(S_2)$. First, since $\tilde{a} \in \mathcal{F}(\tilde{S})$, we know that $a \in \mathcal{F}(S_1)$ and $b \in \mathcal{F}(S_2)$. Let $\tilde{L} \subseteq M \cap [d_i, d_{i+j}]$ denote the set of nodes d such that for the unique path P from v to d holds $V_D(P) \cap V_{\tilde{S}}^3 = \emptyset$. Denote by L_1 and L_2 the corresponding node sets of S_1 and S_2. From the fact that $\tilde{L} = L_1 \cup L_2$ we conclude that $a \in \mathcal{F}_v(S_1)$ and $b \in \mathcal{F}_v(S_2)$. Finally, note that $\eta(\tilde{S}) = \eta(S_1) + \eta(S_2)$. Summing up, we obtain that

$$\begin{aligned} y^1(\tilde{S} \cap I_{\tilde{a}^1}) + y^2(\tilde{S} \cap I_{\tilde{a}^2}) - \eta(\tilde{S}) &= y^1(S_1 \cap I_{a^1}) + y^2(S_1 \cap I_{a^2}) - \eta(S_1) + \\ &\quad\; y^1(S_2 \cap I_{b^1}) + y^2(S_2 \cap I_{b^2}) - \eta(S_2) \\ &\geq l_1(v,i,r-1) + l_1(v,i+r,j-r) \\ &\geq l_2(v,i,j) \geq l_1(v,i,j). \end{aligned}$$

(2) $d_{\tilde{S}}(v) = 1$. If $|V_D(\tilde{S}) \cap [d_i, d_{i+j}]| = 1$, we conclude that there exists an $r \in \{1, \ldots, j\}$ such that $\tilde{S} \in \mathcal{S}_{i,r-1}^v$ and $\emptyset \in \mathcal{S}_{i+r,j-r}^v$, or vice versa, $\emptyset \in \mathcal{S}_{i,r-1}^v$ and $\tilde{S} \in \mathcal{S}_{i+r,j-r}^v$ (note that $j > 0$). Since both $r-1$ and $j-r$ are less than or equal to $j-1$, we conclude by the assumption of the induction that $(*)$ holds. Now, suppose $|V_D(\tilde{S}) \cap [d_i, d_{i+j}]| \geq 2$. This implies that there exists a node $u \in V_{\tilde{S}}^3$ such that $\tilde{S} = W(v,u) \cup S'$, $W(v,u) \cap S' = \emptyset$, with $d_{S'}(u) \geq 2$ and $S' \in \mathcal{S}_{i,j}^u$, where $W(v,u)$ is the unique path in $\tilde{S}$ from v to u (for a more detailed discussion, see the corresponding case in the proof of Lemma 3.7). Set $a_e^k := \tilde{a}_e^k$ for all $e \in E \setminus W(v,u)$ and $a_e^k := 0$ for all $e \in W(v,u)$, $k = 1,2$. Obviously, $\eta(\tilde{S}) = \eta(S')$. Since $\tilde{a} \in \mathcal{F}_v(\tilde{S})$ we obtain that $a \in \mathcal{F}_u(S')$. Moreover, $\tilde{a} \in \mathcal{F}_v(\tilde{S})$ and $v \in V_D(\mathrm{O}_{G_{l_{\tilde{S}}, r_{\tilde{S}}}}(\tilde{S} \cup \{e_{l_{\tilde{S}}, r_{\tilde{S}}}\}))$ imply that, for all $e \in W(v,u)$, $\tilde{a}_e^k = 1$ if $t_{i-1} \in T_k$ or $t_{i+j} \in T_k$ for $k = 1,2$. Thus, taking the correctness of case (1) into account we get the following:
If $t_{i-1}, t_{i+j} \in T_k$ for some $k \in \{1,2\}$, we obtain that

$$\begin{aligned} y^1(\tilde{S} \cap I_{\tilde{a}^1}) + y^2(\tilde{S} \cap I_{\tilde{a}^2}) - \eta(\tilde{S}) &\geq y^k(W(v,u)) + \\ &\quad\; y^1(S' \cap I_{a^1}) + y^2(S' \cap I_{a^2}) - \eta(S') \\ &\geq y^k(W(v,u)) + l_2(u,i,j) \\ &\geq l_1(v,i,j). \end{aligned}$$

