On Mimicking Networks*

Shiva Chaudhuri† K.V. Subrahmanyam† Frank Wagner‡ Christos D. Zaroliagis†

December 4, 1997

Abstract

A *mimicking network* for a k-terminal network, N, is one whose realizable external flows are the same as those of N. Let $S(k)$ denote the minimum size of a mimicking network for a k-terminal network. We prove the following results (the values in brackets are the previously best known results): $S(4) = 5$ $[2^{16}]$, $S(5) = 6$ $[2^{32}]$. For bounded treewidth networks we show $S(k) = O(k)$ $[2^{2^k}]$, and for outerplanar networks we show $S(k) \leq 10k - 6$ $[k^2 2^{k+2}]$.

1 Introduction

Network flows is a problem domain of fundamental importance in computer science and other areas (e.g., operations research, engineering, etc.), having a wealth of theoretical and practical applications [1]. One of the central (and classical) problems in network flows is the characterization of the flow behavior of multiterminal networks, i.e., networks with $k > 2$ terminals, first motivated and solved by Gomory and Hu [8] and later improved and simplified by many others (see e.g., [6, 7, 14]). The Gomory-Hu approach, as well as its subsequent improvements and simplifications, deal mainly with the case where every vertex of the network is a potential candidate for being a terminal. However, there may be cases where the number of terminals is much smaller than the number of vertices in the network. For example, in computing max-flows/min-cuts on networks that can be decomposed into subnetworks that share a small number of vertices with the rest of the network. Such a situation appears in hierarchical methods for integrated circuit layout problems: a circuit (network) is described and processed as a collection of components (subnetworks), where each component is in turn described and handled succinctly, i.e., in terms of the pins that connect it to the rest of the circuit [12].

Under this perspective, there is a recent, renewed interest on the problem of characterizing the flow behavior of networks with a small (usually constant) number of terminals [10]. More precisely, it was shown in [10] that for a k-terminal network G, there is a set of 2^k inequalities that characterize the feasible flow values at the terminals (the external flows). Moreover, it was shown that there exists a network $M(G)$ with 2^{2^k} vertices, of which k are terminals, that has the same feasible external flows as G. The network $M(G)$ is called the *mimicking network* of G. For constant k this implies that the size of $M(G)$ is bounded by a (maybe large) constant. Mimicking networks constituted the main building block in the development of a linear time algorithm for computing a maximum s-t flow in a bounded treewidth network [10] and for obtaining an optimal solution to the all-pairs min-cut problem in the same class of networks [2].

*This work was partially supported by the EU ESPRIT LTR Project No. 20244 (ALCOM-IT).

†Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. E-mail: {shiva, kv, zaro}@mpi-sb.mpg.de.

‡Institut für Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany. This author was supported by the Heisenberg program of the Deutsche Forschungsgemeinschaft. E-mail: wagner@inf.fu-berlin.de.

A natural question is whether there are more efficient constructions of mimicking networks, i.e., constructions such that $|M(G)|$ does not depend double-exponentially on k, as this would immediately lead to more practical algorithms for all problems whose solution is based on the existence of small mimicking networks.

This question was partially answered for outerplanar networks. More precisely, in [2] it was proved that the mimicking network $M(G_o)$ of a k-terminal outerplanar network G_o has size $k^2 2^{k+2}$, and moreover $M(G_o)$ is also outerplanar. The outerplanarity of the resulting mimicking network was crucial in the development of better algorithms for computing an s-t min-cut and all-pairs min-cuts in planar networks [2] (an improvement that cannot be achieved using the construction of mimicking networks given in [10]).

In this paper, we make a step forward in constructing smaller mimicking networks. We have two types of results.

The first result concerns the case of general undirected k-terminal networks for specific values of k; namely, for $k = 3, 4, 5$. We show that any 3-terminal network has a mimicking network consisting of 3 vertices, and any 4-terminal (resp. 5-terminal) network has a mimicking network consisting of 5 (resp. 6) vertices (Section 4). We also prove that these constructions are *optimal* by showing that, for any $k \geq 4$, there exists a k-terminal network for which the mimicking network must have at least one vertex in addition to the k terminals. We further provide necessary and sufficient conditions for a vector of min-cut values to be realizable, i.e., whether the values of a given vector represent min-cut values of a 3,- 4-, and 5-terminal network. Our approach is simple and provides a new insight in the construction of mimicking networks. Our methods are based on a result of independent interest (Section 3) concerning values of multisets of min-cuts in arbitrary k-terminal networks that seems to have further applications. In particular, it can be used to prove a conjecture by Lengauer [12, Sec. 6.1.5] regarding the problem of mimicking the cut properties of a hypergraph by a graph, a problem arising in circuit partitioning (see Section 7 for details).

