Vertex-Disjoint Packing of Two Steiner Trees: Polyhedra and Branch-and-Cut

Eduardo Uchoa
e-mail: uchoa@inf.puc-rio.br
Marcus Poggi de Aragão
e-mail: poggi@inf.puc-rio.br

PUC-Rio.Inf.MCC 53/98 December, 1998

Abstract: *Consider the problem of routing the electrical connections among two large terminal sets in circuit layout. A realistic model for this problem is given by the vertex-disjoint packing of two Steiner trees (2VPST), which is known to be NP-complete. This work presents an investigation on the 2VPST polyhedra. The main idea is to depart from facet-defining inequalities for a vertex-weighted Steiner tree polyhedra. Some of these inequalities are proven to also define facets for the packing polyhedra, while others can be lifted to derive new important families of inequalities, including proven facets. Separation algorithms are also presented. The resulting branch-and-cut code has an excellent performance and is capable of solving problems on grid graphs with up to 10000 vertices and 5000 terminals in a few minutes.*

Keywords: Steiner trees, VLSI circuit layout, cutting, Integer Programming

Resumo: *Considere o problema de rotear as conexões elétricas entre dois grandes conjuntos de terminais, seja no desenho de uma placa de circuito impresso ou num circuito integrado VLSI. Um modelo realista para este problema é dado pelo empacotamento de duas árvores de Steiner disjuntas nos vértices (2VPST), um problema NP-completo. Esse trabalho apresenta uma investigação sobre o poliedro do 2VPST. A principal idéia é partir de desigualdades que definem facetas para o poliedro do problema da árvore de Steiner com peso nos vértices. Provamos que algumas dessas desigualdades também definem facetas para o poliedro do empacotamento, enquanto outras podem ser submitidas a um procedimento de* lifting *para gerar novas famílias de desigualdades, incluindo facetas. Procedimentos de separação são também estudados. O algoritmo de* branch-and-cut *resultante é capaz de resolver problemas em grafos de grade com até 10000 vértices e 5000 terminais em alguns minutos.*

Palavras-chave: Árvores de Steiner, Desenho de circuitos, Cortes poliédricos, Programação Inteira

1 Introduction

The Vertex-disjoint Packing of Steiner Trees problem (VPST) is the following: given a connected graph $G = (V, E)$ and K disjoint vertex-sets $N_1, \ldots, N_K$, called *terminal sets*, find disjoint vertex-sets $T_1, \ldots, T_K$ such that, for each $k \in \{1, \ldots, K\}$, $N_k \subseteq T_k$ and $G[T_k]$ (the subgraph induced by T_k) is connected. Deciding whether a VPST instance is feasible or not is NP-complete, and many times a very challenging task. VPST has a wide application in layout of printed boards or chips, as it can model in an accurate way the routing of electrical connections among the circuit components (see Korte et al. [3] or Lengauer [4]).

This work studies the case where $K = 2$, the Vertex-Disjoint Packing of Two Steiner Trees (2VPST). This restricted problem is still NP-complete [3]. Although most VPST applications involves more than two terminal sets, many problems in circuit layout have only two large terminal sets, for instance the ground and the V_{cc} terminals. Finding a solution for these two large sets may represent a major step in solving the whole routing problem. In fact, it is usual in chip layout to route the ground and V_{cc} connections over special layers, separated from the remaining connections. This particular routing problem leads to the 2VPST.

We address an optimization version of the 2VPST: find two disjoint vertex-sets T_1 and T_2, such that $G[T_1]$ and $G[T_2]$ are connected, minimizing $|N_1 \setminus T_1| + |N_2 \setminus T_2|$. In other words, we look for a solution leaving the minimum possible number of unconnected terminals. It is also allowed to set weights in order to specify which terminals have priority in being connected. These features may be very useful for a practitioner and suggests an iterative approach for layout: if the 2VPST instance is not feasible, a best possible solution gives valuable information on how to make small changes in terminal placement to achieve feasibility.

A previous work that is somehow related to ours is Grötschel et al. [1, 2], where Edge-disjoint Packing of Steiner Trees (EPST) is studied. EPST gives an alternative model for the routing problem in circuit layout. Their branch-and-cut code could solve switchbox instances (G is a grid graph and all terminals are placed in the outer face) with up to 345 vertices, 24 terminal sets, and 68 terminals.

2 Formulation

For each vertex $i \in V$ define a binary variable x_i, where $x_i = 1$ if i belongs to T_1 and $x_i = 0$ if i belongs to T_2. This choice of variables obliges every vertex to belong either to T_1 or T_2. Of course, terminals of N_1 in T_2 and terminals of N_2 in

T_1 are considered unconnected. A formulation for 2VPST follows:

$$
\text{(F)} \begin{cases}
\text{Min } z = & -\sum_{i \in N_1} w_i + \sum_{i \in V} w_i x_i & & (1) \\
\text{s.t.} & \sum_{i \in S} x_i - \sum_{i \in R} x_i \leq 1 & \forall S \subseteq V,\ |S| = 2;\ \forall R \in \Delta(S) & (2) \\
& -\sum_{i \in S} x_i + \sum_{i \in R} x_i \leq |R| - 1 & \forall S \subseteq V,\ |S| = 2;\ \forall R \in \Delta(S) & (3) \\
& x_i \in \{0, 1\} & \forall i \in V & (4)
\end{cases}
$$

where w_i must be negative if $i \in N_1$, positive if $i \in N_2$ and 0 otherwise. The number of unconnected terminals matches z when $w_i = -1$ for all $i \in N_1$ and $w_i = 1$ for all $i \in N_2$. In any case the 2VPST instance is feasible iff $z^* = 0$. The symbol $\Delta(S)$ denotes the set of all minimal separators of S. A vertex-set R is a *separator of* S if any path joining the pair of vertices in S has at least one vertex in R. Constraints F.2 says: if both vertices in S are in T_1, then at least one vertex in any separator of S must also be in T_1. Constraints F.3 are complementary to F.2: if both vertices in S are in T_2, then at least one vertex in any separator of S must be in T_2.