If $t_{i-1} \in T_1$ and $t_{i+j} \in T_2$ or $t_{i-1} \in T_2$ and $t_{i+j} \in T_1$, we have that

$$\begin{aligned} y^1(\tilde{S} \cap I_{\tilde{a}^1}) + y^2(\tilde{S} \cap I_{\tilde{a}^2}) - \eta(\tilde{S}) &\geq \hat{y}(W(v,u)) + \\ &\quad\; y^1(S' \cap I_{a^1}) + y^2(S' \cap I_{a^2}) - \eta(S') \\ &\geq \hat{y}(W(v,u)) + l_2(u,i,j) \\ &\geq l_1(v,i,j). \end{aligned}$$

We conclude that $(*)$ is true. Relation $(*)$ finally implies that

$$\begin{aligned} l_{min} &= \min_{v \in V_D \setminus T_D} l_1(v, 1, z-1) \\ &\leq \min_{v \in V_D \setminus T_D} \min_{S \in \mathcal{S}^v_{1,z-1}} \min_{a \in \mathcal{F}_v(S)} y^1(S \cap I_{a^1}) + y^2(S \cap I_{a^2}) - \eta(S) \\ &\leq \min_{S \in \mathcal{S}} \min_{a \in \mathcal{F}(S)} y^1(S \cap I_{a^1}) + y^2(S \cap I_{a^2}) - \eta(S). \end{aligned}$$

This completes the proof. □

An edge set S_l that yields the minimum value l_{min} can easily be obtained by labeling the corresponding edges in Recursion 4.6. Per construction, S_l is an element of $\mathcal{S}$. Note that the recursion formulas 4.6 implicitly define a vector $a_l \in \{0,1\}^{\{T_1,T_2\} \times E}$ such that $l_{min} = y^1(S_l \cap I_{a^1_l}) + y^2(S_l \cap I_{a^2_l}) - \eta(S_l)$. If $l_{min} \geq -1$, we conclude from Theorem 4.7 that there does not exist a violated alternating cycle inequality. If $l_{min} < -1$, unfortunately, a_l is not necessarily a sifting of S_l.

Example 4.8 *Consider the example depicted in Figure 9. Given a complete rectangular* 3×3 *grid graph, where the terminals of net* 1 *are printed as small black rectangles and these of net* 2 *as small black cycles. All other nodes are depicted as white cycles. The solid lines represent edges* e *with value* $y^1_e = 1$, *dashed lines edges* e *having value* $y^2_e = 0.5$ *and dotted lines edges* e *with value* $y^1_e = 0.5$. *The edge set in* G_D *yielding* l_{min} *is drawn in thick black lines.* l_{min} *results from the following computation:* $l_{min} = l_1(4,4,0) + l_1(4,3,0) + l_1(4,1,1) = (y_{-1}(4,d_4) - \frac{1}{2}) + (y_{-1}(4,d_3) - \frac{1}{2}) + (y^1(4,2) + l_1(2,1,0) + l_1(2,2,0)) = (0.0 - \frac{1}{2}) + (0.0 - \frac{1}{2}) + (0.5 + (0.0 - \frac{1}{2}) + (0.0 - \frac{1}{2})) = -1.5$. *The branching nodes of the recursion are the nodes* 4 *and* 2, *i.e., edge* $\{1,2\}$ *is counted twice. The branching nodes of the edge set* S_l *are the nodes* 4 *and* 1. *Therefore,* a_l *does not define a sifting for* S_l. *However,* a_l *can be modified to a sifting* $\bar{a}_l$ *for* S_l. *In this case we obtain* $y^1(S_l \cap F_{\bar{a}^1_l}) + y^2(S_l \cap F_{\bar{a}^2_l}) - \alpha(S_l) = -1.0$ *implying that the corresponding cycle inequality is not violated.*

We do not know how to avoid these cases. We have no alternative but check whether a_l actually is an element of $\mathcal{F}(S_l)$. Thus, the algorithm proposed is not an exact separation method. However, it provides a lower bound for the slack of the most violated extended alternating cycle inequality and, if $l_{min} \geq -1$, a proof that no extended alternating cycle inequality exists that is violated by y.

Clearly, the recursion itself for computing l_{min} takes time $O(|V_D|^2(|T_1| + |T_2|)^2)$. The values $y(u,v)$ and $y^k(u,v)$, $k \in \{1,2\}$, can be determined by any shortest path algorithm. However, at first sight it is not obvious how to compute the values $y_{-1}(u,v)$. It turns out that these values can be calculated by calling a shortest path algorithm twice. This will be the topic of the following subsection. Thus, the overall running time of our algorithm is $O(|V_D|^2(|T_1|+|T_2|)^2+\gamma)$, where γ is the time to compute shortest paths between all pairs of nodes.