The second result concerns the case of (directed or undirected) bounded treewidth networks. Informally, the treewidth t is a parameter that indicates how close the structure of the network is to a tree (see Section 2 for a formal definition). The class of bounded treewidth networks – pioneered by the work of Robertson and Seymour [13] and continued by many others (see e.g., [3, 4, 5]) – includes outerplanar networks ($t = 2$), series-parallel networks ($t = 2$), and networks with bounded bandwidth or cutwidth [3, 5]. For any k-terminal bounded treewidth network, we show that there is a mimicking network of size $O(k)$, where the constant in the Big-O depends on the treewidth (Section 5). For the special case of outerplanar networks, we provide further improvements. We show that the size of the (outerplanar) mimicking network in this case is much smaller than the one we get with the general method of bounded treewidth networks (Section 6). For both the bounded treewidth and the outerplanar cases, our results are optimal up to constant factors.

2 Preliminaries

The treatment in this section follows that in [2]. A *network* is a (directed or undirected) graph $G = (V, E)$ with a nonnegative real capacity c_e associated with each edge $e \in E$. The *terminals* of G are the elements of a distinguished subset, $Q = \{q_1, q_2, \ldots, q_k\}$, of its vertices. A *flow* in G is an assignment of a nonnegative real value $f_e \leq c_e$ to each edge e such that the net flow out of each non-terminal vertex is zero, where the net flow out of a vertex is the sum of flows on edges leaving the vertex minus the sum of flows on edges entering the vertex. An *external flow* $x = (x_1, \ldots, x_k)$ is an assignment of a real value x_i to each terminal $q_i \in Q$, $1 \leq i \leq k$. A *realizable external flow* is an external flow such that there exists a flow in which the net flow out of each terminal q_i is x_i. A *cut* $(S, \overline{S})$ is a partition of the vertices of G into two subsets S and $\overline{S} = V \setminus S$; S is called the *defining* *subset* of the cut. The *capacity* of the cut $(S, \overline{S})$ is the sum of capacities of edges going from vertices in S to vertices in $\overline{S}$. For a subset R of Q, an *R-separating cut* is a cut $(S, \overline{S})$ where $Q \cap S = R$. A *minimum* R-separating cut (or just *min-cut* when R is well understood from the context) is an R-separating cut of minimum capacity. We denote the capacity of a minimum R-separating cut by b_R. The sum of the net flows out of the terminals in R is called the *R-value* of a flow. A *maximum R-flow* is a flow of maximum R-value. If $Q = \{s, t\}$, an *s-t max-flow* is a maximum $\{s\}$-flow, and an *s-t min-cut* is a minimum $\{s\}$-separating cut.

Let $(S_1, \overline{S_1})$ and $(S_2, \overline{S_2})$ be two min-cuts in a network G where $Q \cap S_1 = R_1$ and $Q \cap S_2 = R_2$. We call these min-cuts *equal* if $R_1 = R_2$, in the case where G is directed, or if $R_1 = R_2$ or $R_1 = Q \setminus R_2$, in the case where G is undirected. Two min-cuts that are not equal are called *distinct*. A k-terminal directed network has 2^k distinct min-cuts (two of which are trivial), and a k-terminal undirected network has $2^{k-1} - 1$ distinct min-cuts (since half of the $2^k - 2$ min-cuts are equal).

Let G be a network with terminal set Q. A network $M(G)$ with terminal set Q' is a *mimicking network* for G if there exists a bijection between Q and Q' such that every realizable external flow in G is also realizable in $M(G)$, and vice versa. The following lemma is a reformulation of the results in [10].

