Steiner Trees with Degree Constraints: Structural Results and an Exact Solution Approach

Frauke Liers, Alexander Martin, Susanne Pape

20. November 2014

In this paper we study the Steiner tree problem with degree constraints. Motivated by an application in computational biology we first focus on binary Steiner trees in which all node degrees are required to be at most three. We then present results for general degree-constrained Steiner trees. It is shown that finding a binary Steiner is $\mathcal{NP}$-complete for arbitrary graphs. We relate the problem to Steiner trees without degree constraints as well as degree-constrained spanning trees by proving approximation ratios. Further, we give some integer programming formulation for this problem on undirected and directed graphs and study the associated polytope for both cases. Some classes of facets are introduced. Based on this study a branch-&-cut approach is developed and evaluated on biological instances coming from the reconstruction of phylogenetic trees. We are able to solve nearly all instances up to 200 nodes to optimality within a limited amount of time. This shows the effectiveness of our approach.

1 Introduction

Scientific or engineering applications often require the solution of optimization problems. Over the years the Steiner tree problem in graphs and its variants have taken an increasingly important role. Many real-life applications in network design in general and VLSI design in particular use Steiner trees to model and solve their problems.
Given an undirected graph $G = (V, E)$ and a terminal set $T \subseteq V$, a **Steiner tree** for T is a subset $X \subseteq E$ that spans all nodes in T. A Steiner tree may contain Steiner nodes of the set $S = V \setminus T$. The **degree-constrained Steiner tree problem** for G is stated as follows: Given a cost function $c : E \to \mathbb{R}_+$ and degree requirements $b(v) \in \mathbb{N}$ for all $v \in V$, find a minimum cost Steiner tree such that the degree at each node $v \in V$ is less than or equal to $b(v)$.
The Steiner tree is called **binary** if all nodes in X have a degree less than or equal to three. Given a cost function $c : E \to \mathbb{R}_+$, the **binary Steiner tree problem** is to find a minimum cost binary Steiner tree.

The Steiner tree problem without any degree constraints has been extensively studied in the literature, see for example Goemans and Myung [2006], Lucena [2005], Proemel and Steger [2002], Koch and Martin [1998], Lucena and Beasley [1998], Chopra and Rao [1994a,b], Hwang et al. [1992], Hwang and Richards [1992], Chopra et al. [1992], Maculan [1987], Winter [1987]. While the introduction of additional degree constraints has been received growing attention for spanning trees (see for example Cunha and Lucena [2007], Cunha [2006], Caccetta and Hill [2001]), studies about degree-constrained Steiner trees are rare. As a subcase of degree-constrained network design problems Khandekar et al. [2013], Louis and Vishnoi [2010], Lau et al. [2009], Bansal et al. [2009], Lau and Singh [2008], Ravi et al. [2001] introduced bicriteria approximation algorithms approximating both the objective value and the degree simultaneously. Besides, Furer and Raghavachari [1994], Khandekar et al. [2013] introduced a related problem of finding a Steiner tree of minimal degree. However, up to our knowledge there do not exist theoretical studies, exact algorithms or heuristics for degree- constrained Steiner tree problems. Hence, in this paper, we consider the subcase of binary Steiner trees and its extension to general degree-constrained Steiner trees. Binary Steiner trees are very important in biological and evolutionary questions, for example for constructing tree alignments or phylogenetic trees. From the viewpoint of biologists, the terminals of a binary Steiner tree represent the given taxa, for example extant species or biomolecular sequences. Steiner nodes are the extinct taxa, i.e. the common ancestors, and the edge length represents the evolutionary time or number of mutations between the taxa. The principle of Maximum Parsimony involves the identification of a phylogenetic tree that requires the smallest number of evolutionary changes. Unfortunately, inferring such trees is a difficult problem, see Felsenstein [2004] for a general introduction in the problem of inferring phylogenetic trees. This paper presents some theoretical and computational results on binary Steiner trees that may help to construct evolutionary trees.

The paper is organized as follows: In section 2 we show that the computation of any binary Steiner tree (not necessarily optimal) is already $\mathcal{NP}$-complete in general. Some approximation ratios based on Steiner trees without degree-constraints and binary spanning trees are presented in section 3. We introduce two integer programming formulations in section 4. The second one is a bidirected version of the first and used for the implementation of the branch-&-cut approach. The binary Steiner tree polytope is defined and some classes of valid and facet-inducing inequalities are considered in section 5. Separation routines for these inequality classes are presented. In Section 6 we introduce a primal heuristic. Section 7 generalizes our results to degree-constrained Steiner trees. Based on the polyhedral study we develop a branch-&-cut algorithm in section 8 and discuss computational results.

2 Complexity Results

In this section we show that already the construction of a binary Steiner tree (not necessarily optimal) is difficult.
Finding a Steiner tree for a given graph $G = (V, E)$ with terminal set $T \subseteq V$ and without any degree constraints is easy. It can be done by breadth-first-search in polynomial time. If we look for a binary Steiner tree, the problem becomes $\mathcal{NP}$-complete.

Problem 2.1. *Let $G = (V, E)$ be a graph with terminal set $T \subseteq V$. Does there exist a binary Steiner tree bST for G?*

Theorem 2.2. *The problem of finding a binary Steiner tree is $\mathcal{NP}$-complete.*

Proof. Obviously, the problem is in $\mathcal{NP}$.
To show completeness, we give a polynomial-time reduction from the $\mathcal{NP}$-complete problem of finding a vertex cover of size less than or equal to k, i.e. a subset $C \subseteq V$ with $|C| = k$ such that for all edges $\{u, v\} \in E$ the node $u \in C$ or the node $v \in C$ (or both).
Let $\overline{G} = (\overline{V}, \overline{E})$ be a graph for which we want to compute a vertex cover of size less than or equal to k. Denote $n = |\overline{V}|$ and $m = |\overline{E}|$. We construct a graph $G = (V, E)$ with terminal set T for which we want to construct a binary Steiner tree as follows:
The node set V: We introduce k nodes which we call $1, \dots, k$. For each node $v_i \in \overline{V}$ of degree $d_i := \deg(v_i)$ we construct d_i nodes $v_i^1, \dots, v_i^{d_i}$. Further, for each edge $e_l \in \overline{E}$ we have a node e_l in V. Finally, we introduce $t = (k + 1) + 2m$ terminal nodes which we call t_p, $p = 1, \dots, t$. Hence,

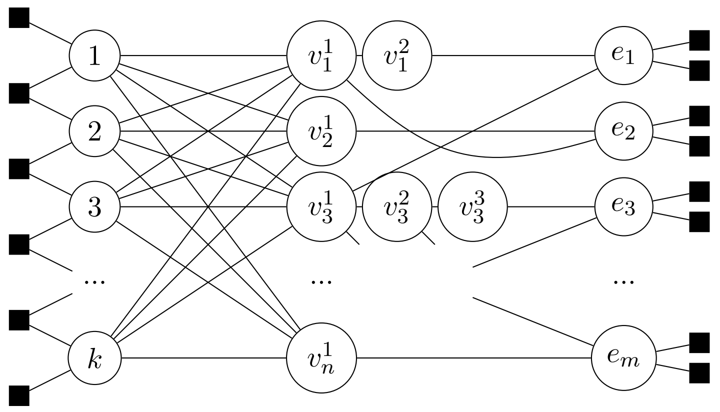


Figure 1: **Reduction from the problem of finding a vertex cover of size k in $\overline{G} = (\overline{V}, \overline{E})$:** The figure shows the constructed graph $G = (V, E)$ as in the proof of Theorem 2.2. Here, the black squares are the terminal nodes.

we have

$$V := \{1, \dots, k\} \cup \{v_i^j : v_i \in \overline{V},\ 1 \le j \le d_i\} \cup \{e_l : e_l \in \overline{E}\} \cup \{t_p : p = 1, \dots, t\}$$

and $T = \{t_p : p = 1, \dots, t\}$.
The edge set E: We connect each node $1, \dots, k$ to all "first" nodes v_i^1, $i = 1, \dots, n$ by one edge, respectively. Then, for each $i = 1 \dots, n$ we introduce an edge $\{v_i^j, v_i^{j+1}\}$ for all $j = 1, \dots, d_i - 1$ to get a path from v_i^1 to $v_i^{d_i}$. Further, if $e_l = \{v_{i_1}, v_{i_2}\} \in \overline{E}$ is an edge in the original graph, we connect the node e_l in V to two nodes $v_{i_1}^{j_1}$ and $v_{i_2}^{j_2}$ with $j_1 \in \{1, \dots, d_{i_1}\}$, $j_2 \in \{1, \dots, d_{i_2}\}$ chosen in such a way that in the end each node v_i^j, $i = 1, \dots, n$, $j = 1, \dots, d_i$ is connected to exactly one node e_l, $l = 1, \dots, m$. This is possible because the number of nodes v_i^j is identical to twice the number of nodes e_l, i.e. $2m = \sum_{i=1}^n d_i$. Finally, we have to connect the terminal nodes. The first $k+1$ terminal nodes are connected by a path $P = (t_1, 1, t_2, 2, \dots, k, t_{k+1})$. Further, each node e_l is connected to two remaining terminal nodes t_p of degree 1. Hence, we have

$$E := \bigcup_{i=1}^{5} E_i$$

with

$$\begin{array}{lcl}
E_1 & := & \{e = \{h, v_i^1\} : h = 1 \dots, k,\ i = 1 \dots, n\} \\
E_2 & := & \{e = \{v_i^j, v_i^{j+1}\} : i = 1 \dots, n,\ j = 1, \dots, d_i - 1)\} \\
E_3 & := & \{e = \{v_i^j, e_l\} : e_l = \{v_i, \cdot\} \in \overline{E},\ \text{appropriate } j\} \\
E_4 & = & \{e = \{t_p, p\} : p = 1, \dots, k\} \cup \{e = \{t_{p+1}, p\} : p = 1, \dots, k\} \\
E_5 & := & \{e = \{e_l, t_{k+1+l}\} : l = 1 \dots, m\} \cup \{e = \{e_l, t_{k+1+m+l}\} : l = 1 \dots, m\}
\end{array}$$

The constructed graph $G = (V, E)$ is shown in Figure 1.
Obviously, the construction can be done in polynomial time.
We show that $\overline{G} = (\overline{V}, \overline{E})$ has a vertex cover of size less than or equal to k if and only if $G = (V, E)$ has a binary Steiner tree.

Let $\overline{G} = (\overline{V}, \overline{E})$ have a vertex cover of size less than or equal to k. We construct a binary Steiner tree in $G = (V, E)$ in the following way: W.l.o.g. let $C = \{v_1, \dots, v_k\}$ be a vertex cover of $\overline{G}$. We now design a binary Steiner bST tree in G: First, we connect the first $k+1$ terminal nodes t_p, $p = 1, \dots, k+1$ and the nodes $1, \dots, k$ by the path P defined above. Then, we connect

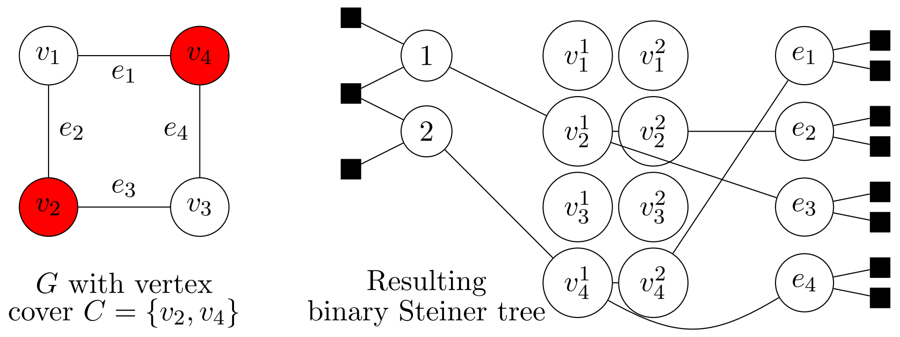


Figure 2: **Example for the construction of a binary Steiner tree in G given a vertex cover in $\overline{G}$:** The vertex cover in $\overline{G}$ consists of the two nodes v_2 and v_4. Hence, the resulting binary Steiner tree in G contains the two Steiner nodes $1, 2$. The nodes $v_2^1, v_2^2, v_4^1, v_4^2$ cover all nodes e_i, $i = 1, \ldots, 4$.

node $i = 1, \ldots, k$ with node v_i^1 and each v_i^1, $i = 1, \ldots, k$ through a path $(v_i^1, v_i^2, \ldots, v_i^{d_i})$ with $v_i^{d_i}$. Since C is a vertex cover, C covers all edges in $\overline{E}$. Hence, for each node $e_l \in V$, we can find one node v_i^j with $i \in \{1, \ldots, k\}$ and suitable $j \in \{1, \ldots, d_i\}$ and add the edge $\{v_i^j, e_l\}$ to bST. Finally, we connect the remaining terminal nodes. Now, all nodes despite the Steiner nodes $v_i^1, \ldots, v_i^{d_i}$ with $i = k+1, \ldots, n$ are connected, there are no cycles and the resulting tree is binary by construction. A small example with $C = \{v_2, v_4\}$ can be found in Figure 2.

For the other direction, let bST be a binary Steiner tree in G. In bST, each node e_l is connected to two terminal nodes. Further, in bST, each e_l has a degree less than or equal to three because bST is binary. Since each e_l has to be connected to the rest of the tree, there is exactly one edge between e_l and one v_i^j for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, d_i\}$. W.l.o.g. let $v_1^{j_1}, \ldots, v_q^{j_q}$ with $j_i \in \{1, \ldots, d_i\}$ be the nodes connected to the nodes e_l, $l = 1, \ldots, m$. If $q \leq k$, then $C = \{v_1, \ldots, v_q\}$ is a vertex cover in $\overline{G}$ of size $q \leq k$ and we are done. So assume that $q > k$. Note that in our construction of bST so far, bST is not connected, but is composed of l connected components. This number may be reduced to q components through edges in E_2. However, these q components cannot be directly connected in bST. Hence, to connect these at least q components there is (at least) one node $h \in \{1, \ldots, k\}$ that is connected to two (or three) nodes in $\{v_1^1, \ldots, v_q^1\}$. Since bST is binary, there is at most one terminal node connected to node h. Hence, bST is not connected. This is a contradiction, since bST is a tree. Hence, $q \leq k$.

After all, $\overline{G} = (\overline{V}, \overline{E})$ has a vertex cover of size less than or equal to k if and only if $G = (V, E)$ has a binary Steiner tree.
Thus, there is a polynomial time reduction from the problem of finding a vertex cover of size less than or equal to k to the problem of computing a binary Steiner tree. Hence, Problem 2.1 is $\mathcal{NP}$-complete. □

The subcase $T = V$, i.e. finding a binary spanning tree is also $\mathcal{NP}$-complete. It can be shown by a reduction of the Hamiltonian path problem. In contrast, the case $|T| = 2$ is easy because we just have to find a path between two nodes. In addition, binary Steiner trees always exist in complete graphs. However, the computation of a minimal one remains $\mathcal{NP}$-hard.

3 Approximation Ratios

Two problems very close to the binary Steiner tree problem are the original Steiner tree problem and the problem of computing binary spanning trees. In this section we study the ratio of the minimal binary Steiner tree optimum to the optimal value for minimal Steiner trees or minimal binary spanning trees, respectively.