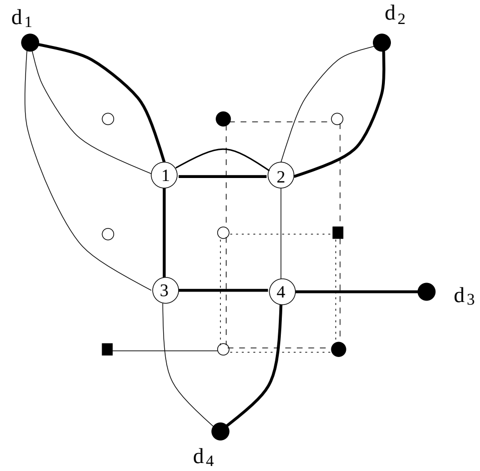

Figure 9:

Finally, let us remark that in our cutting plane algorithm for computing a minimum weight Steiner tree packing, we do not only try to determine the most violated extended alternating cycle inequality by using Recursion 4.6. Instead, we also compute Steiner trees for M in G_D heuristically. Again, we use the algorithm proposed by [TM80] with cost function $y^1 + y^2$. Thereafter, we determine the best possible sifting for the resulting Steiner trees.

Up to now, all partitions $V_1, \ldots, V_k$ of V considered in our separation algorithm have cardinality $k = |M|$. In other words, each node in the contracted graph $G(V_1, \ldots, V_k)$ is part of the alternating cycle. It is natural to generalize this in the sense that not all elements of the partition $V_1, \ldots, V_k$ belong to the alternating cycle, but are viewed as certain additional nodes in the contracted graph. We have analyzed this generalization from a theoretical point of view and have identified conditions under which the resulting inequalities are facet-defining (see [GMW92a]). Moreover, the dynamic program described above can be adapted to this generalization. It also provides a lower bound for the slack of the most violated inequality. A detailed description of this algorithm requires many technicalities that we do not want to present here. For a discussion of this separation algorithm we refer to [M92].

5 Determining Cheapest Paths with Costfree Edges

The following combinatorial optimization problem is an interesting variant of the shortest path problem. We are given a graph $G = (V, E)$ with costs $c_e \geq 0$ for all edges $c_e \in E$, two nodes $s, t \in V$, and a nonnegative integer k. We want to find a cheapest path from s to t where the "cost" of a path is the usual cost minus the sum of the costs of the k (or at most k) most expensive edges of the path. Another way to view the problem is the following. We have k tokens that allow us to use k (or at most k) edges for free. We want to choose a path from s to t and employ the k tokens to use k (or at most k) edges without any costs in such a way that the total sum paid for the use of the remaining edges is as small as possible. Clearly, this problem also has a directed version; we can similarly search for odd or even paths or cycles where k of the edges can be used for free.

We are particularly interested in the case of $[s,t]$-paths where $k=1$, since the computation of cheapest paths with one costfree edge is necessary to compute the values (1) $l_1(v,i,0) = y_{-1}(v,d_i) - \frac{1}{2}$ in Recursion 4.6. There is an obvious way to determine a cheapest $[s,t]$-path with one costfree edge. For every edge $e \in E$, we do the following: We define a new cost function by setting $c^e_f := c_f$, if $f \neq e$, and $c^e_e := 0$, and we compute a shortest $[s,t]$-path in G with cost function c^e. Every shortest of the $|E|$ $[s,t]$-paths determined this way is a cheapest $[s,t]$-path with one costfree edge. (This process can be clearly generalized to the case $k \geq 1$.) However, a cheapest $[s,t]$-path with one costfree edge can be computed faster by calling a shortest path algorithm only twice as follows.

Algorithm 5.1 (Cheapest paths from s to all other nodes with one costfree edge)

Input:

A graph $G=(V,E)$, edge costs $c_e \geq 0$, $e \in E$ and a node $s \in V$.

Output:

The costs of cheapest paths from s to all $v \in V$ with one costfree edge.