Lemma 1 *Let G and G' be two networks with the same terminal set Q. Then, every realizable external flow in G is also realizable in G' (and vice versa) iff the capacities of the minimum R-separating cuts are equal for all $R \subseteq Q$.*

To compute a mimicking network of a network G using either of the approaches in [2, 10], as well as the approaches described in this paper, we have to find the 2^k minimum R-separating cuts. This is done by the standard method of solving 2^k s-t max-flow/min-cut problems in a network G' which consists of G augmented with two vertices s and t and edges of infinite capacity from s to every vertex in R and from every vertex in $Q \setminus R$ to t. However, G' may not preserve the structural properties (e.g., outerplanarity, planarity, bounded treewidth) that G may have and which may be desirable for reasons of efficiency. In [2, Lemma 2.3] it was shown how to overcome this problem and find the 2^k minimum R-separating cuts by solving $k^2 2^k$ s-t max-flow/min-cut problems in G. Therefore, we assume henceforth that the 2^k minimum R-separating cuts are provided to us with the input.

A *separator* in a graph is a set of vertices which disconnects the graph. Let $G = (V(G), E(G))$ be a graph and $U \subseteq V(G)$ be a separator. Let V_1 and V_2 be a partition of $V(G) \backslash U$ such that in $G \backslash U$, there is no path from $v \in V_1$ to $w \in V_2$ for any v, w. Let G_1 and G_2 be graphs such that $V(G_1) = V_1 \cup U$, $V(G_2) = V_2 \cup U$, and $E(G_1) \cap E(G_2) = \emptyset$, $E(G_1) \cup E(G_2) = E(G)$. We say G_1 and G_2 are subgraphs obtained by *splitting G at U*.

Let $Q \subseteq V(G)$ be the set of terminals of G. Let $Q_1 = V_1 \cap Q$ and $Q_2 = V_2 \cap Q$. Suppose we construct mimicking networks $M(G_1)$ for G_1 at terminals $Q_1 \cup U$ and $M(G_2)$ for G_2 at terminals $Q_2 \cup U$. We join $M(G_1)$ and $M(G_2)$ by identifying, for each $u \in U$, the vertex in $M(G_1)$ corresponding to u with the vertex in $M(G_2)$ corresponding to u. It was shown in [10] that the resulting network is a mimicking network for G at terminals Q. We call this process *glueing* the mimicking networks together. The notions of *splitting* and *glueing* will be used extensively in Sections 5 and 6.

A *tree decomposition* of a (directed or undirected) graph $G = (V(G), E(G))$ is a pair (X, T), where $T = (V(T), E(T))$ is a tree, X is a family $\{X_i : i \in V(T)\}$ of subsets of $V(G)$ that cover $V(G)$, and the following conditions hold:

- (*edge mapping*) $\forall (v, w) \in E(G)$, there exists an $i \in V(T)$ with $v \in X_i$ and $w \in X_i$.
- (*continuity*) $\forall i, j, k \in V(T)$, if j lies on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$, or equivalently: $\forall v \in V(G)$, the nodes $\{i \in V(T) : v \in X_i\}$ induce a connected subtree of T.

The *width* of the tree decomposition is $\max_{i\in V(T)}|X_i|-1$. The *treewidth* of G is the minimum width over all possible tree decompositions of G.

Bodlaender [4] gave an $O(n)$ time algorithm that tests, for all constant $t \in \mathbb{N}$, whether a given n-vertex graph G has treewidth $\leq t$ and if so, computes a tree decomposition (X,T) of G of width $\leq t$, where $|V(T)| = n-t$. Furthermore, (X,T) can be converted into another tree decomposition (X',T') with width t, such that T' is binary and $|V(T')| \leq 2(n-t)$. We call such a tree decomposition a *binary tree decomposition.*

Let G be an n-vertex graph of constant treewidth and let (X,T) be its tree decomposition of constant width. The edge mapping condition ensures that the endpoints of each edge in G appear together in some set $X_i \in X$, belonging to vertex i of T. Thus, in a sense, each edge is represented in at least one vertex of T. For our purposes, we need to explicitly associate each edge of G with exactly one vertex of T. We will, therefore, compute an *augmenting function* $h: E(G) \to V(T)$, satisfying the property that both endpoints of an edge are present in the set belonging to the vertex that the edge is mapped to by h. More precisely, $\forall (v,w) \in E(G), \{v,w\} \subseteq X_{h(v,w)}$. Any augmenting function will suffice for our purposes. It is easy to compute one such function, by doing a traversal of T and assigning $h(v,w) = i$ for each $i \in V(T)$, if $\{v,w\} \subseteq X_i$, $(v,w) \in E(G)$ and $h(v,w)$ has not yet been assigned a value. This takes time proportional to $\sum_{i\in V(T)}|X_i|^2$, which is $O(n)$, since the tree decomposition is of constant width. The resulting tree decomposition with the values $h(v,w), \forall(v,w) \in E(G)$, is called an *augmented tree decomposition.* The discussion above is summarized as the following result.