3.1 Approximation with Minimal Steiner Trees

We start with analyzing the lower bound given by minimal Steiner trees. Let $G = (V, E, c)$ be an undirected weighted graph with $c \geq 0$ and terminal set $T \subseteq V$. Let $ST(G) = ST$ be a minimal Steiner tree for G and $bST(G) = bST$ a minimal binary Steiner tree. Obviously, we have

$$D := \frac{c(bST)}{c(ST)} \geq 1,$$

since each binary Steiner tree is also a Steiner tree. Can D become arbitrary large, i.e. for example

$$\lim_{|V|\to\infty} D = \infty \quad \text{or} \quad \lim_{c_{\max}\to\infty} D = \infty\ ?$$

For non-metric graphs, the answer can be "yes". There exist simple examples with $D \to \infty$, even if $c(e) = 1$ for all $e \in E$. However, if we restrict to complete metric graphs, the answer is "No". Similar to Khuller et al. [1996] and Ravi et al. [1993] who answered this problem for the degree-constrained spanning tree case, we can show that $D \leq 2$ in complete, metric graphs, i.e. in graphs with a nonnegative and symmetric cost function that satisfies triangle inequality.

Theorem 3.1. *Let $G = (V, E, c)$ be a complete undirected graph with a metric cost function c. Then there exists a binary Steiner tree B whose weight is at most twice the weight of the minimal Steiner tree for G, i.e.*

$$c(bST) \leq c(B) \leq 2 \cdot c(ST),$$

that means

$$D := \frac{c(bST)}{c(ST)} \leq 2.$$

Proof. Let ST be a minimal Steiner tree for G. Construct a binary Steiner tree B as follows:

1. Root ST at an arbitrary leaf $r \in V(ST)$.
2. Partition the edge set of ST into n_i sets where n_i is the number of internal nodes in ST. For an internal node v such a set consists of edges going from v to its children. In each set sort the edges in non-decreasing order with respect to the edge costs, i.e. if a node v has $b(v)$ children $v_1, \ldots, v_{b(v)}$, then $c(v, v_i) \leq c(v, v_{i+1})$ for $i = 1, \ldots, b(v) - 1$.
3. While there exist internal nodes $v \in V(ST)$ with degree greater than three, do the following:
   - if $\deg_{ST}(v) = b(v)+1$, replace in its corresponding edge set the edges $\{v, v_2\}, \ldots, \{v, v_{b(v)-1}\}$ by the path $P_v = (v_1, v_2, \ldots, v_{b(v)-1})$. (This is the case for an internal node at the beginning of the construction.)
   - otherwise replace in its corresponding edge set (this set does not get modified during the procedure!) the edges $\{v, v_2\}, \ldots, \{v, v_{b(v)}\}$ by the path $P_v = (v_1, v_2, \ldots, v_{b(v)})$. (During construction it is possible that $\deg_{ST}(v) = b(v) + 2$.)

Obviously, the replacement of subtrees forms again a tree. Each vertex v is on at most two paths and is an interior vertex of at most one path. Hence, each vertex has degree at most three.
For each $v \in V(ST)$ denote with *child*(v) the nodeset of children of v in ST. Since c is a metric, it satisfies the triangle inequality. Hence, for each edge $\{u, w\}$ in the path P_v with $u \neq v$ we have

$$c_{uw} \leq c_{uv} + c_{vw}.$$

We conclude that for each path P_v we have

$$c(P_v) = \sum_{e \in P_v} c_e \leq 2 \cdot \sum_{e=\{v,w\}, w \in child(v)} c_e,$$

i.e. each path P has weight at most twice the weight of the edges it replaces.
In summary, we get that

$$c(B) \leq 2 \cdot c(ST).$$

□

Remark 3.2. *We make the following observations:*

1. *Since*

$$c(bST) \leq c(B) \leq 2 \cdot c(ST) \leq 2 \cdot c(bST)$$

   the construction of B like in the proof of Theorem 3.1 yields an approximation algorithm with approximation ratio 2. However, it requires the computation of a minimal Steiner tree which is $\mathcal{NP}$-hard in general.

2. *For the correctness of the proof the order of edges (step 2.) does not matter. However, in practice a shortest path should be preferred.*

In fact, we can construct examples in which the ratio comes indeed arbitrarily close to two.

3.2 Approximation with Minimal Binary Spanning Trees

We now look at another ratio, the so called **Steiner ratio**. The Steiner ratio is the largest possible ratio between the total length of a minimum spanning tree and the total length of a minimum Steiner tree. We consider the Steiner ratio for binary trees.
Again, let $G = (V, E, c)$ be an undirected weighted graph with $c \geq 0$ and terminal set $T \subseteq V$. Let $bSP(G) = bSP$ be a minimal binary spanning tree for $G(T)$ and $bST(G) = bST$ a minimal binary Steiner tree for G. Obviously, we have

$$R := \frac{c(bSP)}{c(bST)} \geq 1,$$

since each binary spanning tree is also a binary Steiner tree.

Can R become arbitrary large? For general graphs and arbitrary objective function c, there exist simple examples with $R \to \infty$. However, for metric cost functions the answer is "No". Similar to Gilbert and Pollak [1968] (with reference to E.F. Moore) who studied the Steiner ratio for Steiner trees without any degree constraints, we can show that $R \leq 2$ by constructing a Traveling Salesman Tour through T, i.e. a tour that visits every node in T exactly once.

Theorem 3.3. *Let $G = (V, E, c)$ be a complete undirected graph with a metric cost function c and $T \subseteq V$. Then there exists a Traveling Salesman Tour TSP through T (and therefore a binary spanning tree B) whose weight is at most twice the weight of the minimal binary Steiner tree for G, i.e.*

$$c(bSP) \leq c(B) \leq c(TSP) \leq 2 \cdot c(bST),$$

that means

$$R := \frac{c(bSP)}{c(bST)} \leq 2.$$

Proof. This proof uses the ideas from the minimum spanning tree heuristic for the Traveling Salesman Problem (for a good description see Applegate et al. [2006]):
Let bST be a minimal binary Steiner tree and let $|V(bST)| = n$. By replacing each edge in bST by two parallel edges, we get a graph where each node has even degree. Hence, this graph has a Eulertour ET with $c(ET) = 2c(bST)$. We construct a Traveling Salesman Tour TSP from ET: Let $ET = (v_1, P_1, v_2, P_2, \ldots, P_n, v_1)$ where $v_1, \ldots, v_n$ are the n nodes of bST and $P_1, \ldots, P_n$ are (maybe empty) node-sequences which consist of nodes that are visited more than once. Then, by skipping the additional node-sequences $P_1, \ldots, P_n$ we get the Traveling Salesman Tour $(v_1, \ldots, v_n)$ through $V(bST)$. Skipping the Steiner nodes results in a tour TSP through T. Since c is a metric, we get that $c(ET) \geq c(TSP) \geq c(TSP^{min})$. By deleting any edge from TSP or TSP^{min}, we obtain a binary spanning tree B (without Steiner points). Since $c \geq 0$ we have

$$2c(bST) = c(ET) \geq c(TSP) \geq c(TSP^{min}) \geq c(B) \geq c(bSP).$$

In conclusion

$$\frac{c(bSP)}{c(bST)} \leq 2.$$

□

Remark 3.4. *We make the following two observations for metric cost functions:*

1. *Theorem 3.3 and its proof show that the computation of a minimal binary spanning tree or the computation of a minimal Traveling Salesman Tour both give a approximation algorithm with approximation ratio 2 for the binary Steiner tree problem. However, the computation of a minimal binary spanning tree or Traveling Salesman Tour is $\mathcal{NP}$-hard in general.*

2. *Each polynomial approximation algorithm with approximation ratio α for the minimal binary spanning tree problem or the minimal Traveling Salesman Tour yields a polynomial approximation algorithm with ratio 2α for the binary Steiner tree problem:*
Let B be the binary spanning tree (TSP) produced by an α-approximation algorithm. Then

$$c(B) \leq \alpha \cdot c(bSP) \quad \Rightarrow \quad \text{(with Theorem 3.3)} \;\; c(B) \leq \alpha \cdot c(bSP) \leq 2\alpha \cdot c(bST).$$

For example, the $\frac{3}{2}$-approximation algorithm of Christofides [1976] for the TSP therefore gives a polynomial 3-approximation algorithm for the binary Steiner tree problem.

For the case of Steiner trees without degree constraints there do exists examples for which the Steiner ratio comes arbitrarily close to two, i.e. the ratio of two is the best one can achieve. Up to our knowledge, for binary Steiner trees there do not exist examples for which the Steiner ratio is exactly two or comes arbitrarily close to two. Therefore, it may be possible to improve the ratio.

The shown approximation results just work for metric cost functions. For general Steiner tree problems there do not exist polynomial-time approximation algorithms for any error bound α which can be shown by a reduction of the Hamiltonian path problem.

4 IP-Models for Binary Steiner Trees

We introduce two different IP-models for binary Steiner trees adapted from models used for Steiner trees without degree constraints, see for example Goemans and Myung [2006]. Both models share the same input information and describe the same restrictions. The first modeling approach is based on connectivity requirements, so-called cut constraints. The second model is the directed version of the first approach. Of course, there do exist further IP-model for Steiner trees and therefore for binary Steiner trees, see for example Polzin and Daneshmand [2001] or Maculan [1987]. However, we use the two most intuitive ones that are already used by Chopra and Rao [1994a] for their study of the Steiner tree polytope.

$$\min \sum_{e\in E} c_e x_e$$

$$(P_1) \qquad \begin{array}{rcll} \sum\limits_{e\in\delta(W)} x_e & \geq & 1 & \forall W \subseteq V,\ W \cap T \neq \emptyset,\ (V\setminus W)\cap T \neq \emptyset \\ \sum\limits_{e=\{i,j\}\in E} x_e & \leq & 3 & \forall i \in V \\ x_e & \in & \{0,1\} & \forall e \in E \end{array}$$

$$\min \sum_{a\in A} c_a x_a$$

$$(P_2) \qquad \begin{array}{rcll} \sum\limits_{a\in\delta^+(W)} x_a & \geq & 1 & \forall W \subseteq V,\ s \in W,\ (V\setminus W)\cap T \neq \emptyset \\ \sum\limits_{a=(i,j),a=(j,i)\in A} x_a & \leq & 3 & \forall i \in V \\ x_a & \in & \{0,1\} & \forall a \in A \end{array}$$

Figure 3: **Model 1 and model 2:** Integer formulation (P_1) for undirected graphs and formulation (P_2) for directed graphs.

4.1 Model 1 - Undirected Cut Formulation

Let $G = (V, E, c)$ be an undirected weighted graph with $c \geq 0$ and let $T \subseteq V$ be the set of terminal nodes. For each edge $e \in E$ we introduce a binary variable $x_e \in \{0, 1\}$ which is set to one if the edge e is in the binary Steiner tree and to zero, otherwise. The integer formulation (P_1) is stated in Figure 3, above. The first constraint set states that the tree should be connected as for each nodeset $W \subseteq V$ with $W \cap T \neq \emptyset$ and $(V \setminus W) \cap T \neq \emptyset$ there is at least one edge in the cut $\delta(W)$. The second constraints are degree constraints. Note that due to the nonnegativity of the objective function we do not need inequalities that forbid cycles because each optimal solution is cycle-free or can be transformed to one.

4.2 Model 2 - Directed Cut Formulation

Let $G = (V, E, c)$ be an undirected weighted graph with $c \geq 0$ and let $T \subseteq V$ be the set of terminal nodes. We replace each edge $e = \{i, j\} \in E$ by two antiparallel arcs (i, j), (j, i) with the same weight c_e. Let A denote the set of arcs and $D = (V, A, c)$ the resulting digraph. We choose some root $s \in T$. We now compute a rooted binary Steiner tree such that the tree contains a directed path from the root to each terminal node. Obviously, there is a one-to-one correspondence between binary Steiner trees in G and rooted binary Steiner trees in D.
As before we introduce a binary variable $x_a \in \{0, 1\}$ for all $a \in A$ which is set to one if the arc a is in the rooted binary Steiner tree and to zero, otherwise. The directed integer formulation (P_2) is also stated in Figure 3 (below). Again, the first constraint set states that the tree should be connected. The second constraints are degree constraints.

4.3 Comparison of the Two Models

Now we compare the two different IP-models. Of course, both models compute the same optimal value. However, if we look at the LP-relaxation, we observe the following similar to Chopra and Rao [1994a]:

Observation 4.1 (LP Relaxation)**.** *Let $LP_i(I)$ denote the optimal value of the LP-relaxation of* *an instance I for model i, $i = 1, 2$. Then*

$$LP_1(I) \leq LP_2(I).$$

Hence the LP-relaxations of model 2 gives better bounds on the optimal value than model 1. We therefore restrict in the branch-&-cut algorithm to model 2. However, we first present some polyhedral results for the first model because the proofs are much simpler and shorter. We later extend the results to the directed version.

5 Structural Results for Undirected and Directed Binary Steiner Trees

We now look at both IP-models and identify valid inequalities to strengthen each formulation. We introduce the undirected and directed binary Steiner tree polytope, i.e. the convex hull of all feasible solutions and describe some of their facets. We start with the undirected binary Steiner tree polytope P. The directed binary Steiner tree polytope P^D for directed graphs is defined in an analogous way and is explained later in less detail.

5.1 The Undirected Binary Steiner Tree Polytope

We start with the introduction of the binary Steiner tree polytope P for undirected graphs $G = (V, E)$. Model 1 gives rise to the following definition of the binary Steiner tree polytope.

Definition 5.1 (Binary Steiner Tree Polytope)**.** *Let $G = (V, E)$ be an undirected graph with terminal set $T \subseteq V$. We denote with*

$$\begin{array}{rcl} P(G) & := & \operatorname{conv}\{x \in \{0,1\}^{|E|} : \ x \textit{ is incidence vector of a bST of } G\} \\ & = & \operatorname{conv}\{x \in \{0,1\}^{|E|} : \ \sum_{e\in\delta(W)} x_e \geq 1 \ \forall\ W \subseteq V,\ W \cap T \neq \emptyset, \\ & & \quad (V \setminus W) \cap T \neq \emptyset,\ \sum_{e\in\delta(v)} x_e \leq 3 \ \forall\ v \in V\} \end{array}$$

the ***binary Steiner tree polytope*** *of G.*

Let us first investigate the dimension of the binary Steiner tree problem. Consider the following decision version of the dimension problem:

Problem 5.2. *Let $G = (V, E)$ be an undirected graph with $T \subseteq V$ and $d \geq 0$. Is the dimension of $P(G)$ at least d?*

As we have mentioned earlier, the decision problem "Does there exist a binary Steiner tree for a given graph G" is $\mathcal{NP}$-complete. Hence, problem 5.2 is also $\mathcal{NP}$-complete, even for the case $d = 0$.

This result does not give much hope for a successful study of binary Steiner tree polyhedra for trees defined on general graphs. Hence, we have decided to consider the binary Steiner tree polyhedron for special instances for which the dimension can be determined easily. Therefore, from now on, we restrict ourselves to complete graphs K_n with $n \geq 5$ nodes.