Datastructures:

$d(v)$ = *Length of a shortest $[s,v]$-path.*
$m(v)$ = *Cost of a cheapest path from s to v with one costfree edge.*
N = *List of unlabeled nodes that are incident to some labeled node.*

(1) Compute $d(v)$ for all $v \in V$ using a shortest path algorithm.
(2) Initialize $m(v) = d(v)$ for all $v \in V$.
(3) Set $m(s) = 0$ and $N = \emptyset$.
Label s, (all other nodes are supposed to be unlabeled).
For all nodes v adjacent to s set
$m(v) = 0$ *and* $N = N \cup \{v\}$.
(4) As long as there exists an unlabeled node, perform the following steps:
(5) Determine a node $v \in N$ with $m(v) = \min\{m(u) \mid u \in N\}$.
(6) Label v and set $N = N \setminus \{v\}$.
(7) For all nodes u adjacent to v perform the following steps:
If $\min\{m(v)+c_{vu},\ d(v)\} < m(u)$, *set*

$m(u) = \min\{m(v) + c_{vu},\, d(v)\}$.
$N = N \cup \{u\}$.
(8) Return the values $m(v)$ for all $v \in V$.
(9) STOP.

The following theorem states the correctness of the algorithm.

Theorem 5.2 *Let $G = (V, E)$ be a graph with nonnegative edge costs c_e, $e \in E$. Then, Algorithm 5.1 determines the cheapest path from s to v with one costfree edge, for all $v \in V$.*

Proof. To avoid confusion we use throughout this proof the following notation: For a path P from u to v, we denote by $c(P) = \sum_{e \in P} c_e$ the "length" of path P and use the term "shortest" if $c(P)$ is minimum among all $[u, v]$-paths. On the other hand, for a path P from u to v with one costfree edge, we call the value $c(P) - \max_{e \in P} c_e$ the "cost" of path P, and speak of a "cheapest" path P if the value $c(P) - \max_{e \in P} c_e$ is minimum among all $[u, v]$-paths.

By induction on the number of labeled nodes we show the following: If a node v is labeled, then $m(v)$ is the cost of a cheapest $[s, v]$-path with one costfree edge. In order to prove this, we need the property that, for all $v \in N$, $m(v)$ is the cost of a cheapest $[s, v]$-path with one costfree edge whose inner nodes are only labeled nodes. This will be simultanously shown by the induction.

If s is the only labeled node, the statement is true due to step (3) of Algorithm 5.1. Suppose, the statement is true for $i - 1$ labeled nodes and we have chosen an i-th node v, say, in step (5). We claim that $m(v)$ is the cost of a cheapest $[s, v]$-path with one costfree edge. If this is not the case, there exists a path from s to v with one costfree edge that is cheaper. Suppose P is such a path with cost m_P. Then, P must contain an edge that connects an unlabeled node with a labeled one. Let uw (with w unlabeled) be the first of these edges. Obviously, $w \in N$. From the assumption of the induction we know that $m(w)$ is the cost of a cheapest $[s, v]$-path with one costfree edge whose inner nodes are only labeled nodes. Thus, $m(w) \leq m_P < m(v)$, a contradiction to the choice of v.

It remains to be shown that for all unlabeled nodes $u \in N$ the value $m(u)$ is the cost of a cheapest $[s, u]$-path with one costfree edge whose inner nodes are labeled. We assume that v was the node chosen in step (5).

Due to the induction assumption, $m(u)$ is the cost of a cheapest $[s, u]$-path with one costfree edge whose inner nodes are labeled and different from v. This value is compared in step (7) with the cost of a cheapest $[s, u]$-path with one costfree edge whose predecessor is v and whose inner nodes are labeled. Suppose, there exists a $[s, u]$-path P with one costfree edge that is cheaper and whose inner nodes are labeled such that $v \in V(P)$ and $wu \in P$, $w \neq v$. Without loss of generality let P be the cheapest of those paths and m_P the cost of P. If $m_P = d(w)$ (i. e. wu is a maximal edge), we conclude that $m_P = d(w) \geq d(v) \geq m(u)$, a contradiction. Otherwise, $m_P = m(w) + c_{wu}$. Since w was labeled before v, there exists due to the assumption of induction a cheapest path P' from s to w with one costfree edge whose inner nodes are labeled and different from v. Let $m_{P'}$ be the cost of P'. We obtain that $m_P = m(w) + c_{wu} \geq m_{P'} + c_{wu} \geq m(u)$, a contradiction. This shows Theorem 5.2. □

6 Computational Results

In this section we report on the success of our separation algorithms for the solution of practical problem instances. We have developed a branch and cut algorithm to solve a certain class of Steiner tree packing problems arising in the design of electronic circuits. Here, the underlying graph is a complete rectangular grid graph and the set of terminals are located on the outer face. The task is to find a Steiner tree packing with minimal weight, where all edge weights are equal to one. These problems are called *switchbox routing problems* in the VLSI literature. We have tested our algorithm on switchbox routing problems discussed in the literature.