3 Polyhedral Investigations

Definition 1 *Given a graph $G = (V, E)$, a* v-tree of G *is a vertex-set $T \subseteq V$ such that $G[T]$ is connected. A* double v-tree of G *is a v-tree T of G such that $V \setminus T$ is also a v-tree of G. (G being clear, we say v-tree or double v-tree for short).*

Definition 2 *Let χ^U be the incidence vector of a vertex-set U. Define $P(G)$ as the convex hull of vectors χ^T such that T is a double v-tree of G. Define $P_1(G)$ as the convex hull of vectors χ^T such that T is a v-tree of G and $P_2(G)$ as the convex hull of vectors χ^T such that $V \setminus T$ is a v-tree of $G = (V, E)$.*

The vectors defining polyhedra P correspond to the solutions of formulation F, so P may be associated to 2VPST. Polyhedra P_1 and P_2 may be associated to a Vertex Weighted Steiner Tree problem: given a graph with positive and negative vertex weights, find a tree with maximum/minimum total vertex weight. Note that P_1 and P_2 are complementary polyhedra and that every valid inequality for one can be converted into a valid inequality for the other by variable complementation, i.e. replacing x by $1 - x$. For instance, inequalities F.2 are valid for P_1 and the complementary inequalities F.3 are valid for P_2. Therefore when we talk about polyhedra P_1, it is implicit that similar results also hold for P_2.

Theorem 1 *$P(G)$, $P_1(G)$ and $P_2(G)$ are full-dimensional.*

Proof: Let A be a rooted spanning tree of G (remember G is supposed to be connected). For every vertex i in V, let U_i be the double v-tree containing i and his descendents in A, then $\chi^{U_i} \in P(G)$. As $\chi^{\emptyset} \in P(G)$, there are $|V|+1$ affine independent vectors in $P(G)$. As $P(G) \subseteq P_1(G) \cap P_2(G)$, the result follows. ■

Our idea is to use subproblem polyhedra P_1 and P_2 as a starting point for investigating main polyhedra P. We show that: (i) some inequalities that define facets of P_1 or P_2 also define facets of P and (ii) some of these inequalities which do not define facets of P can be lifted to derive other important valid inequalities for P.

3.1 The p-separator facets of P_1

Definition 3 *Let S be a vertex-set, where $|S| \geq 2$. A vertex-set R is a* separator *of S if any path joining a pair of vertices in S has at least one vertex in R. Given a separator R of S, we say that R is* proper *if $S \cap R = \emptyset$ and that R is* minimal *if no other separator of S is contained in R. The symbol $\Delta(S)$ indicates the set of all proper and minimal separators of S. For each vertex $i \in R \in \Delta(S)$, define S_i^R as the maximum subset of vertices in S that can belong to a v-tree T_i such that $T_i \cap R = \{i\}$. Note that $|S_i^R| \geq 2$ for each $i \in R$.*

Definition 4 *Let S be a vertex-set, where $|S| = p \geq 2$, and R be a separator in $\Delta(S)$. The following inequalities are called* p-separators, *where the coefficient α_i must be an integer greater than $|S_i^R| - 1$.*

$$\sum_{i \in S} x_i - \sum_{i \in R} \alpha_i x_i \leq 1$$

Constraints F.2 are 2-separators. In graph $G1$ (figure 1), $x_a + x_f - x_b - x_e - x_j \leq 1$ is an example of such inequality. In the same graph, $x_a + x_f + x_g - x_b - 2x_e - 2x_k \leq 1$ is a 3-separator valid for $P_1(G1)$. Sometimes it is not possible to take $\alpha_i = |S_i^R| - 1$ for all $i \in R$, as in the above examples. For instance, in graph $G2$ (figure 1), the 4-separator $x_a + x_b + x_c + x_d - x_e - x_f \leq 1$ is not valid for $P_1(G2)$. But 4-separators $x_a + x_b + x_c + x_d - 2x_e - x_f \leq 1$ and $x_a + x_b + x_c + x_d - x_e - 2x_f \leq 1$ are valid and facet-defining for $P_1(G2)$.