Lemma 5.3. *Let $n \geq 5$ and $G = K_n$. Then, $P(G)$ is full-dimensional, i.e.*

$$\dim(P(G)) = |E(G)| = \frac{n(n-1)}{2}.$$

Proof. W.l.o.g. let $V = \{v_1, v_2, \ldots, v_n\}$. Consider the following subgraphs of G:

$$\begin{array}{rcll} T_0 & := & \{\{v_i, v_{i+1}\} : \ i = 1, \ldots, n-1\}, & \\ T_e & := & T_0 \cup \{e\} & \forall\ e \in E(G) \setminus T_0, \\ T_i & := & (T_0 \cup \{\{v_1, v_n\}\}) \setminus \{\{v_i, v_{i+1}\}\} & \forall\ i = 1, \ldots, n-1. \end{array}$$

Obviously, the incidence vectors of these subgraphs are in $P(G)$ and affinely independent. We have

$$1 + (|E(G)| - (n-1)) + (n-1) = |E(G)| + 1$$

trees. Hence, $P(G)$ is full-dimensional. □

We now analyze the binary Steiner polytope $P(G)$. So far, we have defined this polytope as the convex hull of all incidence vectors of valid binary Steiner trees of a given directed graph $G = (V, E)$. The goal of the next subsection is to describe the polytope through its facets instead of its vertices.

5.2 Polyhedral Study of the Undirected Binary Steiner Tree Polytope

Binary Steiner trees are the common integer points of the Steiner tree polytope and the 3-matching polytope. However, in the literature only few cases exist where the intersection of two integer polytopes remains integer. We can construct points in the intersection of the Steiner tree polytope and the 3-matching polytope that are not within the binary Steiner tree polytope. Hence, the intersection of these two polytopes is not equal to the binary Steiner tree polytope. We will see later, that the binary Steiner tree polytope has indeed facets that are not facets of either of the two other polytopes.
We start with classical inequalities and then study the inequalities coming from the Steiner tree and the 3-matching polytope. Finally we introduce a completely new class of facet-defining inequalities.

5.2.1 Bounds and Degree Constraints

In this subsection we introduce the most simplest classes of facet-defining inequalities for the binary Steiner tree polytope. We start with the trivial inequalities, i.e. the lower and upper bound for each variable.

Theorem 5.4 (Trivial Inequalities). *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. Then, the trivial inequalities $0 \leq x \leq 1$ are valid and facet-defining for $P(G)$.*

Proof. Obviously, the inequalities $0 \leq x \leq 1$ are valid for $P(G)$.
To show that these inequalities induce facets, consider the subgraphs T_0, T_e and T_i defined in the proof of Lemma 5.3. A detailed proof can be found in the forthcoming PhD-Thesis of Pape [2015]. □

Besides the upper and lower bounds the degree constraints define another simple class of facet-inducing inequalities for $P(G)$:

Theorem 5.5 (Degree Constraints). *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. For $v \in V$, the degree constraint $deg(v) \leq 3$ is valid and facet-defining for $P(G)$.*

Proof. Obviously, the degree constraint $deg(v) \leq 3$ for $v \in V$ is valid for $P(G)$.
The facet-defining property can be easily shown in an indirect way. A detailed proof can be found in the forthcoming PhD-Thesis of Pape [2015]. □

Since the binary Steiner tree polytope is the convex hull of all integer points in the intersection of Steiner tree polytope and 3-matching polytope, we now analyze how the results of Steiner tree or matching polytope carry over to the binary Steiner tree polytope. Trivially, inequalities valid for either of these polytopes remain valid for the intersection.

5.2.2 Facets Coming from the Steiner Tree Polytope

We start with the Steiner tree polytope without degree constraints and show that the partition inequalities and odd-hole inequalities of this polytope remain facets if we introduce the degree constraints.

Let $k \in \mathbb{N}$, $k \geq 2$. Consider a partition $P = (V_1, \dots, V_k)$ of the nodes with $V = \bigcup_{i=1}^{k} V_i$ and $V_i \cap V_j = \emptyset$ for $i, j = 1, \dots, k, i \neq j$. If $V_i \cap T \neq \emptyset$ for $i = 1, \dots, k$, i.e. each subset V_i contains at least one terminal node, the partition is called a **Steiner partition**. Let $\delta(P)$ be the set of edges having endpoints in two distinct subsets of the partition.

Theorem 5.6 (Partition Inequalities). *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. Given a Steiner partition P, the partition inequality*

$$\sum_{e \in \delta(P)} x_e \geq k - 1$$

is valid for the binary Steiner tree polytope $P(G)$. It is facet-defining if and only if V_i is not composed of exclusively two terminal nodes for all $i = 1, \dots, k$.

Proof. Let $G = K_n = (V, E)$ and $P = (V_1, \dots, V_k)$ be a Steiner partition of the nodes. The proof of validity of the partition inequality is similar to that of Lemma 5.1.1. in Chopra and Rao [1994a].

We proof the facet-defining properties:
First let V_i be not composed of exactly two terminal nodes for all $i = 1, \dots, k$. To show that the partition inequality is facet-defining for $P(G)$, consider any valid inequality for $P(G)$

$$a^T x \geq b$$

such that

$$F := \{x \in P(G) : \sum_{e \in \delta(P)} x_e = k - 1\} \quad \subseteq \quad \{x \in P(G) : \ a^T x = b\}.$$

We show that $a^T x \geq b$ is a nonnegative multiple of the partition constraint:
We first show that $a_e = a_f$ for all $e, f \in \delta(P)$. W.l.o.g. let each node set V_i in the partition consist of l_i nodes and let $V_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_{l_i}}\}$ for $i = 1, \dots, k$. In each node set V_i, we construct the following tree

$$T_i := \{\{v_{i_k}, v_{i_{k+1}}\} : \ k = 1, \dots, l_i - 1\}.$$

T_i is a path, hence all nodes in T_i have degree one or two. We now construct the following binary Steiner trees (see Figure 4):

$$T_e^1 := \bigcup_{i=1}^{k} T_i \ \cup \ \{\{v_{i_{l_i}}, v_{(i+1)_1}\} : i = 2, \dots, k-1\} \ \cup \ \{e\} \quad \forall \ e \in \delta(V_1).$$

T_e^1 is a binary Steiner tree and its incidence vector $x(T_e^1)$ is in F. Moreover, two trees T_e^1 and T_f^1 differ only in the two edges $e, f \in \delta(V_1)$. Hence, $a_e = a_f$ for all $e, f \in \delta(V_1)$.
In the same way, we can show for each $i = 2, \dots, k$ that $a_e = a_f$ for $e, f \in \delta(V_i)$.
Since $\delta(V_i) \cap \delta(V_j) \neq \emptyset$ (each edge $e = \{v_{i.}, v_{j.}\} \in \delta(P)$ is in $\delta(V_i)$ and $\delta(V_j)$), it follows that $a_e = a_f$ for all $e, f \in \delta(P)$.
We now show that $a_e = 0$ for all $e \notin \delta(P)$. W.l.o.g. let $e \in E(V_1)$. If $|V_1| = 1$ there does not exist such an edge e. For $|V_1| = 2$, the set V_1 contains one Steiner node and one terminal node by assumption. Let v_{1_1} be the one Steiner node and v_{1_2} the one terminal node. Hence, $e = \{v_{1_1}, v_{1_2}\}$. Consider the tree $T_f^1 \in F$ with $f \in \delta(V_1) \cap \delta(v_{1_1})$. Then $T_f^1 \setminus \{e\}$ and therefore $a_e = 0$. Finally, let $|V_1| \geq 3$. If $e \notin T_1$, we add e to T_f^1 with $f \in \delta(v_{1_1})$, since each node in T_1 has a degree less

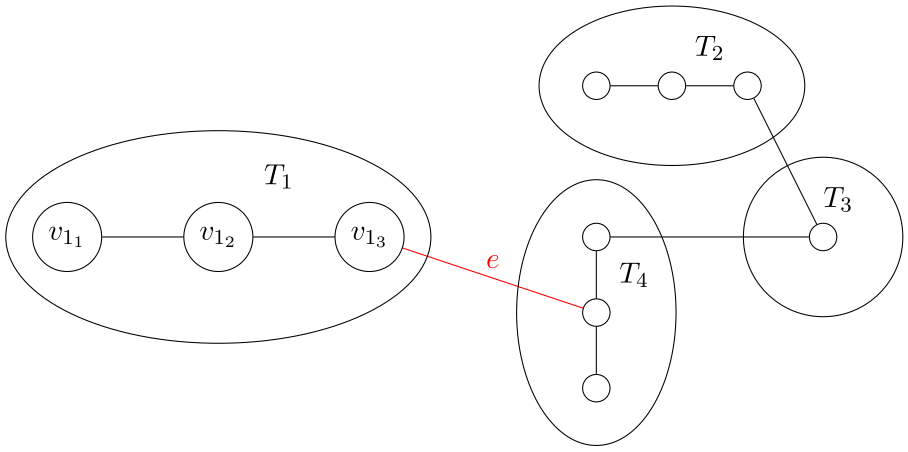


Figure 4: **Partition inequality:** The binary Steiner tree T_e^1 introduced in the proof of Theorem 5.6 satisfies the partition inequality with equality.

than three. The incidence vector $x(T_f^1 \cup \{e\})$ still remains in F. Thus, $a_e = 0$. For $e \in T_1$ we consider a different permutation π of the nodes in V_1 such that $e \notin \pi(T_1)$. In conclusion, $a_e = 0$ for all $e \notin \delta(P)$.

In conclusion, $a^T x \leq b$ is a nonnegative multiple of the partition constraint. This implies that the partition constraint is facet-defining for $P(G)$.

Finally, if $|V_i| = 2$ for at least one $i \in \{1, \ldots, k\}$ containing exactly two terminal nodes, the partition inequality can be written as the sum of another partition inequality and a trivial inequality: Let $i \in \{1, \ldots, k\}$ with $|V_i| = 2$. Let $u, w \in V_i$. Denote with P' the partition constructed through P by separating V_i into two sets $\{u\}$ and $\{w\}$, each containing only one terminal node. Then

$$x_{uw} \leq 1, \quad \text{and} \quad \sum_{e \in \delta(P')} x_e \geq k$$

are valid for $P(G)$. Further

$$\sum_{e \in \delta(P)} x_e = \sum_{e \in \delta(P')} x_e - x_{uw} \geq k - 1.$$

Hence, the partition inequality does not induce a facet. □

For instances with $|T| = |S|$, i.e. the number of terminal nodes is equal to the number of Steiner nodes, we now give another class of facet-inducing inequalities coming from the Steiner tree polytope. Consider a graph $G_k = (V, E_k)$ with $T \subseteq V$ and $|T| = |S| = k$. Let $T = \{t_1, \ldots, t_k\}$ and $S = \{s_1, \ldots, s_k\}$. The edge set E_k consists of two circles

$$E_k := E(C_k^1) \cup E(C_k^2)$$

with $C_k^1 := (t_1, s_1, t_2, s_2, \ldots, t_k, s_k, t_1)$ and $C_k^2 := (s_1, s_2, \ldots, s_k, s_1)$, compare Figure 5.

Lemma 5.7 (Odd-Hole Inequalities). *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. Let $|T| = |S| = k \geq 3$ and G_k be defined as above.*

1. *The hole inequality*

$$\sum_{e \in E_k} x_e \geq 2(k-1)$$

is valid for $P(G_k)$. It defines a facet for $P(G_k)$ if k is odd.

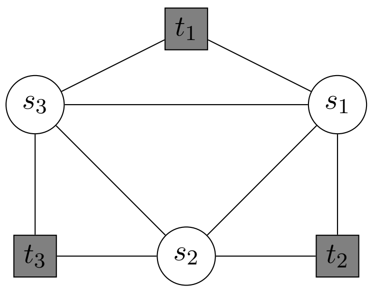


Figure 5: **Graphs with $|T| = |S|$:** The graph G_k for $k = 3$ as introduced in Lemma 5.7.

2. *The lifted hole inequality*
$$\sum_{e \in E_k} x_e + 2 \sum_{e \in E \setminus E_k} x_e \geq 2(k-1)$$
is valid for $P(G)$. It is facet-defining for $P(G)$ if k is odd.

Proof. For 1.) and the validity of the lifted hole inequality see Chopra and Rao [1994a], Lemma 5.2.1 and Lemma 5.2.2.

To show the facet-property of the lifted odd-hole inequality, consider any valid inequality for $P(G)$
$$a^T x \geq b$$
such that
$$F := \{x \in P(G) : \sum_{e \in E_k} x_e + 2 \sum_{e \in E \setminus E_k} x_e = 2(k-1)\} \quad \subseteq \quad \{x \in P(G) : \ a^T x = b\}.$$
We show that $a^T x \geq b$ is a nonnegative multiple of the considered constraint.

The proof of Theorem 5.2.1 in Chopra and Rao [1994a] shows that $a_e = a_f$ for $e, f \in E_k$.

We show that $a_e = a_f$ for $e, f \in E \setminus E_k$. Consider the following binary Steiner trees T_f for all $f = \{t_k, v\}$ with $v \neq s_k, s_{k-1}$
$$T_f := E(C_k^1) \setminus \{\{t_{k-1}, s_{k-1}\}, \{s_{k-1}, t_k\}, \{t_k, s_k\}, \{s_k, t_1\}\} \cup \{f\}.$$
$T_f \in F$ and T_{f_1} and T_{f_2} only differ in f_1, f_2, compare Figure 6. Hence, $a_{f_1} = a_{f_2}$ and therefore $a_e = a_f$ for $e, f = \{t_k, v\}$ with $v \neq s_k, s_{k-1}$. Instead of removing edges $\{t_{k-1}, s_{k-1}\}$, $\{s_{k-1}, t_k\}$, $\{t_k, s_k\}$, $\{s_k, t_1\}$ and adding one edge incident to t_k, we do the same procedure by choosing $i \in \{1, \dots, k-1\}$ instead of k, i.e. removing the edges $\{t_{i-1}, s_{i-1}\}$, $\{s_{i-1}, t_i\}$, $\{t_i, s_i\}$, $\{s_i, t_{i+1}\}$ and adding one edge incident to t_i. This shows that $a_e = a_f$ for $e, f \in E \setminus E_k$.

Finally, consider $T_{\bar{f}}$ for one fixed $\bar{f}$ as above and
$$\tilde{T} = E(C_k^1) \setminus \{e_1 = \{t_k, s_k\}, e_2 = \{s_k, t_1\}\}.$$
$\tilde{T}$ and $T_{\bar{f}}$ are in F and differ only in $\bar{f}$ vs. e_1, e_2. Therefore, $a_{\bar{f}} = a_{e_1} + a_{e_2} = 2a_e$ for $e \in E_k$. It follows that $2a_e = a_f$ for all $e \in E_k$, $f \in E \setminus E_k$.

In conclusion, $a^T x \leq b$ is a nonnegative multiple of the odd-hole constraint. This implies that this constraint is facet-defining for $P(G)$.

□

In the case of an even value of k, one can show that the above inequalities are the sum of two partition inequalities.

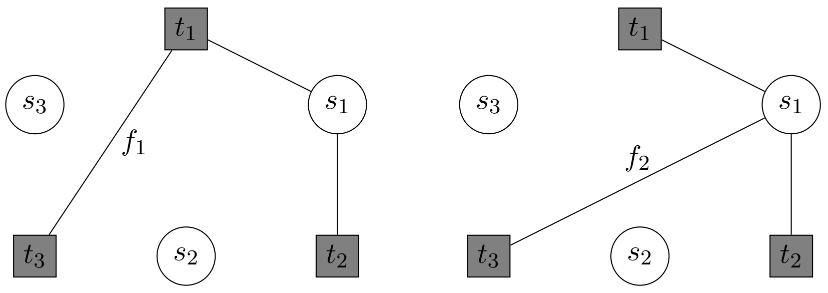


Figure 6: **Lifted odd-hole inequalities:** The trees T_{f_1} and T_{f_2} introduced in the proof of Theorem 5.7 differ only in f_1 and f_2.