Lemma 2 *Given an n-vertex graph G of constant treewidth t, we can compute in $O(n)$ time an augmented binary tree decomposition of G of width t.*

3 A Fundamental Lemma

In the following, let $w(c)$ denote the capacity of a cut c belonging to a (multi)set C of cuts, and let $w(C) = \sum_{c\in C} w(c)$. The next lemma plays a central role in the paper.

Lemma 3 *Let $G = (V,E)$ be a network with k terminals $q_1, \ldots q_k$. Let $M = \{m_1, \ldots m_p\}$ be a multiset of min-cuts in G and let δ_{ij} be the number of m_i's, that separate q_i and q_j. Let N be a second multiset of min-cuts with $N = \{n_i^r \mid i \in \{1,\ldots,k\}, r \in \{1,\ldots,u_i\}\}$, where each n_i^r, $1 \leq r \leq u_i$, separates q_i from all other terminals. If for all i,j, $u_i + u_j \leq \delta_{ij}$, then $w(N) \leq w(M)$.*

Proof. The intersection of all min-cuts in M induce a partition of G into (disjoint) subsets U of V, which we call *parts*. We consider each such part U as a 0-1 vector of length p, where $U[i] = 1$ if U is a subset of the defining subset of m_i, and $U[i] = 0$ otherwise.

Let τ_i^1 be the part that contains q_i (such a part always exists). Define $\tau_i^r = \{U \mid H(\tau_i^1, U) \leq r-1\}$, for $r \in \{1,\ldots,u_i\}$, where H is the Hamming distance of the two vectors (i.e., the number of bits in which they differ) corresponding to the parts. The sets τ_i^r satisfy the following two claims.

Claim 1: For every $1 \leq i \leq k$, $\tau_i^1 \subseteq \tau_i^2 \subseteq \ldots \subseteq \tau_i^{u_i}$.

Claim 2: For any $i \neq j$, any $r \in \{1,\ldots,u_i\}$ and any $s \in \{1,\ldots,u_j\}$, $\tau_i^r \cap \tau_j^s = \emptyset$.

The first claim follows directly from the definition of τ_i^r. For the second claim, assume that there is a pair τ_i^r and τ_j^s whose intersection is not the empty set. This means that there is a part from the partition induced by M which simultaneously has Hamming distance $r-1 \leq u_i - 1$ from τ_i^1 and $s-1 \leq u_j - 1$ from τ_j^1. By the triangle inequality this in turn implies that $H(\tau_i^1, \tau_j^1) \leq u_i + u_j - 2$. But this contradicts the fact that $H(\tau_i^1, \tau_j^1) = \delta_{ij} \geq u_i + u_j$. Hence, Claim 2 is also true.

To complete the proof of the lemma, it suffices to show that

$$w(N) \leq w(N') := w\left(\{\tau_i^r \mid i \in \{1,\ldots,k\}, r \in \{1,\ldots u_i\}\}\right) \leq w(M).$$

We first show $w(N) \leq w(N')$. It follows from Claim 2, that every τ_i^r contains only a single terminal, namely q_i. Thus, $w(\tau_i^r) \geq w(n_i^r)$, for all i and r, since n_i^r is a min-cut. Hence, $w(N) \leq w(N')$.

We now show $w(N') \leq w(M)$. By Claim 1, the set N' induces a partition of V into (disjoint) vertex sets ${\tau_i}^1, {\tau_i}^2 \setminus {\tau_i}^1, \ldots, {\tau_i}^{u_i} \setminus {\tau_i}^{u_i-1}$, $i = 1,\ldots,k$, plus the set of the remaining vertices, say V^*. For convenience, we assume in the following that $\tau_i^0 = \emptyset$. There are three types of edges that are cut by the cuts in N'.

Type-1 edges: are those edges (x,y) with x in some $\tau_i^r \setminus \tau_i^{r-1}$ and y in some $\tau_i^s \setminus \tau_i^{s-1}$, $r \neq s$, $r,s \in \{1,\ldots,u_i\}$.