Theorem 2 *For each S and each $R \in \Delta(S)$, there is at least one p-separator inequality which is valid and facet-defining for $P_1(G)$.*

Corollary 1 *If the p-separator $\sum_{i \in S} x_i - \sum_{i \in R}(|S_i^R| - 1)x_i \leq 1$ is valid for $P_1(G)$, then it defines a facet of $P_1(G)$.*

The above theorem does not tell how to compute the α coefficients of a p-separator facet. But given sets S and $R \in \Delta(S)$, the coefficients of a valid p-separator for $P_1(G)$ can be obtained by the following procedure:

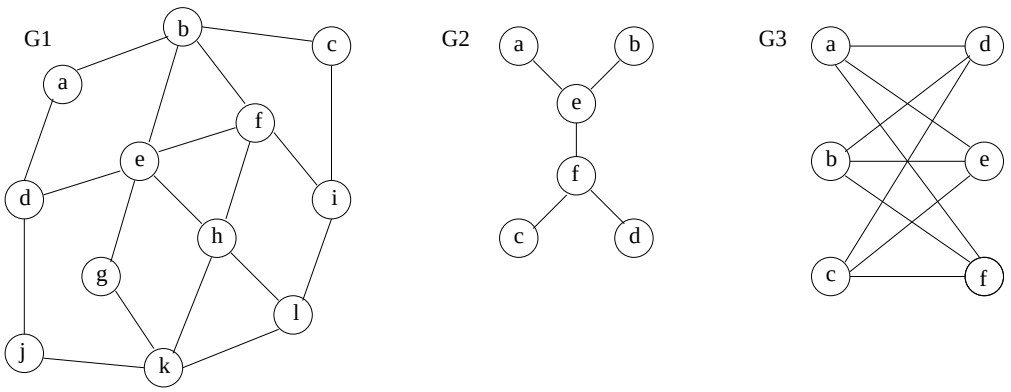

Figure 1: Graphs for examples

Procedure Compute α
$M \longleftarrow \emptyset$;
For each $i \in R$ {
If (exists $j \in M$ such that $S_i^R \cap S_j^R = \emptyset$)
Then { $\alpha_i \longleftarrow |S_i^R|$; }
Else { $\alpha_i \longleftarrow |S_i^R| - 1$; $M \longleftarrow M \cup \{i\}$; }
}

As example, take graph G2 with $S = \{a, b, c, d\}$ and $R = \{e, f\}$. If we first pick vertex e, the procedure yields $\alpha_e = 1$ and $\alpha_f = 2$. If vertex f is first picked, the coefficients are reversed. If $p = 2$ or $p = 3$, the procedure always yields $\alpha_i = |S_i^R| - 1$ for all $i \in R$. Then, by corollary 1, all such 2-separators and 3-separators define facets of $P_1(G)$. If $p \geq 4$, the above procedure is not optimal, therefore the resulting p-separator is not always facet-defining.

3.2 From P_1 to P

Definition 5 *Let T be a double v-tree and t a vertex in T. A* sequence of nested double v-trees from $\{t\}$ to T *is any sequence of double v-trees $T_1, \ldots, T_{|T|}$ such that $\{t\} = T_1 \subset \ldots \subset T_{|T|} = T$.*

Lemma 1 *Suppose G is biconnected. Given a double v-tree T of G and a vertex $t \in T$, it is always possible to construct a sequence of nested double v-trees from $\{t\}$ to T.*

Definition 6 *A separator R of $S = \{s_1, \ldots, s_p\}$ induces the following structure in G. Graph $G[V \setminus R]$ is split in a number of connected components: those containing exactly one vertex in S are called* A-components *and those with no vertices in S*

are called B-components. *Define A_i as the vertex-set of the A-component containing s_i (when R is not proper some A-components do not exist). The vertex-sets corresponding to B-components are numbered arbitrarily as $B_1, \ldots, B_q$.*

Theorem 3 *Suppose G is biconnected. Let $\sum_{i\in S} x_i - \sum_{i\in R}(|S_i^R|-1)x_i \leq 1$ be a p-separator valid for $P_1(G)$ (by corollary 1, a facet of $P_1(G)$). A necessary condition for this p-separator to also define a facet for $P(G)$ is: for every $i \in R$ there exists a double v-tree T_i such that $T_i \cap R = \{i\}$ and $T_i \supseteq S_i^R$. Moreover, if R induces a structure without B-components this condition is also sufficient.*

Proof: Suppose that the necessary condition is satisfied. Let $a^t x \leq a_0$ be a valid inequality for $P(G)$ such that

$$F = \{x \in P(G) \mid \sum_{i\in S} x_i - \sum_{i\in R}(|S_i^R|-1)x_i = 1\} \subseteq F_a = \{x \in P(G) \mid a^t x = a_0\}$$

Consider the structure induced by R. For each $j \in \{1, \ldots, p\}$, A_j is a double v-tree. Applying lemma 1, construct a sequence of nested double v-trees from s_j to A_j. For each double v-tree T in this sequence, $\chi^T \in F$. So $a_j = a_0$ and for every $i \neq j \in A_j$, $a_i = 0$. For every vertex $i \in R$, define $T_i' = T_i \setminus (\cup_{j=1}^q B_j)$ where T_i is the double v-tree given by hypothesis. As $\chi^{T_i'} \in F$, $a_i = -(|S_i^R|-1)a_0$. We already proved that the p-separator can only be dominated by inequalities obtained by lifting coefficients in B-components. If there are no B-components the sufficient condition is proved.

Now suppose there is a vertex $i \in R$ not satisfying the necessary condition. We show that for every double v-tree T such that $i \in T$, $|S \cap T| - \sum_{i\in R\cap T} \alpha_i \leq 0$, therefore coefficient α_i can be lifted to at least $|S_i^R| - 2$.

Case(a): *There is a vertex $s_j \in S_i^R$ not contained in T.* Consider the structure induced by R. As $T \cup A_j$ is a v-tree and the p-separator is valid for $P_1(G)$, then $|S \cap (T \cup A_j)| - \sum_{i\in R\cap T} \alpha_i \leq 1$, so $|S \cap T| - \sum_{i\in R\cap T} \alpha_i \leq 0$.