5.2.3 Facets Coming from the 3-Matching Polytope

We now identify some classes of facets that stem from the 3-matching polytope. These inequalities are called blossom inequalities. Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. A blossom inequality is of the form

$$x(E(H)) + x(J) \leq \left\lfloor \frac{1}{2}(3|H| + |J|) \right\rfloor, \quad \forall\ H \subseteq V,\ J \subseteq \delta(H).$$

Not all of the blossom inequalities define facets of $P(G)$. However, we have identified some exponentially large sets of blossom facets. Here, we restrict ourselves to the following two sets:

Theorem 5.8 (Blossom Facets)**.** *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. Let $\bar{S} \subseteq S$ be a (maybe empty) subset of the Steiner nodes and $\bar{G} = (\bar{V} = T \cup \bar{S}, \bar{E})$ the complete graph on $T \cup \bar{S}$.*

a) For $m = |\bar{V}|$ odd, the following inequality is valid and facet-defining for $P(G)$:

$$x(\bar{E}) \leq \left\lfloor \frac{3}{2} m \right\rfloor.$$

b) For $m = |\bar{V}|$ even, $H \subseteq \bar{V}$ with $|H| = |\bar{V}| - 1$ and $J \subseteq \delta(H) \cap \bar{E}$, $|J| = 2$ the following inequality is valid and facet-defining for $P(G)$:

$$x(\bar{E}(H)) + x(J) \leq \left\lfloor \frac{1}{2}(3m - 1) \right\rfloor.$$

Proof. The inequalities are valid for $P(G)$ since they are blossom inequalities which are valid for the 3-matching-polytope. The facet-defining property can be again shown in an indirect way. We construct subgraphs that lie within the facet and contain exactly one "free" node of degree two. The whole proof can be found in the forthcoming PhD-Thesis of Pape [2015]. □

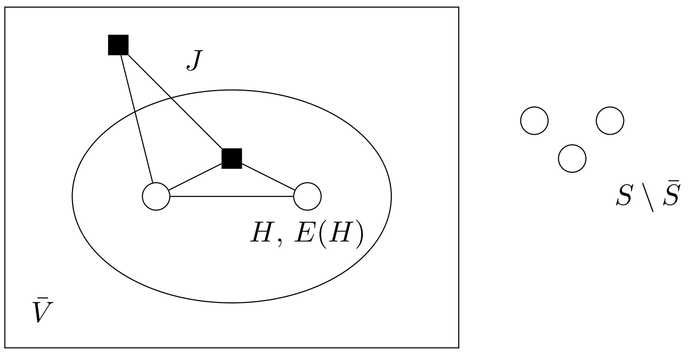


Figure 7: **A blossom facet of Theorem 5.8 with** $m = 4$**.**

We have identified many more classes of facets induced by blossom inequalities. However, blossom facets do not have a big influence if we are interested in optimal binary Steiner trees with a nonnegative objective function. Indeed, we can show that many blossom inequalities, especially the ones that we have identified to induce facets, do not lie in the direction of a nonnegative objective function. That is equivalent to the fact that all incidence vectors lying within these facets belong to non-cyclefree binary Steiner trees.

Theorem 5.9 (Steiner Circles). *For $H \subseteq V$, $J \subseteq \delta(H)$ let*

$$x(E(H)) + x(J) \leq \left\lfloor \frac{1}{2}(3|H| + |J|) \right\rfloor \quad \text{with } (3|H| + |J|) \text{odd},$$

be a blossom inequality for a binary Steiner polytope $P(G)$. Let $|H| \geq |V_J| := |V(J) \setminus H|$ and let x be an incidence vector of a binary Steiner tree T' that fulfills the inequality with equality, i.e. that lies within the hyperplane of the blossom inequality. Then, T' possesses a circle.

Proof. Let x be an incidence vector of a binary Steiner tree T' that fulfills the blossom inequality with equality. Then

$$\begin{aligned} x(E(H)) + x(J) &= \left\lfloor \tfrac{1}{2}(3|H| + |J|) \right\rfloor \\ &= \tfrac{1}{2}(3|H| + |J| - 1) \\ &\geq \tfrac{1}{2}(3|H| + |V_J| - 1) \\ &= |H| + \tfrac{1}{2}|H| + \tfrac{1}{2}|V_J| - \tfrac{1}{2} \\ &\geq |H| + \tfrac{1}{2}|V_J| + \tfrac{1}{2}|V_J| - \tfrac{1}{2} \\ &= |H| + |V_J| - \tfrac{1}{2} \end{aligned}$$

Since $x(E(H)) + x(J) \in \mathbb{N}$ and $|H| + |V_J| \in \mathbb{N}$ it follows that

$$x(E(H)) + x(J) \geq |H| + |V_J|.$$

Since each graph $G = (V, E)$ with $|E| \geq |V|$ possesses a circle we conclude that T' possesses a circle. □

The inequalities of Theorem 5.8 obviously fulfill the conditions of Theorem 5.9. In fact, we have not found any class of blossom facets that does not fulfill these conditions.

5.2.4 A New Class of Inequalities and Facets

In this section we introduce a new class of facets that does not originate from the Steiner tree polytope or the matching polytope. For these inequalities we choose two certain edge sets $E_1, E_2 \in E$ with $E_1 \cap E_2 = \emptyset$. The inequalities state that if a certain number of edges of E_1 are contained in a binary Steiner tree, then there need to be a special number of edges of E_2 in the binary Steiner tree as well.

We denote with $K_{|W|}(W)$ the complete graph of all nodes $v \in W$, $W \subseteq V$. Further, for a subgraph $F \subseteq G$ we define the degree of freedom $\text{free}_F(v)$ of node v in F as

$$\text{free}_F(v) = \begin{cases} 0 & \text{if } \deg_F(v) \geq 3 \\ 3 - \deg_F(v) & \text{otherwise} \end{cases}$$

Theorem 5.10 (Combined Inequalities). *Let $G = K_n = (V, E)$ with $n \geq 5$ and $T \subseteq V$. Further, let $V = V_1 \cup V_2$ be a partition of the nodes with $V_1 \cap V_2 = \emptyset$, $V_i \cap T \neq \emptyset$ for $i = 1, 2$. We write $V_1 := \{v_1, \ldots, v_N\}$, $T \cap V_2 := \{t_1, \ldots, t_K\}$ and $S \cap V_2 := \{s_1, \ldots, s_M\}$. We construct two edge sets $E_1, E_2 \subseteq E$ as follows:*

- *Let E_1 consist of edges incident only to nodes in V_1.*

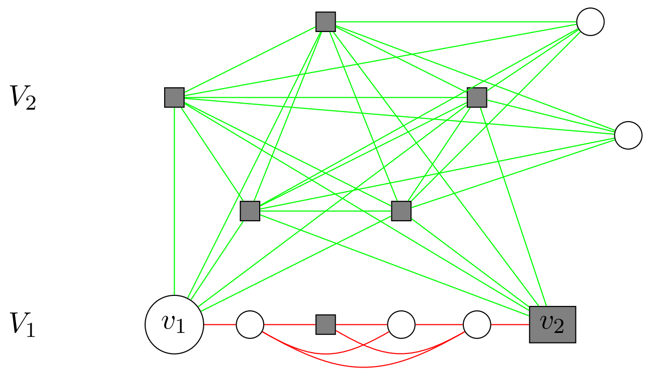


Figure 8: **Combined Inequalities with $L = 2$ special nodes in V_1 :** The red edges are within E_1, the green edges belong to E_2. Since V_1 consists of $N = 6$ nodes, and V_2 contains $K = 5$ terminal nodes and $M = 2$ Steiner nodes, the corresponding combined inequalities reads as $x(E_1) - x(E_2) \leq 4$.

- *Let $L \in \{1, \ldots, N\}$ and*

$$\begin{aligned} E_2 \quad := \quad & K_K(T \cap V_2) \cup \{\{t_i, v_j\} : \ i = 1 \ldots, K, \ j = 1, \ldots, L\} \\ & \cup \{\{s_l, t_i\} : l = 1, \ldots, M, \ i = 1 \ldots, K\}. \end{aligned}$$

Let $D := \sum_{j=1}^{L} \deg_{E_1}(v_j)$. If both $(3N - K)$ and $(3L - D)$ are odd, then the inequality

$$\sum_{e \in E_1} x_e \ - \sum_{e \in E_2} x_e \ \leq \ \left\lfloor \frac{1}{2}(3N - K) \right\rfloor - \left\lfloor \frac{1}{2}(3L - D) \right\rfloor .$$

is valid for $P(G)$. Otherwise, the inequality

$$\sum_{e \in E_1} x_e \ - \sum_{e \in E_2} x_e \ \leq \ \left\lfloor \frac{1}{2}(3N - K) \right\rfloor - \left\lceil \frac{1}{2}(3L - D) \right\rceil .$$

is valid for $P(G)$.

Proof. Let first $(3N - K)$ and $(3L - D)$ not both be odd. Let x be an incidence vector of a binary Steiner tree T' in G. If $x(E_1) \leq \lfloor \frac{1}{2}(3N - K) \rfloor - \lceil \frac{1}{2}(3L - D) \rceil$, the combined inequality is trivially fulfilled. Hence, assume that

$$x(E_1) = \left\lfloor \frac{1}{2}(3N - K) \right\rfloor - \left\lceil \frac{1}{2}(3L - D) \right\rceil + q \ \text{ for } q = 1, 2, \ldots \qquad (*)$$

Then

$$\begin{aligned} \textstyle\sum_{v \in V_1} \deg_{T'(V_1)}(v) \quad &= \quad \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + \sum_{j=L+1}^{N} \deg_{T'(V_1)}(v_j) \\ &= \quad \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + \sum_{j=L+1}^{N} \left(3 - \text{free}_{T'(V_1)}(v_j)\right) \\ &= \quad \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + 3(N - L) - \sum_{j=L+1}^{N} \text{free}_{T'(V_1)}(v_j) \end{aligned}$$

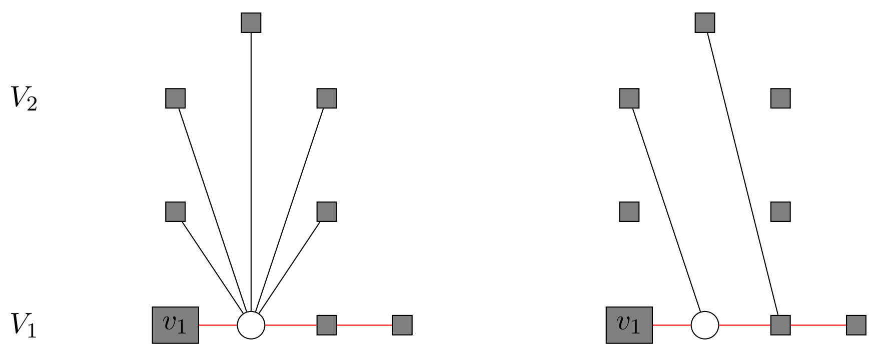


Figure 9: **Combined Inequalities are not valid for the Steiner tree polytope and the 3-matching polytope:** Let V_1 consists of $N = 4$ nodes with $L = D = 1$ special node. Let E_1 be complete on V_1. With $K = 5$ terminal nodes and $M = 0$ Steiner nodes in V_2, the combined inequality reads as $x(E_1) - x(E_2) \leq 2$. Hence, the Steiner tree (left) and the 3-matching (right) do not fulfill the inequality.

Hence

$$\begin{aligned}\textstyle\sum_{j=L+1}^{N} \mathrm{free}_{T'(V_1)}(v_j) &= \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + 3(N-L) - \sum_{v \in V_1} \deg_{T'(V_1)}(v) \\ &= \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + 3(N-L) - 2x(E_1) \\ &\overset{(*)}{=} \textstyle\sum_{j=1}^{L} \deg_{T'(V_1)}(v_j) + 3N - 3L \\ &\quad -2(\lfloor \frac{1}{2}(3N-K) \rfloor - \lceil \frac{1}{2}(3L-D) \rceil + q) \\ &\leq D + 3N - 3L - 2(\lfloor \frac{1}{2}(3N-K) \rfloor - \lceil \frac{1}{2}(3L-D) \rceil + q)\end{aligned}$$

because of the definition of D. If $(3N - K)$ and $(3L - D)$ are both even, this amounts to

$$\begin{aligned}\textstyle\sum_{j=L+1}^{N} \mathrm{free}_{T'(V_1)}(v_j) &\leq D + 3N - 3L - 3N + K + 3L - D - 2q \\ &= K - 2q.\end{aligned}$$

However, we need to connect at least K terminals of V_2 to the nodes in V_1 by the definition of K. Because of the above calculation this is only possible using at least $K - (K - 2q) = 2q$ edges in E_2. Hence

$$\begin{aligned}x(E_1) - x(E_2) &\leq \lfloor \tfrac{1}{2}(3N-K) \rfloor - \lceil \tfrac{1}{2}(3L-D) \rceil + q - 2q \\ &= \lfloor \tfrac{1}{2}(3N-K) \rfloor - \lceil \tfrac{1}{2}(3L-D) \rceil - q,\end{aligned}$$

i.e. the combined inequality is fulfilled.

If one term of $(3N - K)$ and $(3L - D)$ is even and the other one is odd, one can show with similar arguments that

$$x(E_1) - x(E_2) \leq \lfloor \tfrac{1}{2}(3N-K) \rfloor - \lceil \tfrac{1}{2}(3L-D) \rceil - q + 1,$$

i.e. the combined inequality is fulfilled.

The case with both $(3N - K)$ and $(3L - D)$ being odd can be proven in a similar way. □

An illustration of this class of inequalities is given in Figure 8. Obviously, the combined inequalities are not valid for the Steiner tree polytope without degree constraints and the 3-matching polytope, compare Figure 9.
Some of these inequalities induce facets for $P(G)$:

Theorem 5.11 (Combined Facets). *Let E_1 and E_2 be defined as in Theorem 5.10 together with the following additional requirements:*

(1.) *E_1 is complete on $\{v_{L+1}, \dots, v_N\}$.*

(2.) *$L = 1$ and $\deg_{E_1}(v_1) = 2$*

or

$L \geq 2$. Each node in $\{v_1, \dots, v_L\}$ has degree one or two in E_1 and is incident only to nodes in $\{v_{L+1}, \dots, v_N\}$. Two nodes of $\{v_{L+1}, \dots, v_N\}$, let's say v_{L+1}, v_N are incident to at most two nodes in $\{v_1, \dots, v_L\}$ and each node in $\{v_{L+2}, \dots, v_{N-1}\}$ is incident to at most one node in $\{v_1, \dots, v_L\}$.

(3.) *$K = N - (L + D) + 3$.*

Then one term of $(3N - K) = (2N + L + D - 3)$ and $(3L - D)$ is even and the other one is odd. Further, the inequality

$$\begin{array}{rcl} x(E_1) - x(E_2) & \leq & \left\lfloor \frac{1}{2}(3N - K) \right\rfloor - \left\lceil \frac{1}{2}(3L - D) \right\rceil \\ & = & \left\lfloor \frac{1}{2}(2N + L + D - 3) \right\rfloor - \left\lceil \frac{1}{2}(3L - D) \right\rceil \\ & = & N + D - L - 2 \end{array}$$

induces a facet for $P(G)$.