name	h	b	N	distribution of the nets					ref.
				2	3	4	5	6	
difficult switchbox	15	23	24	15	3	4	1	1	[BP83]
more difficult switchbox	15	22	24	15	3	5		1	[CH88]
terminal intensive switchbox	16	23	24	8	7	5	4		[L85]
dense switchbox	17	15	19	3	11	5			[L85]
augmented dense switchbox	18	16	19	3	11	5			[L85]
modified dense switchbox	17	16	19	3	11	5			[CH88]
pedagogical switchbox	16	15	22	14	4	4			[CH88]

Table 1:

Table 1 summarizes the data of our test problems. Column 1 presents the names of the instances used in the literature. In column 2 and 3 the height and width of the underlying grid graph is given. Column 4 contains the number of nets.

Columns 5 to 9 provide information about the distribution of the nets; more precisely, column 5 gives the number of 2-terminal nets, column 6 gives the number of 3-terminal nets and so on. Finally, the last column states the reference to the paper the example is taken from.

In [GMW92b] we report on our experiences for solving these problems with a branch and cut algorithm. For more details on these switchbox routing problems and on the general outline of our branch and cut algorithm we refer to that paper.

We focus in this section on our evaluation of the various separation algorithms described in the previous sections. We have, in total, inplemented 9 exact and heuristic separation routines. We have executed many test runs using just a single separation routine, two, three or more separation routines in various combinations and orders. It seems impossible to present all the data of these runs here and discuss the relative merits of the choices. We rather want to describe our final selection of separation algorithms and to indicate why we have made some of the choices.

Initially, we started with the trivial LP relaxation consisting of just the upper and lower bounds and the degree constraints for all terminals. This turned out to be a disastrous beginning. It took the separation routines almost forever to add sufficiently many cutting planes so that the graphs G^k induced by the edges $E^k := \{e \in E \mid x_e^k > 0\}$ became connected. We therefore added a preprocessing stage that, for each net, generates certain Steiner partition inequalities by analyzing the positions of the terminals of the net. In particular, our program determines all horizontal and vertical cuts that separate two terminals of a net and a number of further suitably chosen Steiner partition inequalities. In this stage we keep an eye on the spatial distribution of the corresponding cuts and multicuts, i .e., we try to select inequalities in such a way that almost every edge appears with a positive coefficient in one of the initial inequalities and only few edges occur in many inequalities. The reason for this rule is that, by this choice, the LP solver is unable to satisfy many inequalities at once by setting just a few variables to a positive value. Practically satisfactory rules for determining Steiner cut and partition inequalities of this type were found by running various combinations of choices and comparing the computational results on many instances. The introduction of this preprocessing stage was, in retrospect, decisive for the practical success of our approach.

For the separation of the Steiner partition inequalities we have programmed the exact separation routine described in section 3 and two heuristics. These heuristics determine cheap Steiner trees in the dual graph G_D introduced in section 3. The running times of these heuristics are only small fractions of the running time of the exact separation algorithm. Moreover, the heuristics tend to find significantly more violated constraints than the exact routine. Our experiments indicated that a certain combination of the heuristics and the exact method seems to perform best. We first run the two heuristics and stop the cutting plane generation if a certain threshold for the number of cutting planes that we want to generate at most in one iteration is surpassed. We control the heuristics by several parameters so that violated Steiner partition inequalities of different structure and small overlap are generated. The time consuming exact method is only called if none of the separation heuristics is able to find a violated Steiner partition inequality. Column 2 of Table 2 shows the number of Steiner partition inequalities generated during the runs of our final combination of exact and heuristic separation algorithms for the Steiner partition inequalities on the test instances. The results show that our methods are quite successful cutting plane generators.

example	Steiner part. ineq.	ext. alt. cycle ineq.
difficult switchbox	3966	565
more difficult switchbox	3494	475
terminal intensive switchbox	5718	981
dense switchbox	3409	469
augmented dense switchbox	5613	584
modified dense switchbox	3117	400
pedagogical switchbox	2139	448

Table 2:

Our computational experiments revealed that a similar strategy also yields the best results with respect to separating extended alternating cycle inequalities. Here our final choice was to execute the separation heuristic described in section 4 first and to call the dynamic program only if the separation heuristic failed to determine a violated extended alternating cycle inequality. Moreover, based on comparing the running time spent with the probability of success, we decided to call the separation algorithms for the extended alternating cycle inequalities not for all net pairs. Our choice is as follows. We determine "conflicting nets", i. e., those nets that our primal heuristic for finding a Steiner tree packing is unable to route simultanously, and run the separation routines for extended alternating cycle inequalities only for these pairs of nets. Column 3 of Table 2 shows the number of violated extended alternating cycle inequalities that were generated with these strategies for our test instances. Again, this combination of separation methods was highly successful.

example	best sol.	lower bound	gap
difficult switchbox	464	464	0.0%
more difficult switchbox	452	452	0.0%
terminal intensive switchbox	537	535	0.4%
dense switchbox	441	438	0.7%
augmented dense switchbox	469	467	0.4%
modified dense switchbox	452	452	0.0%
pedagogical switchbox	331	331	0.0%

Table 3:

Table 3 demonstrates the success of the separation algorithms. Column "lower bound" shows the LP-value at the end of the initial cutting plane phase before the branching phase of our code is entered. We are able to solve most of these examples to optimality without branching. For all other examples we obtain a bound that is at most 1% below the optimal solution value. This does not only indicate that the separation algorithms work very well, but also that the Steiner partition inequalities and the alternating cycle inequalities describe the (for our type of problems) relevant part of the Steiner tree packing polyhedron quite well. Moreover, it has turned out that most of the violated inequalities were found by the separation heuristics and that the dynamic programs were called only a few times. Thus, we have good hopes that this approach is also applicable to practical problem instances where only separation heuristics are at hand, i. e., where the underlying graph is not planar or the terminals are not located on a fixed number of faces.

References

[AMO93] R. K. Ahuja, T. L. Magnanti, J. B. Orlin: *Network flows: theory, algorithms, and applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[BP83] M. Burstein, R. Pelavin: *Hierarchical wire routing*, IEEE Transactions on Computer-Aided-Design CAD-2, 1983, 223 – 234.

[CH88] J. P. Cohoon, P. L. Heck: *BEAVER: A computational-geometry-based tool for switchbox routing*, IEEE Transactions on Computer-Aided-Design CAD-7, 1988, 684 – 697.

[CR88a] S. Chopra, M. R. Rao: *The Steiner tree problem I: Formulations, compositions and extension of facets*, Technical Report No. 88-82, Graduate School of Business Administration, New York University, New York, 1988.

[CR88b] S. Chopra, M. R. Rao: *The Steiner tree problem II: Properties and classes of facets*, Technical Report No. 88-83, Graduate School of Business Administration, New York University, New York, 1988.

[DW71] S. E. Dreyfus, R. A. Wagner: *The Steiner problem in graphs*, Networks 1, 1971, 195 – 207.

[EMV87] R. E. Erickson, C. L. Monma, A. F. Veinott: *Send-and-split method for minimum concave-cost network flows*, Mathematics of Operations Research 12, 1987, 634 – 664.

[GM90] M. Grötschel, C. L. Monma: *Integer polyhedra associated with certain network design problems with connectivity constraints*, SIAM Journal on Discrete Mathematics 3, 1990, 502 – 523.

[GMS92] M. Grötschel, C. L. Monma, M. Stoer: *Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints*, Operations Research 40, 1992, 309 – 330.

[GMW92a] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: polyhedral investigations*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 92-8, 1992.

[GMW92b] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: a cutting plane algorithm and computational results*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 92-9, 1992.

[GMW93] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: further facets*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 93-1, 1993.

[HRW92] F. K. Hwang, D. S. Richards, P. Winter: *The Steiner tree problem*, Annals of Discrete Mathematics 53, North-Holland Amsterdam, 1992.

[KL84] M. R. Kramer, J. van Leeuwen: *The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits*, F. P. Preparata (ed.): "Advances in Computing Research", Bd. 2: VLSI theory, Jai Press, London, 1984, 129 – 146.

[L85] W. K. Luk: *A greedy switch-box router*, Integration 3, 1985, 129 – 149.

[M92] A. Martin: *Packen von Steinerbäumen: Polyedrische Studien und Anwendung*, Ph.D. Thesis, Technische Universität Berlin, 1992.

[S87] M. Sarrafzadeh: *Channel-routing problem in the knock-knee mode is $\mathcal{NP}$-complete*, IEEE Transactions on Computer-Aided-Design CAD-6, 1987, 503 – 506.

[TM80] H. Takahashi, A. Matsuyama: *An approximate solution for the Steiner problem in graphs*, Math. Japonica 24, 1980, 573 – 577.