Case(b): $S_i^R \subseteq T$. Let U be a maximal v-tree such that $U \subset T$, $U \cap R = \{i\}$ and $U \supseteq S_i^R$. Graph $G[V \setminus U]$ is split in connected components with vertex-sets $C_1, \ldots, C_m$, $m \geq 2$, and at least two such components have vertices in R. As T is a double v-tree, w.l.o.g. we may say that $T = U \cup C_1 \cup \ldots \cup C_{m-1} \cup D$, where $D \subseteq C_m$. Observe that $|S \cap U| - \sum_{i\in R\cap U} \alpha_i = 1$. Consider the structure induced by R. As G is biconnected, every A-component contained in component C_j is adjacent to at least two vertices in $R \cap C_j$ and at least one vertex in $R \cap C_j$ is adjacent to an A-component corresponding to a vertex in $s \in S_i^R$. So for every $j \in \{1, \ldots, m\}$, $|S \cap C_j| - \sum_{i\in R\cap C_j} \alpha_i \leq -1$ if C_j has vertices in R and $|S \cap C_j| - \sum_{i\in R\cap C_j} \alpha_i = 0$ otherwise. Finally, $|S \cap D| - \sum_{i\in R\cap D} \alpha_i \leq 0$, otherwise the p-separator would not be valid for $P_1(G)$ ($D \cup U$ is a v-tree). Therefore $|S \cap T| - \sum_{i\in R\cap T} \alpha_i \leq 0$. ∎

In graph G1 (figure 1), 2-separator $x_a + x_f - x_b - x_e - x_j \leq 1$ defines a facet of $P(G1)$, but $x_a + x_c - x_b - x_e - x_h - x_l \leq 1$ do not (α_e and α_h can be lifted to

0). 3-separator $x_a + x_f + x_g - x_b - 2x_e - 2x_k \leq 1$ defines a facet of $P(G1)$, but $x_a + x_c + x_g - x_b - 2x_e - 2x_k \leq 1$ do not (α_e can be lifted to 1). In graph G3 (figure 1), 2-separator $x_a + x_b - x_d - x_e - x_f \leq 1$ satisfies the necessary conditions of theorem 3, but it is dominated by $x_a + x_b + x_c - x_d - x_e - x_f \leq 1$ (a coefficient in a B-component can be lifted). This last inequality can also be obtained by lifting the 3-separator $x_a + x_b + x_c - 2x_d - 2x_e - 2x_f \leq 1$.

Definition 7 *The inequalities valid for $P(G)$ that can be obtained by lifting the α coefficients of a p-separator facet of $P_1(G)$ or $P_2(G)$ are called* lifted p-separators.

Lifted p-separators form a very large family and includes many facets of P. Even applying theorem 3 necessary conditions, it would be cumbersome to design separation procedures for lifted p-separators. Also, it is not easy to identify which of these inequalities are facet-defining. Therefore we give a characterization of a subfamily of lifted p-separators that turned out to be very important in our branch-and-cut.

Definition 8 *Let $S = \{s_1, \ldots, s_p\}$ and $R = \{r_1, \ldots, r_p\}$ be two disjoint vertex-sets with $p \geq 2$. Suppose that for any integers a, b, c, d in the range $\{1, \ldots, p\}$, with $a \leq b < c \leq d$, there are no two vertex-disjoint paths joining s_a to s_c and r_b to r_d (in other words, the vertices in every path from r_b to r_d define a separator of $\{s_a, s_c\}$ and vice-versa). Then the following inequality is called a p-*cross

$$\sum_{i \in S} x_i - \sum_{i \in R} x_i \leq 1$$

In graph G1, $x_e + x_j - x_d - x_g \leq 1$ is a 2-cross and $x_a + x_c + x_l + x_j - x_b - x_i - x_k - x_d \leq 1$ is a 4-cross. Most p-crosses are lifted p-separators. 3-cross $x_a + x_c + x_k - x_b - x_l - x_d \leq 1$ is a lifting of 3-separator $x_a + x_c + x_k - 2x_b - x_f - x_l - x_d \leq 1$ facet of $P_1(G1)$ or $-x_b - x_l - x_d + x_a + x_c + 2x_e + x_f + x_k \leq 4$ facet of $P_2(G1)$.

Theorem 4 *The p-crosses are valid for P.*

Proof: By induction on p. The 2-crosses are clearly valid. Suppose that $\sum_{i=1}^{p} x_{s_i} - \sum_{i=1}^{p} x_{r_i} \leq 1$ is a p-cross with $p \geq 3$. By induction hypothesis, $(p-1)$-crosses $\sum_{i=1}^{p-1} x_{s_i} - \sum_{i=1}^{p-1} x_{r_i} \leq 1$ and $\sum_{i=2}^{p} x_{s_i} - \sum_{i=2}^{p} x_{r_i} \leq 1$ are valid. $x_{s_1} + x_{s_p} - x_{r_1} - x_{r_p} \leq 1$ is a 2-cross. Summing these 3 inequalities, dividing the result by 2, and rounding down the right-hand-side, the desired p-cross is obtained. ■

Theorem 5 *If G is biconnected, every 2-cross defines a facet of $P(G)$.*

Proof: Let $x_{s_1} + x_{s_2} - x_{r_1} - x_{r_2} \leq 1$ be a 2-cross and $a^t x \leq a_0$ be a valid inequality for $P(G)$ such that

$$F = \{x \in P(G) \mid x_{s_1} + x_{s_2} - x_{r_1} - x_{r_2} = 1\} \subseteq F_a = \{x \in P(G) \mid a^t x = a_0\}$$

We show that $a^t x \leq a_0$ must be an scalar multiple of this 2-cross. As G is biconnected, there are two vertex-disjoint paths, except in the endpoints, joining r_1 to r_2. Let I_1 and I_2 be the set of vertices of any two such paths. As I_1 and I_2 are separators of S, it is not possible to have $S \subseteq I_1$ or $S \subseteq I_2$.