Proof. The fact that one term of $(2N + L + D - 3)$ and $(3L - D)$ is even and the other one is odd, can be easily verified by considering the four combinations of D, L being odd and/or even. To show that the given inequality is facet-defining for $P(G)$, consider any valid inequality for $P(G)$

$$a^T x \leq b$$

such that

$$F := \{x \in P(G) : \sum_{e \in E_1} x_e \; - \sum_{e \in E_2} x_e \; = \; N + D - L - 2\} \; \subseteq \; \{x \in P(G) : \; a^T x = b\}.$$

We show that $a^T x \leq b$ is a nonnegative multiple of the given constraint.
In the case $L = 1$ let v_1 be incident to v_2 and v_N. Otherwise let $\{v_{L+1}, \dots, v_N\}$ be ordered such that v_1 is incident to v_{L+1} and v_L is incident to v_N. We introduce the following sets of edges: Let $P = (v_{L+1}, v_{L+2}, \dots, v_N)$ be a path through edges in E_1. Such a path exists because of assumption (1.). Let $\tilde{P}$ be a path from v_1 to v_L passing all nodes v_j in $\{v_2, \dots, v_{L-1}\}$ that have $\deg_{E_1}(v_j) = 1$. Because of assumption (2.) it follows that $\tilde{P} \cap \{E_1, E_2\} = \emptyset$. We define

$$P' = P \cup \tilde{P} \cup \{\delta(\{v_1, \dots, v_L\}) \cap E_1\},$$

compare Figure 10. In P' all nodes of V_1 are connected and fulfill the degree requirements because of assumption (2.). Further, we define

$$P_j = P' \setminus \{\{v_j, v_{j+1}\}\} \quad \text{for } j = L + 1, \dots, N - 1.$$

Since $\tilde{P} \cap E_1 = \emptyset$ it follows that $|P_j \cap E_1| = |P'| + D - 1 = (N - L - 1) + D - 1 = N + D - L - 2$. Further, the nodes in $\{v_{L+1}, \dots, v_N\}$ have a total degree of freedom

$$\sum_{h=L+1}^{N} \text{free}_{P_j}(v_h) = (N - L - 4) \cdot 1 + 4 \cdot 2 - D = N - (L + D) + 4 \overset{(3.)}{=} K + 1.$$

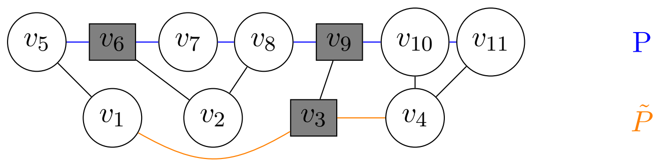


Figure 10: **Path P' introduced in the proof of Theorem 5.11 for $N = 11$ and $L = 4$:** P' consists of the path P (blue), the path $\tilde{P}$ (orange) and the edges in $\delta(\{v_1, \dots, v_L\}) \cap E_1$. The path $\tilde{P}$ starts in v_1, passes all nodes in $\{v_2, \dots, v_{L-1}\}$ that have $\deg_{E_1}(v_j) = 1$ and ends in v_L.

Hence, it is possible to connect the terminal nodes in V_2 directly to nodes in $\{v_{L+1}, \dots, v_N\}$, i.e. without using any edges of E_2 and without violating the degree constraints. Let π_j such a valid assignment of $T \cap V_2$ to P_j and T_{π_j} the corresponding tree, i.e.

$$T_{\pi_j} := P_j \cup \{e : \ e \in \pi_j\}.$$

Then $T_{\pi_j} \in F$. Further, we denote with Π_j the set of all valid assignments of $T \cap V_2$ to P_j. Since the total degree of freedom in P_j is $K+1$, there do exist $\pi_j^1, \pi_j^2 \in \Pi_j$ such that $\{t_K, v_j\} \in T_{\pi_j^1}$ and $\{t_K, v_{j+1}\} \in T_{\pi_j^2}$ and $T_{\pi_j^1} \setminus \{\{t_K, v_j\}\} = T_{\pi_j^2} \setminus \{\{t_K, v_{j+1}\}\}$, compare Figure 11. In other words we have

$$\deg_{T_{\pi_j^1}}(v_{j+1}) = 2 \text{ and } \deg_{T_{\pi_j^2}}(v_j) = 2.$$

Further, in

$$P'' = P' \setminus \tilde{P}$$

the nodes $v_{L+1}, \dots, v_N$ have a total degree of freedom of $K - 1$. Let π^* be an assignment of $t_1, \dots, t_{k-1}$ to P'' and T_{π^*} the corresponding graph. Again, we denote with Π^* the set of all such valid assignments. Then, the following graphs (compare Figure 12)

$$\begin{array}{lcll}
T_{\pi^*}^{v_j} & = & T_{\pi^*} \cup \{\{t_K, v_j\}\} & \text{for } j = 1, \dots, L, \\
T_{\pi^*}^{t_i} & = & T_{\pi^*} \cup \{\{t_K, t_i\}\} & \text{for } i = 1, \dots, K-1, \\
T_{\pi^*}^{s_l, v_j} & = & T_{\pi^*} \cup \{\{t_K, s_l\}\} \cup \{\{s_l, v_j\}\} & \text{for } l = 1, \dots, M, \ j = 1, \dots, L,
\end{array}$$

contain exactly one edge of E_2 and $|E(T_{\pi^*}) \cap E_1| = (N - L - 1) + D$. Because of assumption (2.) these graphs are connected and fulfill the degree constraints. Hence, they are in F.

We show that $a_e = 0$ for all $e \notin E_1 \cup E_2$. First, let $e = \{t_i, v_j\}$ for $i = 1, \dots, K$ and $j = L+1, \dots, N$. Consider an assignment $\pi_h \in \Pi_h$ for arbitrary $h \in \{L+1, \dots, N-1\}$ such that v_j has a degree of two in T_{π_h}. Such an assignment obviously exists since the total degree of freedom in P_h is $K+1$, hence after adding K terminal nodes, the degree of freedom is still one in T_{π_h}. Further, $T_{\pi_h} \cup \{e\} \in F$. Hence, $a_e = 0$.
For $e = \{s_l, v_j\}$ with $l = 1, \dots, M$ and $j = 1, \dots, N$ we can do the same. (Note that $v_1, \dots, v_L$ have at most a degree of two by assumption (2.).)
For $e = \{s_{l_1}, v_{l_2}\}$ with $l_1, l_2 = 1, \dots, M$ we can add e to any described tree.
Finally, let $e = \{v_{j_1}, v_{j_2}\}$ for $j_1, j_2 = 1, \dots, L$. Then $T_{\pi^*}^{t_i} \cup \{e\} \in F$ for arbitrary $i \in \{1, \dots, K-1\}$. Hence $a_e = 0$ for all $e \notin E_1 \cup E_2$.

We show that $a_e = a_f$ for all $e, f \in E_1$. For $j = L+1, \dots, N-1$ consider $\pi_j^1 \in \Pi_j$ and $T_{\pi_j^1}$ as explained above. Then $T_{\pi_j^1} \cup \{\{v_j, v_{j+1}\}\} \setminus \{g\}$ for $g = \{v_{j-1}, v_j\}$ or $g \in \delta(v_j) \cap E_1$ is binary and connected by construction. Further, it is element of F. Hence, $a_g = a_{v_j, v_{j+1}}$. We can do the same with π_j^2. Because of assumption (1.) we can use different permutations of the nodes

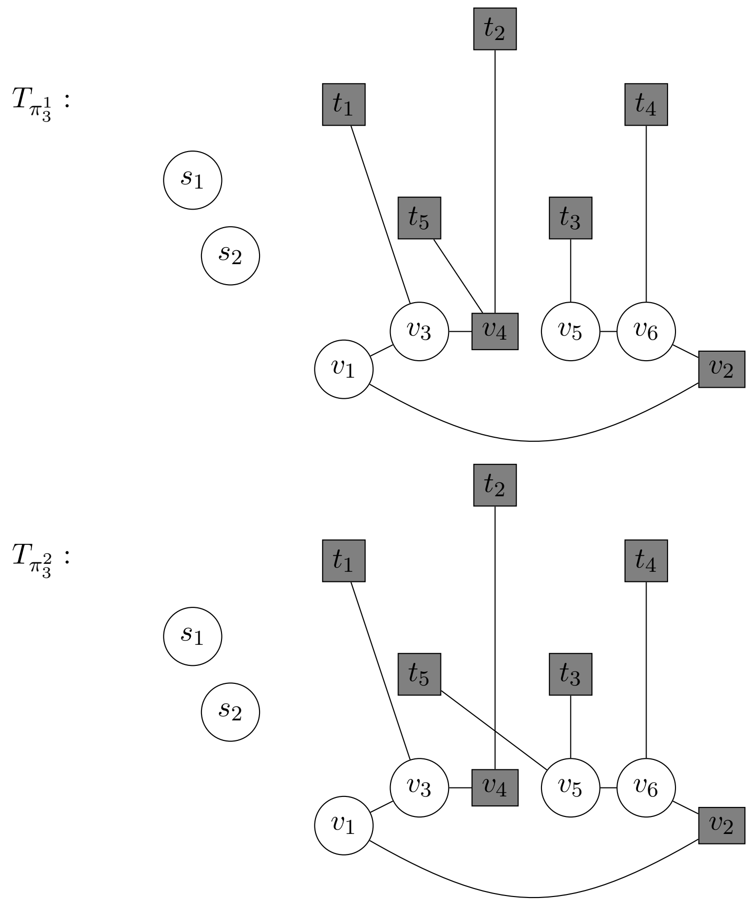


Figure 11: **Combined facets for $N = 6$, $L = D = 2$, $K = 5$ and $M = 2$:** The trees $T_{\pi_j^1}$ and $T_{\pi_j^2}$ introduced in the proof of Theorem 5.11 for $j = 3$.

$v_{L+2}, \ldots, v_{N-1}$ to conclude that $a_e = a_f$ for all $e, f \in E_1 \setminus \{\{v_{L+1}, v_N\}\}$. For $e = \{v_{L+1}, v_N\}$ consider $\pi_1^2 \in \Pi_1$ and $T_{\pi_1^2} \cup \{\{v_{L+1}, v_N\}\} \setminus \{\{v_{N-1}, v_N\}\} \in F$. In conclusion we get $a_e = a_f$ for all $e, f \in E_1$.

We show that $a_e = a_f$ for all $e, f \in E_2$. Let $\pi^* \in \Pi^*$. We compare the graphs $T_{\pi^*}^{v_j}$, $T_{\pi^*}^{t_i}$ and $T_{\pi^*}^{s_l,v_j}$:

$$\begin{array}{rcll} T_{\pi^*}^{v_1} & = & T_{\pi^*}^{v_j} \setminus \{\{t_K, v_j\}\} \cup \{\{t_K, v_1\}\} & \text{for } j = 2, \ldots, L \\ & = & T_{\pi^*}^{t_i} \setminus \{\{t_K, t_i\}\} \cup \{\{t_K, v_1\}\} & \text{for } i = 1, \ldots, K \\ & = & T_{\pi^*}^{s_l,v_j} \setminus \{\{t_K, s_l\}, \{s_l, v_1\}\} \cup \{t_K, v_1\} & \text{for } l = 1, \ldots, M. \end{array}$$

Note that $a_{s_l,v_1} = 0$. Hence, it follows that $a_e = a_f$ for $e, f \in E_2$ and e, f incident to t_K. Defining $\Pi^*(t_K) = \Pi^*$ and $\Pi^*(t_i)$ in a similar way for $i = 1, \ldots, K-1$, we get that $a_e = a_f$ for all $e, f \in E_2$.

Finally, we show that $a_e = -a_f$ for all $e \in E_1$ and $f \in E_2$. Let $\pi^* \in \Pi^*$. Then

$$T_{\pi^*}^{t_1} \cup \tilde{P} \cup \{\{t_K, v_{L+1}\}\} \setminus \{\{t_K, t_1\}, \{v_{L+1}, v_{L+2}\}\} = T_{\pi_1}$$

for suitable $\pi_1 \in \Pi_1$. Hence, $0 = (\sum_{v \in \tilde{P}} a_v) + a_{t_K,v_{L+1}} = a_{t_K,t_1} + a_{v_{L+1},v_{L+2}}$ and therefore $a_{t_K,t_1} = -a_{v_{L+1},v_{L+2}}$. Since $\{t_K, t_1\} \in E_2$ and $\{v_{L+1}, v_{L+2}\} \in E_1$ and because of the fact that $a_e = a_f$ for e, f both being element of E_1 or both being in element of E_2 we get that $a_e = -a_f$ for all $e \in E_1$ and $f \in E_2$.

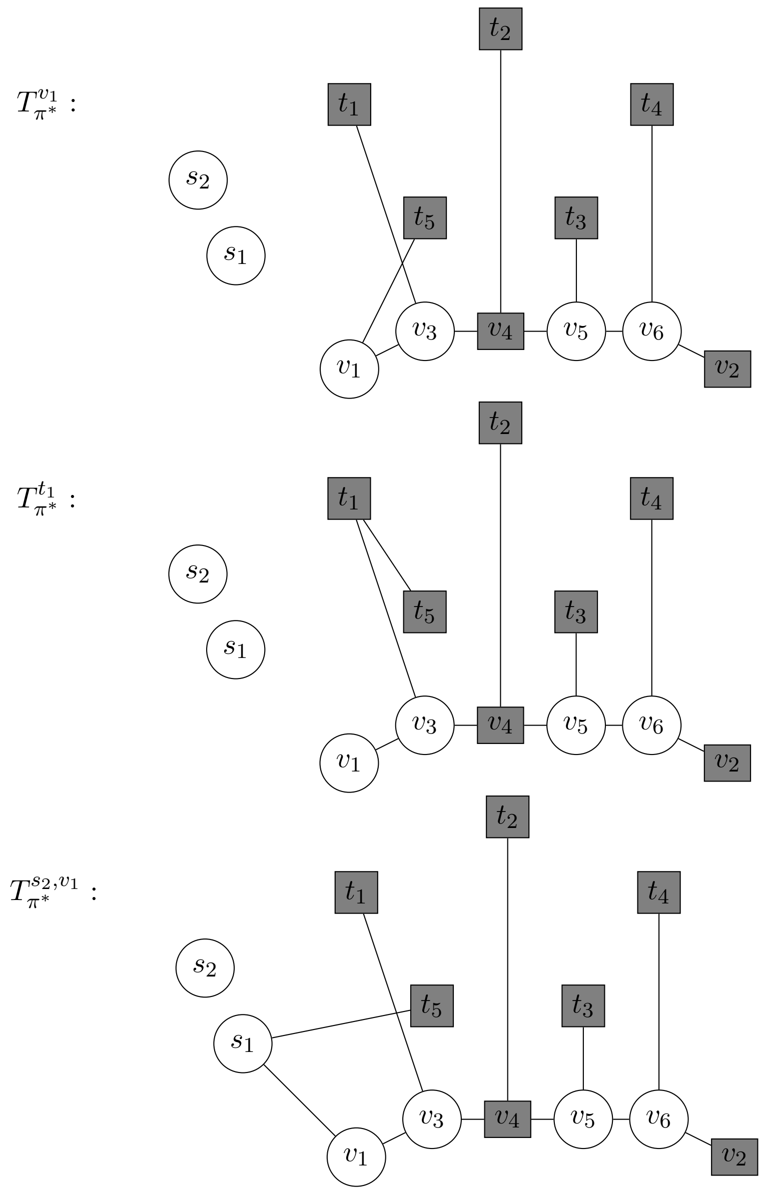

$T^{v_1}_{\pi^*}$:
t_1 t_2 t_3 t_4 t_5
s_1 s_2
v_1 v_2 v_3 v_4 v_5 v_6
$T^{t_1}_{\pi^*}$:
t_1 t_2 t_3 t_4 t_5
s_1 s_2
v_1 v_2 v_3 v_4 v_5 v_6
$T^{s_2,v_1}_{\pi^*}$:
t_1 t_2 t_3 t_4 t_5
s_1 s_2
v_1 v_2 v_3 v_4 v_5 v_6

Figure 12: **Combined facets for $N = 6$, $L = D = 2$, $K = 5$ and $M = 2$:** The trees $T^{v_1}_{\pi^*}$, $T^{t_1}_{\pi^*}$ and $T^{s_2,v_1}_{\pi^*}$ introduced in the proof of Theorem 5.11.