Type-2 edges: are those edges (x,y) with x in some $\tau_i^r \setminus \tau_i^{r-1}$ and y in some $\tau_j^s \setminus \tau_j^{s-1}$, where $i \neq j$, $r \in \{1,\ldots,u_i\}$, and $s \in \{1,\ldots,u_j\}$.

Type-3 edges: are those edges (x,y) with x in some $\tau_i^r \setminus \tau_i^{r-1}$ and y in V^*, where $i \in \{1,\ldots,k\}$ and $r \in \{1,\ldots,u_i\}$.

Let us count how many times the above edges are cut by (the cuts in) N' and M. A type-1 edge is cut by N' exactly $s-r$ times (assuming w.l.o.g. that $s > r$). By the triangle inequality we have $H(x,y) \geq H(y,q_i) - H(x,q_i) = s - r$. As $H(x,y)$ is exactly the number of times such an edge is cut by M, we have shown the desired inequality in this case. A type-2 edge is cut by N' exactly $u_i - (r-1) + u_j - (s-1) = u_i + u_j - r - s + 2$ times, since it is cut only by those τ_i^a with $r \leq a \leq u_i$ and by those τ_j^b with $s \leq b \leq u_j$. By the generalized triangle inequality we have $H(x,y) \geq H(q_i,q_j) - H(q_i,x) - H(q_j,y) = \delta_{ij} - (r-1) - (s-1) = \delta_{ij} - r - s + 2 \geq u_i + u_j - r - s + 2$, where the last inequality follows by the assumption of the theorem. Therefore, the type-2 edges are cut by M at least as many times as they are cut by N'. For the type-3 edges, we can show (similarly to the type-1 edges) that they are cut by M at least as many times as they are cut by N'.

Hence, in total M cuts at least so many edges as N' does, and consequently $w(N') \leq w(M)$. ∎

4 Mimicking k-terminal networks for $k = 3, 4, 5$

Recall that a k-terminal undirected network has $2^{k-1} - 1$ distinct min-cuts. Throughout this section, we shall use the following notation. The (at most 5 element) terminal set is denoted as $Q = \{A,B,C,D,E\}$. The capacities of the minimum R-separating cuts, for $R \subseteq Q$, will be denoted as: $\alpha := b_{\{A\}}$, $\beta := b_{\{B\}}$, $\gamma := b_{\{C\}}$, $\delta := b_{\{D\}}$, $\varepsilon := b_{\{E\}}$, $\alpha\beta := b_{\{A,B\}}$, $\alpha\gamma := b_{\{A,C\}}$, $\alpha\delta := b_{\{A,D\}}$, $\alpha\varepsilon := b_{\{A,E\}}$, $\beta\gamma := b_{\{B,C\}}$, $\beta\delta := b_{\{B,D\}}$, $\beta\varepsilon := b_{\{B,E\}}$, $\gamma\delta := b_{\{C,D\}}$, $\gamma\varepsilon := b_{\{C,E\}}$, and $\delta\varepsilon := b_{\{D,E\}}$.

For vertices X, Y of a mimicking network $M(G)$, we shall denote the capacity of the edge XY by $w(XY)$, and the capacity of a cut with defining subset S by $w(S \mid \overline{S})$.

4.1 3-terminal networks

Let G be a network with three terminals A, B, C. Note that G has three distinct min-cuts whose values are α, β and γ.

Theorem 1 *A 3-terminal network G with terminal set $\{A,B,C\}$ has a mimicking network $M(G)$ which is the complete graph on vertices $\{A,B,C\}$ with the following capacities on its edges: $w(AB) = \frac{\alpha+\beta-\gamma}{2}$, $w(BC) = \frac{\beta+\gamma-\alpha}{2}$, and $w(AC) = \frac{\alpha+\gamma-\beta}{2}$.*

Proof. First we have to show that the edge capacities are nonnegative. To show that $w(AB) \geq 0$, it suffices to show that $\alpha + \beta \geq \gamma$. This true, because $\alpha + \beta \geq \alpha\beta = \gamma$ (a cut separating A from $\{B, C\}$, and B from $\{A, C\}$ definitely separates C from $\{A, B\}$).