Case (a): I_1 or I_2 is a proper separator of S.

Suppose I_1 is a proper separator of S and consider the induced structure. As G is biconnected, each component is adjacent to at least two vertices in I_1. We assert that at least one of the two A-components is adjacent to both r_1 and r_2. If this is not true, the path corresponding to I_2 must pass through a B-component and I_1 would not be a separator of S. Suppose A_2 is an A-component adjacent to r_1 and r_2.

Case (a.1): A_1 is adjacent to a vertex in $I_1' = I_1 \setminus \{r_1, r_2\}$. In this case, $C_1 = A_1 \cup I_1' \cup \{B_k \,|\, B_k \text{ is adjacent to } I_1'\}$ and $C_2 = A_2$ are double v-trees. Construct a sequence of nested double v-trees from s_1 to C_1 and from s_2 to C_2. For each T in these sequences, $\chi^T \in F$. So $a_{s_1} = a_{s_2} = a_0$ and $a_i = 0$ if $i \in (C_1 \cup C_2) \setminus S$. As $\chi^{C_1 \cup C_2 \cup \{r_1\}}$ and $\chi^{C_1 \cup C_2 \cup \{r_2\}} \in F$, $a_{r_1} = a_{r_2} = -a_0$. Each component B_k not adjacent to I_1' must be adjacent to both r_1 and r_2. Let j be a vertex in B_k adjacent to r_1. Construct a sequence of nested double v-trees from j to B_k. For each T in this sequence $C_1 \cup C_2 \cup \{r_1\} \cup T$ is also a double v-tree. So $a_i = 0$, for all $i \in B_k$.

Case (a.2): A_1 is not adjacent to I_1'. In this case, A_1 is adjacent to r_1 and r_2. Let $C_1 = A_1$, $C_2 = A_2$ and proceed similar to case a.1, concluding that $a_{s_1} = a_{s_2} = a_0$, $a_i = 0$ if $i \in (C_1 \cup C_2) \setminus S$ and $a_{r_1} = a_{r_2} = -a_0$. Now define C_3 as $I_1' \cup \{B_k \,|\, B_k \text{ is adjacent to } I_1'\}$. Let j be a vertex in I_1' adjacent to r_1. Construct a sequence of nested double v-trees from j to C_3. For each T in this sequence $C_1 \cup C_2 \cup \{r_1\} \cup T$ is also a double v-tree. So $a_i = 0$, for each $i \in C_3$. B-components adjacent to both r_1 and r_2 are treated in a similar way.

Case (b): I_1 and I_2 are not proper separators of S.

Suppose I_1 contains s_1 and I_2 contains s_2. Consider the structure induced by separator I_1. As I_2 contains s_2, A_2 is adjacent to r_1 and r_2. Let $C_1 = I_1' \cup \{B_k \,|\, B_k \text{ is adjacent to } I_1'\}$, $C_2 = A_2$ and proceed similar to case a.1. B-components adjacent to both r_1 and r_2 are treated like in a.1. ■

Using similar techniques it is possible to prove that under mild conditions p-crosses with $p \geq 3$ also define facets of $P(G)$. For instance, supposing G is biconnected and planar, all p-crosses are facet-defining.

4 Separation Procedures

The 2-separators can be separated by applying the min-cut algorithm over a suitable network. When $p \geq 3$, the procedures to find violated p-separators inequalities are harder to devise. One heuristic procedure to find such a p-separator is by the

successive application of the 2-separator procedure. Note that if $p \geq 4$ there is the additional complication of calculating proper α coefficients.

The p-crosses can be separated in polynomial time over any graph G, for fixed p. This follows from definition 8, since there exists an $O(|V|.|E|)$ algorithm to determine if it is possible to connect two given pairs of vertices by vertex-disjoint paths (Shiloach [6]). However, far better p-cross separation schemes may exist if the particular topological structure of G is considered. If G is planar and supposing that no 2-separator is violated, a violated p-cross only occurs when the sequence $s_1, r_1, \ldots, s_p, r_p$ happen in this cyclic order on some face f of G. By this property, p-cross inequalities can be separated in $O(|V|)$ over planar graphs, by dynamic programming.

5 Computational Results

The branch-and-cut for 2VPST was implemented in C++ using the ABACUS framework [7] and CPLEX 3.0 LP solver, running on a Sun ULTRA workstation. The code used 2-separators, 3-separators (heuristically separated) and p-crosses.

In all experiments described in table 1, G is a square grid graph (grid graphs are typical in circuit layout applications). Two kind of instances were generated:

- Unweighted instances with $|N_1| = |N_2| = |V|/4$. The terminals in each of these sets are randomly chosen among the vertices of G. Vertices in N_1 have weight -1 and vertices in N_2 have weight 1. These instances are referenced as u_n_$Inst$, where $n = |V|$ and $Inst$ is the instance number. The value z^* gives the number of unconnected terminals in an optimal solution.