In conclusion, $a^T x \leq b$ is a nonnegative multiple of the combined facet. This implies that this constraint is facet-defining for $P(G)$. $\square$

Note that the class of facets of Theorem 5.11 has cardinality that grows exponentially in the number of terminal nodes: Adding two terminal nodes to a graph G leads to a duplication of its facets introduced by Theorem 5.11 since a new terminal node can be put either in V_1 or V_2.

5.3 The Directed Binary Steiner Tree Polytope

We now turn to the directed binary Steiner tree polytope based on IP-model 2 and extend the results from the undirected polytope to the directed version. The directed binary Steiner tree polytope is defined similar to the undirected case.

Definition 5.12 (Directed Binary Steiner Tree Polytope)**.**

Let $D = (V, A)$ be a directed graph with terminal set $T \subseteq V$ and $s \in S$. We denote with

$$\begin{aligned} P^D(D) &:= \operatorname{conv}\{x \in \{0,1\}^{|A|} : \textit{ x is incidence vector of a } bST^D \textit{ of } G\} \\ &= \operatorname{conv}\{x \in \{0,1\}^{|A|} : \textstyle\sum_{a \in \delta^+(W)} x_a \geq 1 \;\forall\; W \subseteq V,\; s \in W, \\ &\qquad (V \setminus W) \cap S \neq \emptyset,\; \textstyle\sum_{a \in \delta(v)} x_a \leq 3 \;\forall\; v \in V, \\ &\qquad x_{ij} + x_{ji} \leq 1 \;\forall\; (i,j),(j,i) \in A\} \end{aligned}$$

the ***directed binary Steiner tree polytope*** *of D.*

Again, we consider directed complete graphs $\vec{K}_n$ for $n \geq 5$. Since a directed binary Steiner tree contains a directed path from the root s to each terminal node, it is possible to restrict our problem to graphs $\tilde{K}_n = \vec{K}_n \setminus \{(v, s) \in A : v \in V\}$.

Similar to the undirected polytope we can show that $P^D(\tilde{K}_n)$ is full-dimensional and that the bound constraints and degree constraints define facets. In case of partition inequalities it can be shown that the cut inequalities define again facets while all other partition inequalities can be written as a sum of $k-1$ cut inequalities. Further, the odd-hole inequalities do not define facets in directed graphs since they are the sum of $2(k-1)$ cut inequalities and some trivial inequalities. In the case of directed blossom inequalities one can show that Theorem 5.8, item a) still holds in directed graphs. Item b) stills holds if we choose $s \notin H$, $J \subseteq \delta^-(H)$ or $s \in H$, $J \subseteq \delta^+(H)$. The proofs are very similar to Theorem 5.8 with suitable orientations of the arcs.
The combined inequalities of Theorem 5.10 are also valid for the directed case. They define facets in $P^D(D)$ in the following way:

Theorem 5.13 (Directed Combined Facets)**.** *Let $D = \tilde{K}_n = (V, A)$, $n \geq 5$, $T \subseteq V$. Further, let $V = V_1 \cup V_2$ be a partition of the nodes with $V_1 \cap V_2 = \emptyset$, $V_i \cap T \neq \emptyset$ for $i = 1, 2$ and $s \in V_1$. We denote $V_1 := \{v_1, \ldots, v_N\}$, $T \cap V_2 := \{t_1, \ldots, t_K\}$ and $S \cap V_2 := \{s_1, \ldots, s_M\}$. Let $L \geq 2$. We construct two arc sets $A_1, A_2 \subseteq A$ as follows:*

- *Let A_1 be complete on $\{v_{L+1}, \ldots, v_N\}$. Connect v_1, v_L to a node (not the same node) in $\{v_{L+1}, \ldots, v_N\}$, let's say to v_{L+1} and v_N, respectively, by two antiparallel arcs. Each remaining nodes $v_2, \ldots, v_{L-1}$ are connected to at most two nodes in $\{v_{L+2}, \ldots, v_{N-1}\}$ such that each node $v_2, \ldots, v_{L-1}$ has at least one ingoing arc and each node in $\{v_{L+2}, \ldots, v_{N-1}\}$ has at most one arc incident to a node in $v_2, \ldots, v_{L-1}$.*
- *Further, we define A_2 as*

$$\begin{aligned} A_2 &:= \vec{K}_K(T \cap V_2) \cup \{(v_j, t_i) : \; j = 1, \ldots, L, \; i = 1 \ldots, K, \} \\ &\quad \cup \{(s_l, t_i) : l = 1, \ldots, M, \; i = 1 \ldots, K\}. \end{aligned}$$

Let $D := \sum_{j=1}^{L} \deg_{A_1}(v_j)$. For $K = N - (L + D) + 3$ the inequality

$$\sum_{a \in A_1} x_a \; - \; \sum_{a \in A_2} x_a \; \leq \; N + D - L - 2.$$

induces a facet for $P^D(D)$.

The proof is similar to the undirected version. More details can be found in the forthcoming PhD-Thesis of Pape [2015].

5.4 Separation Routines

Let a solution x of a initial LP relaxation only including a subset of the required constraints be given. It is likely that x does not satisfy all constraints or identified valid inequalities of the

model. The separation problem is to identify violated inequalities.
The number of trivial inequalities $0 \leq x \leq 1$ and degree constraints $\sum_{e \in \delta(v)} x_e \leq 3$, $v \in V$ is $2 \cdot |E| + |V|$, hence all inequalities can potentially be checked in polynomial time. Though, in the branch-&-cut approach the first LP is set up consisting of these inequalities. Hence, we do not need to separate the bounds and the degree constraints.
For the separation of partition inequalities, we distinguish two cases: Cut inequalities and general partition inequalities. The separation problem for cut inequalities can be solved in polynomial time with max-flow-min-cut algorithms. In contrast, Grötschel et al. [1992] showed that the separation problem for partition inequalities is $\mathcal{NP}$-hard in general. The separation of odd-hole inequalities is also $\mathcal{NP}$-hard. However, since for directed graphs partitions P with $|P| \geq 3$ and odd-holes do not induce facets, we restrict ourselves to the separation of cut inequalities.
Letchford et al. [2008] give a polynomial time separation algorithm for blossom inequalities. Since all binary Steiner trees of the introduced blossom facets contain cycles, we do not investigate in any special separation routines for blossom facets.
The complexity of separating the combined inequality is open. We conjecture that it is $\mathcal{NP}$-hard. We therefore propose a simple heuristic that uses the fact that the separation of cut inequalities already gives some minimal cuts. For the two node sets V_1 and V_2 induced by a minimal cut we construct the corresponding edge sets E_1 and E_2 as explained in Theorem 5.11. The special nodes $v_1 \ldots, v_L$ are chosen in a greedy manner. We test if the associated combined inequality is violated. We do not check the condition whether $K = N - (L + D) + 3$. Thus, we might separate inequalities that are not facets.

6 A Primal Heuristic for Binary Steiner Trees

We now present a simple algorithm to compute feasible binary Steiner trees based on a approach of Takahashi and Matsuyama [1979]. It is used in the branch-&-cut approach to compute upper bounds. It is based on undirected graphs. The idea of this heuristic is to start with one terminal node and connect another terminal that is closest to the starting terminal by a shortest path while respecting the degree restriction. At each next step of this procedure a new terminal that is closest to the existing path or tree is inserted. In detail, the adapted TM-algorithm looks as follows:

Input: Undirected complete weighted graph $G = (V, E, c)$ with terminal set $T \subseteq V$ and $c \geq 0$.

Output: Binary Steiner B.

(1.) Choose $v_1 \in T$ and set $V(B) = v_1$ and $E(B) = \emptyset$.

(2.) For $i = 2, \ldots, |T|$ do: Let $B_i = B \setminus \{v \in B : \deg(v) = 3\}$. For $v \in T \setminus V(B)$ and $w \in V(B_i)$ let $P(v, w)$ be a shortest path between v and w in $G \setminus B_i$. Set $P_i = \min\{P(v, w) : v \in T \setminus V(B), w \in V(B_i)\}$. Set $V(B) = V(B) \cup V(P_i)$ and $E(B) = E(B) \cup \{P_i\}$.

Since our implementation of the branch-&-cut algorithm is based on directed graphs, we transform the resulted binary Steiner tree B into a rooted tree. In step (2) if P_i contains more than one minimal path, we choose the one with maximal number of edges. Obviously, this heuristic computes a binary Steiner tree in complete graphs in polynomial time. However, it may fail in non-complete graphs which is not surprising since we have shown that the problem of finding a binary Steiner tree is $\mathcal{NP}$-complete.
We have also experimented with a modification of the approach of Rayward-Smith [1983]. The computed solutions values are quite good. However, the running times are significant slower, especially for large instances.

7 Structural Results for the General Degree-Constrained Steiner Tree Problem

We now generalize the previous results to the Steiner tree problem with general degree constraints. Let $G = (V, E)$ be an undirected graph with terminal set $T \subseteq V$ and let $b(v) \in \mathbb{N}$ for $v \in V$. The task is to construct a (minimum) Steiner tree such that the degree at each node $v \in V$ is less than or equal to $b(v)$. If $b(v_j) = 0$ for some terminal node $v_j \in T$, there does not exist a Steiner tree satisfying the degree constraint. If $b(v_j) = 0$ for some Steiner node $v_j \in S$, the node v_j and its incident edges can be deleted. Hence we restrict to the case $b(v) > 0$ for all $v \in V$. Since binary Steiner trees and degree-constrained spanning trees are subcases of degree-constrained Steiner trees, the general problem of finding a Steiner tree satisfying given degree constraints, is $\mathcal{NP}$-complete.
The approximation results of section 3 can be adapted for most degree constraints. If $b(v) \geq 3$ for all $v \in V$, then $D \leq 2$ still holds since the construction in the proof of Theorem 3.1 can be revised by simple modifications. The proof does not work if there exist nodes v_i with $b(v_i) \leq 2$. However, the simple short-cutting heuristic of Rosenkrantz et al. [1977] for the TSP proves this ratio for the case $T = V$ and $b(v) = 2$ for all $v \in V$. In contrast, Theorem 3.3 showing $R \leq 2$ obviously holds for all degrees if $|\{v \in V : \ b(v) = 1\}| \leq 2$.
The IP-models given in section 4 can be adapted easily to the general case by just modifying the degree inequalities. Observation 4.1 then still holds. Further, the undirected and directed degree-constrained Steiner tree polytope can be defined analogously to the binary Steiner tree polytope. However, the polyhedral study gets more complicated. If $b(v) \geq 2$ for all $v \in V$ it is easy to show that the polytope of complete graphs remains full-dimensional. If there exist nodes $v_j \in V$ with $b(v_j) = 1$ the dimension varies depending on each value $b(v)$ for $v \in V$. It is even possible that no degree-constrained Steiner tree exists. In the following we therefore restrict ourselves to the case $b(v) \geq 2$ for all $v \in V$. The bound constraints $0 \leq x \leq 1$ remain facets. If $b(v_j) \geq n - 1$ for some $v_j \in V$ the degree constraint $d(v_j) \leq b(v_j)$ is redundant and does not induce a facet. However, for $2 \leq b(v_j) \leq n - 2$ one can show that the degree constraint induces a facet using some simple modifications in the proof of Theorem 5.5. Further, Theorem 5.6 and Lemma 5.7 obviously hold if $b(v) \geq 3$ for all $v \in V$. If there exists nodes $v_j \in V$ with $b(v_j) = 2$ it gets more complicated. For example, if $b(v) = 2$ for all $v \in V$ then only the partition inequalities that satisfies $(V_i \cap S) \geq 1$ or $(|V_i \cap T)| = 1$ for all $i = 1, \dots, k$ induce facets. Further, the lifted odd-hole inequalities are not facet-inducing.
The general blossom inequalities of arbitrary degree now read

$$x(E(H)) + x(J) \leq \left\lfloor \frac{1}{2} \left(\left(\sum_{v \in H} b(v) \right) + |J| \right) \right\rfloor, \quad \forall \ H \subseteq V, \ J \subseteq \delta(H).$$

The blossom inequalities are valid. However, general statements about facet-properties are difficult and depend on the number of nodes and on the value of $\sum_{v \in H} b(v)$. The blossom described in Theorem 5.8a) for example induces facets if $b(v) = b \leq n - 2$ for all $v \in V$ with $b \in \mathbb{N}$ odd and $m = |\bar{V}|$ odd. A similar result holds for Theorem 5.8b). The combined inequalities and facets can be generalized in the following way: For simplicity assume that $b(v) = b \geq 3$ for all $v \in V$. Then the combined inequalities of Theorem 5.10 turn into

$$\sum_{e \in E_1} x_e \ - \ \sum_{e \in E_2} x_e \ \leq \ \left\lfloor \frac{1}{2}(bN - K) \right\rfloor - \left\lfloor \frac{1}{2}(bL - D) \right\rfloor.$$

and

$$\sum_{e \in E_1} x_e \ - \ \sum_{e \in E_2} x_e \ \leq \ \left\lfloor \frac{1}{2}(bN - K) \right\rfloor - \left\lceil \frac{1}{2}(bL - D) \right\rceil.$$

respectively. If we set $K = (b-2)N - (b-2)L - D + 3$ and allow each node in $\{v_1, \ldots, v_L\}$ to have a degree in $\{1, \ldots, b-1\}$ (the rest remains the same as in Theorem 5.11), the inequality

$$\sum_{e \in E_1} x_e \; - \sum_{e \in E_2} x_e \; \leq \; N + D - L - 2$$

induces a facet for $P(G)$.

The primal heuristic described in section 6 is generalized by setting $B_i = B \setminus \{v \in B : \deg(v) = b(v)\}$ in step (2).
Since all described inequalities remain valid for the general case or can be modified into a valid inequality, the following branch-&-cut algorithm can be adapted easily for general instances as well.

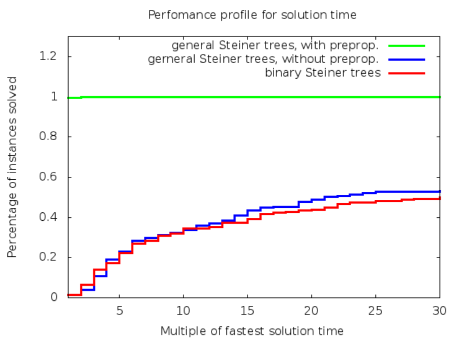
8 Computational Results

The branch-&-cut approach for solving the degree-constrained Steiner tree problem is based on IP-model 2. For a general outline of branch-&-cut algorithms see for example Caprara and Fischetti [1997] or Lucena and Beasley [1996].
In the initialization phase we set up the first LP consisting only of the trivial inequalities and the degree constraints of the directed graph. For solving the LP-relaxations we use the solver SCIP [2009], Version 3.0.1.2. See Achterberg [2009] for more information.
At the separation step we separate the cut inequalities exactly and the combined inequalities heuristically as explained in subsection 5.4. We use the algorithm of Hao and Orlin [1992] to determine minimal (s,t)-cuts between the root node s and all terminal nodes $t \in T$. In addition, flow-balance inequalities are added to strengthen the LP formulation, see Duin [1993] for more details.