- Weighted instances. For each unweighted instance we generated a corresponding weighted instance, by picking an integer w_i from an uniform distribution in the range $[-10, -1]$ if $i \in N_1$ and from $[1, 10]$ if $i \in N_2$. The value z^* gives the sum of the absolute weights of the unconnected terminals in an optimal solution. Weighted instances are referenced as w_n_$Inst$.

In table 1, the LB_r column presents the lower bound obtained in the root node, z^* is the value of the optimal integer solution, **Nd** is the number of nodes explored in the branch-and-bound tree, **LPs** is the total number of solved LPs and **T** is the total cpu time in seconds spent by the algorithm. Remark that LB_r bounds includes p-crosses and 3-separators. The lower bounds given by the linear relaxation of formulation F alone are rather weak, for example, in instance u1600_1 it is 8.7.

Figure 2 shows an optimal solution of instance u1600_1, the black boxes represents terminals in N_1, the empty boxes are the terminals in N_2, the full lines are edges of a spanning tree for $G[T_1]$ and the dotted lines a spanning tree for $G[T_2]$.

Instance	LB_r	z^*	Nd	LPs	T	Instance	LB_r	z^*	Nd	LPs	T
u1600_1	45.7	46	4	96	20	w1600_1	222.0	222	2	35	9
u1600_2	40.0	40	2	40	8	w1600_2	218.0	218	1	25	5
u2500_1	46.0	46	2	56	33	w2500_1	233.0	233	2	37	14
u2500_2	55.0	55	3	49	32	w2500_2	227.5	228	3	39	17
u3600_1	76.5	77	2	45	41	w3600_1	364.0	364	2	48	33
u3600_2	74.0	74	4	57	50	w3600_2	311.0	311	2	74	54
u4900_1	99.1	100	4	116	170	w4900_1	420.0	420	2	77	130
u4900_2	92.0	92	3	66	108	w4900_2	400.0	400	3	53	90
u6400_1	104.0	104	3	126	443	w6400_1	436.0	436	3	106	295
u6400_2	127.0	128	4	123	322	w6400_2	534.0	534	3	128	367
u8100_1	131.5	132	4	217	1207	w8100_1	549.5	550	4	113	563
u8100_2	115.0	115	6	125	713	w8100_2	467.0	467	3	82	264
u10000_1	159.8	161	8	120	676	w10000_1	685.0	685	3	60	345
u10000_2	151.5	152	4	157	991	w10000_2	632.5	634	7	116	621

Table 1: Branch-and-Cut Results

The code turned out to be robust under changes on terminal placement, terminal density and weight distribution. In fact, we used the same set of parameters for solving all instances in table 1, weighted and unweighted. An interesting experience is to generate a random terminal placement avoiding the most obvious causes of infeasibility: terminals on the outer face, two pairs of opposite terminals crossed over the same grid square and terminals blocked by its neighbors. An optimal solution of such an instance (called t1600_1) with 800 terminals and unitary weights is shown in figure 3. The unconnected terminals are in coordinates (4,33), (8,21), (33,36) and (34,15). This solution was obtained in 21 seconds.

Appendix

Proof of theorem 2

Lemma *(Sequential Maximum Lifting Lemma, pages 261-263 of [5]) Let P be a polyhedron in $\mathbb{R}^n$. Let N be the set of indices $\{1,\ldots,n\}$ and Z a subset of N. Define P^Z as the polyhedron $P \cap \{x \in \mathbb{R}^n | x_i = 0, i \in Z\}$. Suppose $\sum_{i \in N \setminus Z} a_i x_i \leq a_0$ is a valid inequality for P^Z defining a face of dimension d. Let $\sum_{i \in N \setminus Z} a_i x_i + \sum_{i \in Z} m_i x_i \leq a_0$ be any inequality obtained by a sequential maximum lifting of one coefficient in Z at a time (in any order). This lifted inequality defines a face of P of dimension $d + |Z_0|$. If the original inequality defines a facet of P^Z, the lifted inequality defines a facet of P.*

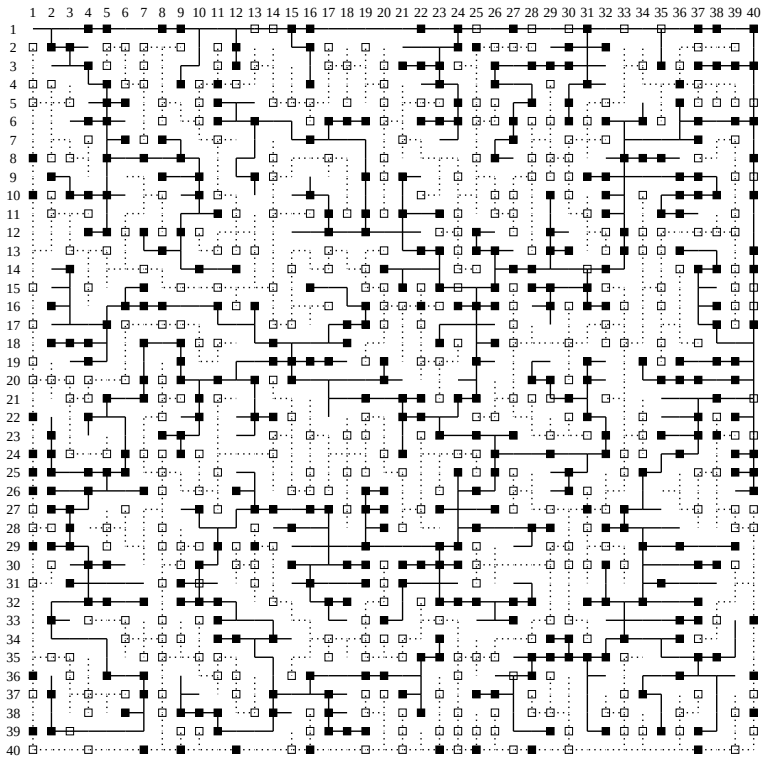

Figure 2: An optimal solution of instance u1600_1, $z^* = 46$.