We used the implementation of Koch and Martin [1998] for Steiner trees without degree constraints and adapted the algorithm to the binary case. Much more implementation details can be found in that paper.

We now present computational results for our branch-&-cut algorithm. Our code is implemented in C and all computations are performed on a compute server with 12-core Opteron processors and 64GB RAM. We have chosen a limit of one hour. We focus on binary Steiner trees. However, the branch-&-cut algorithm also works for Steiner tree problems with arbitrary degree. The instances include some realistic problems arising in the construction of tree alignments and phylogenetic trees as well as complete graphs taken from the SteinLib benchmark set [SteinLib [2008]]. The biological instances are taken from [Balibase [2005]] and from [Treefam [2008]]. For more information about Treefam see Li et al. [2006] and Ruan et al. [2008]. For more information about Balibase see Thompson et al. [2005] and Thompson et al. [1999]. Each file from Treefam and Balibase consists of a set of protein sequences. We generate instances for the binary Steiner tree problem in the following way:
For each sequence in the file we introduce one terminal node. Between each pair of nodes we add one edge with cost equal to the pairwise alignment cost of the corresponding sequences, see Needleman and Wunsch [1970]. We introduce Steiner nodes with the coalescement method similar to Foulds [1992]. We connect the Steiner nodes to all other nodes through edges with edge weight equal to the alignment cost.

The Balibase set consists of small instances with 10-70 nodes and 5-30 terminal nodes. The instances of Treefam are divided into three parts of small ($\leq$ 100 nodes), medium (100-200 nodes) and large ($\geq$ 200 nodes) size. The number of terminal nodes is approximately one half of the



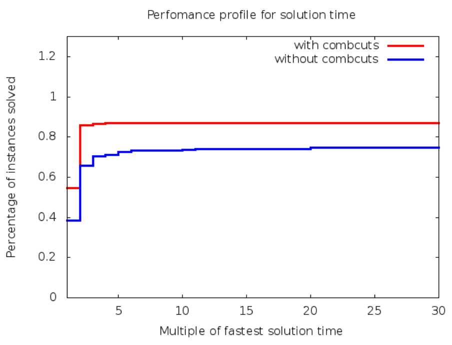


Figure 13: **Performance profiles: Left:** Performance plot comparing the solution times of instances of the binary Steiner tree problem and of the general Steiner tree problem (with and without preprocessing). **Right:** Performance plot comparing the solution times using the additional separation of the combined inequalities.

number of the entire nodes. The complete instances of SteinLib include three instances of the random MC-set, the sets P4E ands P4Z with random euclidean graphs and the set X with adapted TSPLIB problems.
The SteinLib benchmark set as well as some instances taken from Treefam can be found on on the webpage of the 11th DIMACS Implementation Challenge on Steiner trees [DIMACS [2014]].

It is interesting to compare the practical difficulty of the general Steiner tree problem with that of the binary Steiner tree problem. In fact, it turns out that the problem with degree constraints is considerably more difficult to solve in practice. The performance profile in Figure 13 (left) compares the solution times of these problems. The plot shows the percentage of solved instances as a function of the solution time that is given in multiples of the fastest solution of all compared approaches. The information of such a performance profile is twofold. First, the intercept of each curve with the vertical axis shows the percentage of instances for which the corresponding approach achieves the fastest solution time. Second, for each value k on the horizontal axis, the profile shows the percentage of instances that an approach can solve within a factor k of the best solution time achieved by the compared approaches. More information about performance profiles can be found in Dolan and More [2002]. We observe that we solve all instance much faster with the original implementation of Koch and Martin [1998] if degree constraints are not present. We can solve only half of the binary Steiner tree instances within a factor of 30 of the solution time for computing Steiner trees without degree constraints. Preprocessing routines have a major influence in general Steiner tree computation. Unfortunately, most preprocessing routines for Steiner trees are not valid for binary trees. Further, up to our knowledge there do not exists successful preprocessing routines for degree-constrained trees.

The solutions of the computational results for the binary Steiner tree instances are shown in Table 14 to 16. The format of our tables is as follows: The first column *instance* shows the problem name. In column two, three and four the size of the instance, i.e. the number of nodes $|V|$, the number of terminal nodes $|T|$ and the number of edges $|E|$ of the problem is given. Column five and six show the value of the upper and lower bound UB_{root} and LB_{root} before branching, i.e. the solution found by the first call to the primal heuristic and the LP value after separating all violated inequalities found in the root node. Comparing these values to the final lower and upper bounds UB and LB in column seven and eight provides information about the quality of the heuristic and the cutting plane procedure. If there is still a *gap* between the final lower and upper bound as depicted in column nine, we have not found the optimal solution within one hour. Column ten gives the number of branch-&-bound *nodes*. Here, 1 means that no branching was necessary. *Cuts* shows the number of violated cuts added to the LP. The last column gives the overall running *time* of the algorithm in seconds.

instance	$\|V\|$	$\|T\|$	$\|E\|$	UB$_{root}$	LB$_{root}$	UB	LB	gap	nodes	cuts	time
ref1_1/1aboA	34	5	561	3020	1665	2953	2953	0.00	194	2181	9.86
ref1_1/1aho	35	5	595	2344	1513	2117	2117	0.00	23	505	2.10
ref1_1/1csp	35	5	595	1672	1499	1499	1499	0.00	1	306	0.96
ref1_1/1csy	35	5	595	3724	3399	3399	3399	0.00	1	315	1.19
ref1_1/1fjlA	65	6	2080	2860	1370	2660	2660	0.00	1285	26945	369.61
ref1_1/1fkj	35	5	595	2823	2116	2622	2622	0.00	35	554	3.77
ref1_1/1hfh	35	5	595	4705	4343	4343	4343	0.00	1	263	1.12
ref1_1/1idy	34	5	561	2364	1191	2270	2270	0.00	142	1778	8.63
ref1_1/1krn	35	5	595	2288	2138	2138	2138	0.00	1	390	1.58
ref1_1/1pfc	35	5	595	3628	2015	3405	3405	0.00	113	1433	6.67
ref1_1/1plc	35	5	595	2318	2128	2136	2136	0.00	3	427	2.57
ref1_1/1wit	35	5	595	4092	3917	3941	3941	0.00	6	356	1.76
ref1_1/2fxb	34	5	561	1361	1248	1248	1248	0.00	1	352	1.59
ref1_1/2mhr	35	5	595	2953	2652	2652	2652	0.00	1	309	1.15
ref1_1/451c	35	5	595	3114	1624	2945	2945	0.00	78	1130	4.86
ref1_1/9rnt	35	5	595	2267	2062	2062	2062	0.00	1	298	1.12
ref1_2/1amk	35	5	595	5124	4689	4689	4689	0.00	1	292	0.95
ref1_2/1bbt3	35	5	595	8170	4045	7995	7995	0.00	313	3232	18.19
ref1_2/1ezm	35	5	595	5614	5112	5112	5112	0.00	1	309	1.30
ref1_2/1havA	35	5	595	8971	4076	8669	8669	0.00	203	2284	11.31
ref1_2/1ppn	35	5	595	5003	4538	4538	4538	0.00	1	351	1.11
ref1_2/1sbp	35	5	595	10926	5763	10534	10534	0.00	166	1932	9.32
ref1_2/1tis	35	5	595	7172	6734	6734	6734	0.00	1	334	1.08
ref1_2/1ton	35	5	595	8839	8458	8458	8458	0.00	1	375	1.28
ref1_2/2cba	35	5	595	9376	8818	8818	8818	0.00	1	316	1.01
ref1_2/5ptp	35	5	595	5851	5560	5560	5560	0.00	1	423	2.16
ref1_2/kinase	35	5	595	11112	5482	10797	10797	0.00	103	997	5.72
ref1_3/1gpb	35	5	595	18183	16615	16615	16615	0.00	1	399	1.52
ref1_3/1gtr	35	5	595	11857	6888	10656	10656	0.00	135	1827	8.06
ref1_3/1lcf	66	6	2145	21864	20326	20326	20326	0.00	1	809	11.82
ref1_3/1pamA	35	5	595	23995	19749	23320	23320	0.00	31	467	2.49
ref1_3/1rthA	35	5	595	13447	12464	12464	12464	0.00	1	374	1.67
ref1_3/1sesA	35	5	595	13459	12604	12604	12604	0.00	1	321	0.94
ref1_3/1taq	35	5	595	25044	23445	23445	23445	0.00	1	344	1.20
ref1_3/2ack	35	5	595	18373	17300	17300	17300	0.00	1	303	0.95
ref1_3/actin	35	5	595	8952	8240	8240	8240	0.00	1	317	1.02
ref1_3/arp	35	5	595	13754	12879	12879	12879	0.00	1	390	1.29
ref1_3/gal4	35	5	595	16293	13604	15774	15774	0.00	80	901	5.69
ref1_3/glg	35	5	595	15495	14550	14646	14646	0.00	7	340	1.33
ref1_3/pol	35	5	595	24973	23445	23445	23445	0.00	1	320	0.99
ref2/1aboA	35	16	595	7562	7148	7148	7148	0.00	1	183	0.82
ref2/1ajsA	50	20	1225	37089	34455	34455	34455	0.00	1	442	4.58
ref2/1cpt	41	15	820	41888	39013	39194	39194	0.00	12	477	4.99
ref2/1csy	57	19	1596	12232	11238	11244	11244	0.00	2	253	3.06
ref2/1havA	42	19	861	27868	26115	26115	26115	0.00	1	274	1.87
ref2/1idy	58	22	1653	7889	7228	7239	7239	0.00	3	512	7.82
ref2/1lvl	68	24	2278	72103	65747	66164	66164	0.00	158	3427	198.78
ref2/1pamA	49	19	1176	48836	45489	45539	45539	0.00	3	590	7.50
ref2/1ped	58	19	1653	32307	29470	29500	29500	0.00	3	702	13.50
ref2/1r69	59	23	1711	11288	10566	10601	10601	0.00	42	541	15.63
ref2/1sbp	41	19	820	31207	29084	29084	29084	0.00	1	238	1.94
ref2/1tgxA	47	20	1081	8411	7770	7770	7770	0.00	1	375	2.92
ref2/1tvxA	38	19	703	6562	6102	6102	6102	0.00	1	158	0.74
ref2/1ubi	41	17	820	9644	8906	8929	8929	0.00	3	284	2.54
ref2/1uky	58	24	1653	31464	28878	28878	28878	0.00	1	418	5.12
ref2/1wit	51	22	1275	16917	15997	16037	16037	0.00	28	607	13.62
ref2/2hsdA	59	22	1711	38043	34885	34908	34908	0.00	3	680	17.84
ref2/2myr	52	19	1326	57857	54076	54151	54151	0.00	9	742	21.11
ref2/2pia	43	18	903	31161	29280	29280	29280	0.00	1	335	2.37
ref2/2trx	55	21	1485	10406	9439	9446	9446	0.00	8	541	15.33
ref2/3grs	42	16	861	24247	22464	22464	22464	0.00	1	323	2.62
ref2/3lad	47	16	1081	47216	42789	42933	42933	0.00	91	758	10.08
ref2/4enl	52	18	1326	30637	28011	28173	28173	0.00	21	657	16.35
ref2/kinase	55	20	1485	36632	34190	34307	34307	0.00	15	611	14.98
ref2/subtilase	28	13	378	30495	29038	29038	29038	0.00	1	144	0.65
ref2/test	38	19	703	6562	6102	6102	6102	0.00	1	158	0.76
ref3/1ajsA	68	28	2278	62191	57154	57244	57244	0.00	11	684	26.68
ref3/1idy	62	27	1891	8480	7918	7930	7930	0.00	14	560	10.96
ref3/1pamA	46	19	1035	49078	45580	45665	45665	0.00	4	479	4.34
ref3/1ped	52	21	1326	39920	36044	36080	36080	0.00	4	490	6.30
ref3/1r69	53	23	1378	11254	10406	10429	10429	0.00	2	337	3.56
ref3/1ubi	51	22	1275	13057	12213	12226	12226	0.00	4	419	4.83
ref3/1uky	58	25	1653	31999	29717	29782	29782	0.00	7	548	8.85
ref3/1wit	43	19	903	13408	12518	12518	12518	0.00	1	219	1.28
ref3/2myr	54	21	1431	61238	56820	56820	56820	0.00	1	590	6.85
ref3/2pia	46	20	1035	32616	30468	30590	30590	0.00	7	403	3.60
ref3/2trx	52	22	1326	12007	10813	10836	10836	0.00	6	464	7.66
ref3/4enl	50	19	1225	29797	27353	27418	27418	0.00	6	547	8.57
ref3/clustalx	67	28	2211	42329	38962	38962	38962	0.00	1	506	7.72
ref3/kinase	56	25	1540	32663	30350	30403	30403	0.00	8	415	5.23
ref3/test	62	27	1891	8480	7918	7930	7930	0.00	14	560	10.78

Figure 14: Solution of alignment instances taken from Balibase.

instance	$\|V\|$	$\|T\|$	$\|E\|$	UB$_{root}$	LB$_{root}$	UB	LB	gap	nodes	cuts	time
TF101057	52	35	1326	3046	2731	2756	2756	0.00	547	1791	69.77
TF101166	23	12	253	97699	95045	95045	95045	0.00	1	82	0.14
TF103004	74	34	2701	15141	14090	14121	14121	0.00	62	850	49.90
TF105297	98	34	4753	36303	33099	33302	33302	0.00	226	5466	496.93
TF105419	55	24	1485	20908	18138	18223	18223	0.00	13	534	10.80
TF105502	10	6	45	82855	81339	81339	81339	0.00	1	30	0.02
TF105551	15	8	105	11316	11133	11133	11133	0.00	1	72	0.07
TF105034	78	45	3003	6477	6116	6124	6124	0.00	11	575	24.11
TF102047	110	41	5995	38698	35030	35127	35127	0.00	85	2405	333.62
TF106403	119	46	7021	57824	53690	53760	53760	0.00	9	1311	155.14
TF106470	102	43	5151	17294	15450	15469	15469	0.00	437	3836	504.33
TF101136	183	62	16653	39369	34776	39369	34915	12.76	96	5424	3600.80
TF101202	188	72	17578	82927	76571	82927	76691	8.13	136	5813	3600.16
TF102011	178	69	15753	72998	66800	67000	67000	0.00	292	4908	2841.50
TF102048	164	63	13366	41953	37824	37914	37914	0.00	409	5279	2604.39
TF106357	154	52	11781	55426	50185	50367	50367	0.00	170	6636	3546.03
TF106478	130	54	8385	61778	54583	54839	54839	0.00	302	2854	724.86
TF101125	304	155	46056	59564	54059	59564	54078	10.15	7	2565	3623.36
TF101141	586	274	171405	105875	93455	105875	93455	13.29	1	2529	3629.54
TF101142	251	143	31375	35050	33040	33165	33052	0.34	37	1996	3603.01
TF102003	832	407	345696	196030	162288	196030	162288	20.79	1	1949	3650.80
TF105035	237	104	27966	35522	32720	35522	32737	8.51	9	2796	3600.43
TF105062	254	112	32131	42268	38164	42268	38225	10.58	19	3334	3600.25
TF105067	341	148	57970	35620	31008	35620	31016	14.85	2	2944	3628.18
TF105272	476	223	113050	138233	122002	138233	122002	13.30	1	2886	3644.92
TF105897	314	133	49141	105399	95145	105399	95150	10.77	7	2183	3638.16
TF106219	302	128	45451	82838	75193	82838	75243	10.09	8	3334	3620.58
TF105309	225	81	25200	81599	74227	81599	74319	9.80	36	3068	3606.95
TF105836	228	100	25878	55425	50343	55425	50419	9.93	67	3673	3600.19
TF105836	227	99	25651	55425	50480	55425	50525	9.70	61	3378	3603.74
TF106190	226	93	25425	93881	86051	93881	86141	8.99	61	3892	3604.12
TF106190	225	92	25200	93881	86169	93881	86251	8.85	69	3812	3600.16