Definition *Let T be a v-tree of G and t a vertex in T.* A sequence of nested v-trees from t to T *is any sequence of v-trees $T_1, \ldots, T_{|T|}$ such that $\{t\} = T_1 \subset \ldots \subset T_{|T|} = T$. Note that given v-tree T and t, is always possible to construct such a sequence.*

Theorem 2 *For each S and each $R \in \Delta(S)$, there is at least one p-separator inequality which is valid and facet-defining for $P_1(G)$.*

Proof: Consider the structure induced by R. Inequality $\sum_{i \in S} x_i \leq 1$ is valid for $P_1(G)^{Z_0}$, where $Z_0 = R \cup (\cup_{j=1}^{q} B_j)$. Define $F = \{x \in P_1(G)^{Z_0} | \sum_{i \in S} x_i = 1\}$. For each $j \in \{1, \ldots, p\}$, A_j is a v-tree. Construct a sequence of nested v-trees from s_j to A_j. For every T in these sequences, $\chi^T \in F$. So there are $\sum_{j=1}^{p} |A_j|$ affine independent vectors in F and $\sum_{i \in S} x_i \leq 1$ defines a facet of $P_1(G)^{Z_0}$.

Number the vertices of R as $\{r_1, \ldots, r_{|R|}\}$. Define $R_0 = \emptyset$ and R_j as the set $\{r_1, \ldots, r_j\}$. For each B-component B_k, define $f(B_k)$ as the smaller number of a vertex in R adjacent to B_k. Define $W_j = R_j \cup \{B_k | f(B_k) \leq j\}$ and $Z_j = Z_0 \setminus W_j$. We show that for each $j \in \{0, \ldots, |R|\}$, there is an inequality $\sum_{i \in S} x_i - \sum_{i \in R_j} \alpha_i x_i \leq 1$, where each α_i is integer and satisfies $\alpha_i \geq |S_i^R| - 1$, defining a facet of $P_1(G)^{Z_j}$. This is true when $j = 0$. Suppose $1 \leq j \leq |R|$. Pick the inequality corresponding to $j - 1$. First lift the coefficient of r_j. As there is a v-tree T_j such

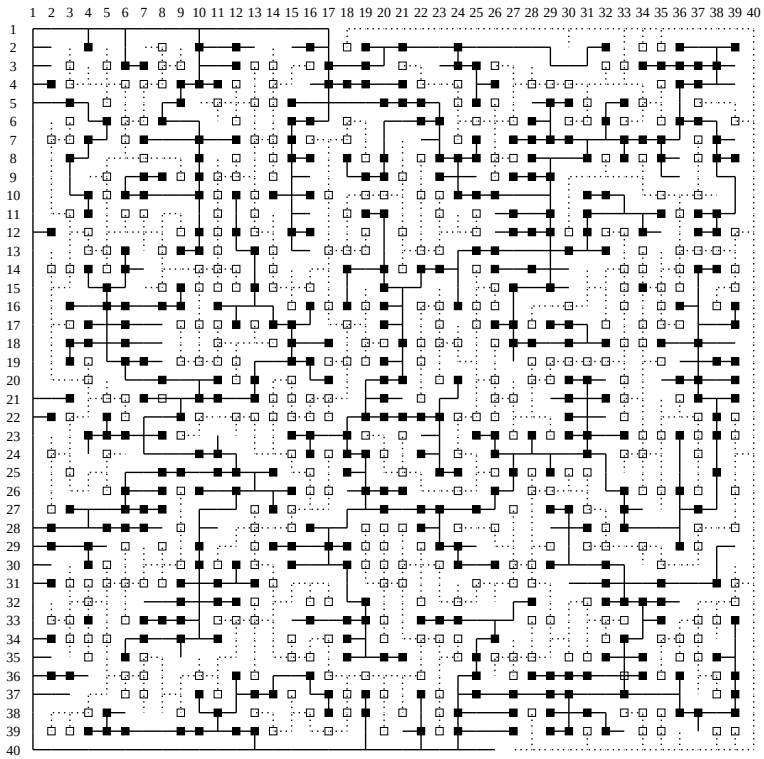

Figure 3: An optimal solution of instance t1600_1, $z^* = 4$.

that $T_j \cap R = \{r_j\}$, $T_j \cap S = S_j^R$ and $T_j \cap (\cup_{j=1}^{q} B_j) = \emptyset$, $\alpha_j \geq |S_j^R| - 1$. As the lifting is maximum, α_j is integral and there exists a v-tree $U \subseteq (\cup_{j=1}^{p} A_j \cup W_{j-1} \cup \{r_j\})$ such that $r_j \in U$ and $|S \cap U| - \sum_{i \in R_j \cap U} \alpha_i = 1$. For each B-component B_k such that $f(B_k) = j$, take a vertex $t \in B_k$ adjacent to r_j. Construct a sequence of nested v-trees from t to B_k. Make a maximum lifting of the coefficients of vertices in B_k in the same order this vertices are introduced in the sequence. Since $U \cup T$ is a v-tree for every T in the sequence, all such coefficients must be zero. Repeating this procedure until $j = |R|$, a p-separator defining a facet of $P_1(G)$ is obtained. ■

Correctness of procedure Calculate α

Theorem *The p-separator inequalities* $\sum_{i \in S} x_i - \sum_{i \in R} \alpha_i x_i \leq 1$ *where the* α *coefficients are calculated by procedure Calculate* α *are valid for* $P_1(G)$.