Figure 15: Solution of phylogenetic instances taken from Treefam of small ($\leq$ 100 nodes), medium (100-200 nodes) and large size ($\geq$ 200 nodes).

instance	$\|V\|$	$\|T\|$	$\|E\|$	UB$_{root}$	LB$_{root}$	UB	LB	gap	nodes	cuts	time
P4E/p455	100	5	4950	1166	1138	1138	1138	0.00	1	468	21.87
P4E/p456	100	5	4950	1239	1228	1228	1228	0.00	1	609	23.35
P4E/p457	100	10	4950	1642	1609	1609	1609	0.00	1	588	22.15
P4E/p458	100	10	4950	1868	1868	1868	1868	0.00	1	614	18.35
P4E/p459	100	20	4950	2386	2347	2347	2347	0.00	1	508	17.02
P4E/p460	100	20	4950	3010	2959	2959	2959	0.00	1	691	22.61
P4E/p461	100	50	4950	4491	4474	4474	4474	0.00	1	504	23.40
P4E/p463	200	10	19900	1544	1510	1510	1510	0.00	1	1241	306.57
P4E/p464	200	20	19900	2574	2545	2545	2545	0.00	1	1435	289.29
P4E/p465	200	40	19900	3896	3853	3853	3853	0.00	1	1170	171.90
P4E/p466	200	100	19900	6253	6234	6234	6234	0.00	1	845	163.22
P4Z/p401	100	5	4950	170	155	155	155	0.00	1	109	5.47
P4Z/p402	100	5	4950	116	116	116	116	0.00	1	34	1.89
P4Z/p403	100	5	4950	184	179	179	179	0.00	1	128	5.93
P4Z/p404	100	10	4950	289	289	289	289	0.00	1	125	5.51
P4Z/p405	100	10	4950	298	298	298	298	0.00	1	133	5.74
P4Z/p406	100	10	4950	293	290	290	290	0.00	1	105	3.92
P4Z/p407	100	20	4950	602	602	602	602	0.00	1	275	9.06
P4Z/p408	100	20	4950	560	556	560	560	0.00	3	274	12.89
P4Z/p409	100	50	4950	966	966	966	966	0.00	1	177	4.22
P4Z/p410	100	50	4950	1016	1016	1016	1016	0.00	1	173	3.43
X/berlin52	52	16	1326	1069	1044	1044	1044	0.00	1	227	1.70
X/brasil58	58	25	1653	13682	13655	13655	13655	0.00	1	262	2.48
X/world666	666	174	221445	123167	116601	123167	116601	5.63	1	2550	3615.46
MC/mc13	150	80	11175	106	93	106	93	15.21	192	12272	3600.09
MC/mc2	120	60	7140	83	72	83	73	13.70	629	10676	3600.85
MC/mc3	97	45	4656	55	45	55	47	19.56	857	62141	3600.03

Figure 16: Solution of complete instances taken from SteinLib.

Table 14 shows the result for the Balibase test series. This test set is easy. We can solve all instances to optimality within some minutes, most of them we can solve within some seconds. For about half of the instances no branching is necessary. The first part of the table (ref1_1, ref1_2, ref1_3) contains instances with only five or six terminal nodes. Hence, the optimal trees for the Steiner tree problem with no degree constraints typically already fulfill the degree constraints in all or almost all nodes. This is not the case for the second part of the table (ref2 and ref3). These instances have more terminal nodes and optimal Steiner trees typically violate some degree constraints that would be given in the binary scenario. However, the solution times are still very fast.

The Treefam test set is more difficult as depicted in Table 15 The small instances with less than 100 nodes can again be solved to optimality within some seconds or minutes. Of the medium

sized instances with 100-200 nodes we can solve all instances with less than 180 nodes. However, two instances with about 180 nodes and all large instance with more than 200 nodes can not be solved to optimality within one hour.
Table 16 shows the result for the SteinLib set. The sets P4E and P4Z are easy and can all be solved to optimality quite fast. If we consider these instances as general Steiner tree problems without degree constraints, the optimal trees already have only a few nodes with degree more than three. The instance *world666* and all instances of MC are more difficult. We could not find optimal solutions (or proof optimality) within one hour.

For most instances in the Balibase, Treefam and SteinLib set the heuristic of Takahashi and Matsuyama computes a good bound which is not very far from the computed lower bound. However, there exist only few cases (compare P4E and P4Z) in which the heuristic already finds the optimal solution. Hence, there is still room for improvements with respect to heuristic algorithms. The results of the lower bound value in the root are much better. For about half of the instances the first LP solution after separating all violated inequalities already gives the optimal solution, i.e. no branching is necessary. Thereby, the separation of the combined inequalities has a big influence. In about 30% of all instances, the separation of these cuts improves the lower bound in the root. This improvement results typically in a reduction of branching nodes and solution time. The performance profile in Figure 13 (right) compares the overall solution times of the implementation without separating the combined inequalities to the solution times that arise if we apply the additional heuristic separation. The plot shows the percentage of solved instances as a function of the solution time that is given in multiples of the fastest solution. Although additional cuts increase the running time for the whole separation step the reduction of branching nodes results often in a decrease of the total running time. A few instances ca be solved just because of the additional separation. Thus, our new cutting planes are indeed advantageous for computing binary Steiner trees. Improvements in the separation heuristic of these inequalities would probably achieve a further benefit in performance.
The number of cuts added to the LP varies between some hundreds and more than 60.000. We could not detect a correlation between the number of violated inequalities added to the LP and the number of nodes or terminal nodes. During the solution process of instance *ref1_1/1fjlA* of the Balibase set for example more than 25.000 cuts are added although the number of nodes and terminal nodes is small.
In conclusion, we can solve almost all small and medium instances up to 200 nodes but unfortunately none of the larger instances with more than 200 nodes. If we can solve an instance, often we can solve it within some seconds or minutes. Allowing more time typically does not help much. Within ten hours we could not solve any of those instances which we have not already solved in one hour.

Further computational results including binary spanning tree instances can be found in the forthcoming PhD-Thesis of Pape [2015].

What one would like to have is of course a comparison with other codes. However, up to our knowledge there do not exist any further implementations for degree-constrained Steiner trees so far.

9 Conclusions

In this paper, hardness and polyhedral results for the binary Steiner tree problem and extensions to the more general case of degree-constrained Steiner trees are presented. A new class of valid inequalities that define facets for the binary Steiner tree polytope is introduced. The results are then used in a branch-&-cut algorithm. Although our findings could be used for the general degree-constrained Steiner tree problem, we focus on the binary case here. The computational experiments are carried out on binary biological instances and binary instances from classical Steiner tree benchmark sets. We are able to solve almost all instances up to 200 nodes to optimality. Thereby, the separation of the new class of facets achieves a significant improvement in running time. The results can be used to construct phylogenetic trees for sets of biological sequences.

10 Acknowledgement

We thank Thorsten Koch for providing his branch-&-cut implementation for Steiner trees.

References

T. Achterberg. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

D.L Applegate, R.E. Bixby, V. Chvatal, and W.J. Cook. *The Traveling Salesman Problem: A Computational Study.* Princeton University Press, 2006.

Balibase. Multiple Sequence Alignemnt Benchmark, 2005. URL `http://lbgi.fr/balibase/`.

N. Bansal, R. Khandekar, and V. Nagarajan. Additive Guarantees for Degree Bounded Directed Network Design. *Siam J. Comput.*, 39(4):1413–1431, 2009.

L. Caccetta and S.P. Hill. A Branch and Cut Method for the Degree-Constrained Minimum Spanning Tree Problem. *Networks*, 37(2):74–83, 2001.

A. Caprara and M. Fischetti. Branch-and-Cut Algorithms. *Annotated Bibliographies in Combinatorial Optimization*, pages 45–64, 1997.

S. Chopra and M.R. Rao. The Steiner Tree Problem I: Formulations, Compositions and Extensions of Facets. *Math. Programming*, 64:209–229, 1994a.

S. Chopra and M.R. Rao. The Steiner Tree Problem II: Properties and Classes of Facets. *Math. Programming*, 64:231–246, 1994b.

S. Chopra, E.R. Gorres, and M.R. Rao. Solving the Steiner Tree Problem on a Graph Using Branch and Cut. *ORSA Journal on Computing*, 4(3):320–335, 1992.

N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, CMU, 1976.

A.S. Cunha. *Arvores otimas em grafos: modelos, algorithmos e applicacoes.* PhD thesis, Programma de Engenharia de Sistemas e Computacao, Universidade Federal do Rio de Janeiro, 2006.

A.S. Cunha and A. Lucena. Lower and Upper Bounds for the Degree-Constrained Minimum Spanning Tree Problem. *Networks*, 50(1):55–66, 2007.

DIMACS. 11th DIMACS Implementation Challange on Steiner Tree Problems, 2014. URL `http://dimacs11.cs.princeton.edu/downloads.html`.

E.D. Dolan and J.J. More. Benchmarking Optimization Software with Performance Profiles. *Math. Programming*, 91:201–213, 2002.

C. Duin. *Steiner's Problem in Graphs.* PhD thesis, University of Amsterdam, 1993.

J. Felsenstein. *Inferring Phylogenies.* Sunderland, 2004.

L.R. Foulds. *Graph Theory Applications.* Springer, 1992.

M. Furer and B. Raghavachari. Approximating the Minimum-Degree Steiner Tree to within One of Optimal. *Journal of Algorithms*, 17(3):409–423, 1994.

E.N. Gilbert and H.O. Pollak. Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

M.X. Goemans and Y.S Myung. A Catalog of Steiner Tree Formulations. *Networks*, 23:19–28, 2006.

M. Grötschel, C.L. Monma, and M. Stoer. Computational Results With a Cutting Plane Algorithm for Designing Communication Networks With Low-Connectivity Constraints. *Operations Research*, 40:309–330, 1992.

J. Hao and J.B. Orlin. A Faster Algorithm for Finding the Minimum Cut in a Graph. *Proceedings of the Third Annual ACM-Siam Symposium on Discrete Algorithms*, pages 165–174, 1992.

F.K. Hwang and D.S. Richards. Steiner Tree Problems. *Networks*, 22:55–89, 1992.

F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem.* North-Holland, 1992.

R. Khandekar, G. Kortsarz, and Z. Nutov. On Some Network Design Problems with Degree-Constraints. *Journal of Computer and System Siences*, 79(5):725–736, 2013.

S. Khuller, B. Raghavachari, and N. Young. Low-Degree Spanning Trees of Small Weight. *Siam J. Comput.*, 25(2):355–368, 1996.

T. Koch and A. Martin. Solving Steiner Trees Problems in Graphs to Optimality. *Networks*, 32: 207–232, 1998.

L.C. Lau and M. Singh. Additive Approximation for Bounded Degree Survivable Network Design. *STOC*, pages 759–768, 2008.

L.C. Lau, J. Naor, M.R. Salavatipour, and M. Singh1062. Survivable Network Desing Problems with Degree or Order Constraints. *Siam J. Comput.*, 39(3):1062–1087, 2009.

A.N. Letchford, G. Reinelt, and O. Theis. Odd Minimum Cut Sets and b-Matching Revisited. *Siam J. Alg. Disc. Math.*, 22(4):1480–1487, 2008.

H. Li, A. Coghlan, J. Ruan, L.J. Coin, J.K. Hériché, L. Osmotherly, R. Li, T. Liu, Z. Zhang, L. Bolund, GK Wong, W. Zheng, P. Dehal, J. Wang, and R. Durbin. TreeFam: A Curated Database of Phylogenetic Trees of Animal Gene Families. *Nucleic Acids Research*, 34, 2006.

A. Louis and N. Vishnoi. Improved Algorithm for Degree Bounded Survivable Network Design Problem. *SWAT*, 6139:408–419, 2010.

A. Lucena. Non Delayed Relax-and-Cut Algorithms. *Annals of Operations Research*, 140:375–410, 2005.

A. Lucena and J.E. Beasley. Branch and Cut Algorithms. *Advances in linear and integer programming*, 4:187–221, 1996.

A. Lucena and J.E. Beasley. A Branch and Cut Algorithm for the Steiner Problem in Graphs. *Networks*, 31:39–59, 1998.

N. Maculan. The Steiner Problem in Graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.

S.B. Needleman and C.D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

S. Pape. Binary Steiner Trees. *forthcoming Phd Thesis*, FAU Erlangen-Nürnberg, Department of Mathematics, 2015.

T. Polzin and S.V. Daneshmand. A Comparison of Steiner Tree Relaxations. *Discrete Applied Mathematics*, 112:241–261, 2001.

H.J. Proemel and A. Steger. *The Steiner Tree Problem.* Amer Mathematical Society, 2002.

R. Ravi, J.S. Marathe, S.S. Ravi, D.J. Rosenkrantz, and H.B. Hunt. Many Birds With One Stone: Multi-Objective Approximation Algorithms. In *Proc. 10th Annual ACM Symposium on Computational Geometry*, 1993.

R. Ravi, J.S. Marathe, S.S. Ravi, D.J. Rosenkrantz, and H.B. Hunt. Approximation Algorithms for Degree-Constrained Minimum-Cost Network-Design Problems. *Algorithmica*, 31: 58–78, 2001.

V.J. Rayward-Smith. The Computation of Nearly Minimal Steiner Trees in Graphs. *Int. J. Math. Educ. Sci. Technol.*, 14:283–294, 1983.

D.J. Rosenkrantz, R.E. Stearns, and P.M. Lewis. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

J. Ruan, H. Li, Z. Chen, A. Coghlan, L.J. Coin, Y. Guo, J.K. Hériché, Y. Hu, K. Kristiansen, R. Li, T. Liu, A. Moses, J. Qin, S. Vang, A.J. Vilella, A. Ureta-Vidal, L. Bolund, J. Wang, and R. Durbin. TreeFam: 2008 Update. *Nu*, 36, 2008.

SCIP. Solving Constraint Integer Programs, 2009. URL `http://scip.zib.de/`.

SteinLib. Collection of Steiner Tree Problems, 2008. URL `http://steinlib.zib.de`.

H. Takahashi and A. Matsuyama. An Approximate Solution for the Steiner Problem in Graphs. *Math. Japonica*, 24(6):573–577, 1979.

J.D. Thompson, F. Plewniak, and O. Poch. Balibase: A Benchmark Alignment Database for the Evolution of Multiple Alignment Programs. *Bioinformatics*, 15(1):87–88, 1999.

J.D. Thompson, P. Koehl, and O. Poch. Balibase 3.0: Latest Developments of the Multiple Sequence Alignment Benchmark. *Proteins*, 61(1):127–36, 2005.

Treefam. Database of Animal Gene Trees, 2008. URL `http://www.treefam.org/`.

P. Winter. Steiner Problem in Networks: A Survey. *Networks*, 17:129–167, 1987.