Proof: Let U be a v-tree and let $Q = R \cap U$. We show that $|S \cap U| - \sum_{i \in Q} \alpha_i \leq 1$. Assume that Q is not empty, otherwise this inequality is clearly true. Sort the vertices in Q as $q_1, \ldots, q_{|Q|}$, using the same order in which these vertices were considered in procedure Calculate α. For each $i \in \{1, \ldots, |Q|\}$, define Q_i as the set $\{q_1, \ldots, q_i\}$ and $S_{Q_i}^R$ as $\cup_{j \in Q_i} S_j^R$.

We show that for each $i \in \{1, \ldots, |Q|\}$, $\gamma_i = |S_{Q_i}^R \cap U| - \sum_{i \in Q_i} \alpha_i \leq 1$. Note that γ_i can only be equal to 1 if a vertex j in Q_i has $\alpha_j = |S_j^R| - 1$. This is true

when $i = 1$. Suppose $1 < i \leq |Q|$. If $\alpha_i = |S_i^R|$, then $\gamma_i \leq \gamma_{i-1} \leq 1$. Suppose $\alpha_i = |S_i^R| - 1$. The procedure assures that if there exists a vertex $j \in Q_{i-1}$ with $\alpha_j = |S_j^R| - 1$ then $S_i^R \cap S_j^R \neq \emptyset$, and $\gamma_i \leq \gamma_{i-1} \leq 1$. If there is no such vertex $\gamma_{i-1} \leq 0$, so $\gamma_i \leq 1$. ■

Proof of lemma 1

Lemma 1 *Suppose $G = (V, E)$ is biconnected. Given a double v-tree T of G and a vertex $t \in T$, it is always possible to construct a sequence of nested double v-trees $\{t\} = T_1 \subset \ldots \subset T_{|T|} = T$ from $\{t\}$ to T.*

Proof: If $T = \{t\}$ the result is immediate. Otherwise, suppose that a partial sequence $T = T_{|T|} \supset \ldots \supset T_i$, $|T| \geq i > 1$, is already formed. Vertices in double v-tree T_i, different from t, but adjacent to a vertex in $V \setminus T_i$ will be called *candidate vertices* (if $T = V$, all vertices in $V - t$ are candidates). As G is biconnected, there is at least one candidate vertex. We show that is always possible to choose a candidate to be removed from T_i to form double v-tree T_{i-1}.

Let v_0 be a candidate. If $T_i - v_0$ is a v-tree, v_0 is chosen. Otherwise $G[T_i - v_0]$ is separated into $j \geq 2$ connected components with vertex-sets $C_1^1, \ldots, C_j^1$. As T_i is a v-tree, at least one vertex in C_j^1 is adjacent to v_0. As G is biconnected, at least one vertex in each C_j^1 is adjacent to $V \setminus T_i$. Adjust notation for C_1^1 to be a set not containing t. Let v_1 be a candidate vertex in C_1^1. $G[C_1^1 - v_1]$ is separated into $j \geq 0$ connected components with vertex-sets $C_1^2, \ldots, C_j^2$. As C_1^1 is a v-tree, at least one vertex in each C_j^2 is adjacent to v_1. As G is biconnected, at least one vertex in each C_j^2 is adjacent to v_0 or to $V \setminus T_i$. If at least one vertex in each C_j^2 is adjacent to v_0 (or $j = 0$), v_1 is chosen. Otherwise, adjust notation for C_1^2 to be a set containing a candidate vertex.

By repeating this procedure, we get a sequence $v_1, \ldots, v_k$ of candidate vertices contained in smaller sets C_1^k. If each component of $G[C_1^k - v_k]$ is adjacent to $T_i \setminus C_1^k$, v_k is the chosen candidate and $T_i - v_k$ is a v-tree. In the worst case, the procedure is repeated until $C_1^k = \{v_k\}$. ■

References

[1] M. Grötschel, A. Martin, R. Weismantel, *Packing Steiner Trees: polyhedral investigations*, Mathematical Programming, Vol. 72, 101-123, 1996.

[2] M. Grötschel, A. Martin, R. Weismantel, *Packing Steiner Trees: a cutting plane algorithm and computational results*, Mathematical Programming, Vol. 72, 125-145, 1996.

[3] B. Korte, H. Prömel and A. Steger, *Steiner Trees in VLSI-Layout*, in B. Korte, L. Lovász, H. Prömel, A. Schrijver (Eds.) "Paths, Flows and VLSI-Layout", Springer-Verlag, Berlin, 1989.

[4] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, John Wiley & Sons, Chichester, 1990.

[5] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience, 1988.

[6] Y. Shiloach, *A Polynomial Solution to The Undirected Two Paths Problem*, Journal of the ACM, Vol. 27, No.3, 445-456, 1980.

[7] S. Thienel, *ABACUS - A Branch-And-Cut System*, PhD thesis, Universität zu Köln, 1995.