To complete the proof, we have to show that the conditions of Lemma 1 are satisfied. We have, $w(A \mid BC) = w(AB) + w(AC) = \frac{\alpha+\beta-\gamma}{2} + \frac{\alpha+\gamma-\beta}{2} = \alpha$. Similarly, $w(B \mid AC) = \beta$ and $w(C \mid AB) = \gamma$. Hence, the conditions of Lemma 1 are indeed satisfied and consequently $M(G)$ is a mimicking network of G. ∎

Let $\mathcal{M}$ be a vector with values $[a, b, c]$. It can be easily verified that these values are realizable min-cut values of a 3-terminal network iff the following inequalities are satisfied: $a+b \geq c$, $b+c \geq a$, and $c + a \geq b$.

4.2 4-terminal networks

Let G be a network with four terminals A, B, C, D. Note that G has seven distinct min-cuts whose values are α, β, γ, δ, $\alpha\beta$, $\alpha\gamma$, $\alpha\delta$.

Theorem 2 *A 4-terminal network G with terminal set $\{A, B, C, D\}$ has a mimicking network $M(G)$ which is the complete graph on vertices $\{A, B, C, D, Z\}$, where Z is an additional vertex, with the following capacities on its edges:*
$w(AB) = \frac{\alpha+\beta-\alpha\beta}{2}$, $w(AC) = \frac{\alpha+\gamma-\alpha\gamma}{2}$, $w(AD) = \frac{\alpha+\delta-\alpha\delta}{2}$,
$w(BC) = \frac{\beta+\gamma-\beta\gamma}{2}$, $w(BD) = \frac{\beta+\delta-\beta\delta}{2}$, $w(CD) = \frac{\gamma+\delta-\gamma\delta}{2}$, *and*
$w(ZA) = w(ZB) = w(ZC) = w(ZD) = \Delta$, *where* $\Delta = \frac{(\alpha\beta+\alpha\gamma+\alpha\delta)-(\alpha+\beta+\gamma+\delta)}{2}$.

Proof. The capacities on the edges not incident on Z are all nonnegative as it was shown in the proof of Theorem 1. The nonnegativity of Δ follows by Lemma 3 with $M = \{\alpha\beta, \alpha\gamma, \alpha\delta\}$ and $N = \{\alpha, \beta, \gamma, \delta\}$. Hence, it remains to show that the conditions of Lemma 1 are satisfied.

It can be easily verified that $w(A \mid BCDZ) = \alpha$, $w(B \mid ACDZ) = \beta$, $w(C \mid ABDZ) = \gamma$, $w(D \mid ABCZ) = \delta$, and that $w(AB \mid CDZ) = w(ABZ \mid CD) = \alpha\beta$, $w(AC \mid BDZ) = w(ACZ \mid BD) = \alpha\gamma$, $w(AD \mid BCZ) = w(ADZ \mid BC) = \alpha\delta$. We must also ensure that any other cut that separates a subset of $\{A, B, C, D\}$ from the rest of terminals has value greater than or equal to these values. For example, we must have $w(AZ \mid BCD) \geq w(A \mid BCDZ)$, which is true since $w(AZ \mid BCD) = \alpha + 2\Delta \geq \alpha = w(A \mid BCDZ)$. The rest of the cuts can be checked in a similar way. Hence, the conditions of Lemma 1 are satisfied and consequently $M(G)$ is a mimicking network of G. ∎

Let $\mathcal{M}$ be a vector with values $[a, b, c, d, ab, ac, ad]$. It can be easily verified that these values are realizable min-cut values of a 4-terminal network iff the following inequalities are satisfied: $a + b \geq ab$, $a + c \geq ac$, $a + d \geq ad$, $b + c \geq ad$, $b + d \geq bd$, $c + d \geq ab$, $ab + ac + ad \geq a + b + c + d$.

4.3 5-terminal networks

Let G be a network with five terminals A, B, C, D, E. Note that G has 15 distinct min-cuts whose values are α, β, γ, δ, ε, $\alpha\beta$, $\alpha\gamma$, $\alpha\delta$, $\alpha\varepsilon$, $\beta\gamma$, $\beta\delta$, $\beta\varepsilon$, $\gamma\delta$, $\gamma\varepsilon$ and $\delta\varepsilon$.

Theorem 3 *A 5-terminal network G with terminal set $\{A, B, C, D, E\}$ has a mimicking network $M(G)$ which is the complete graph on 6 vertices $\{A, B, C, D, E, Z\}$, where Z is an additional vertex, with the following capacities on its edges:*
$w(AB) = \frac{\alpha+\beta-\alpha\beta}{2}$, $w(AC) = \frac{\alpha+\gamma-\alpha\gamma}{2}$, $w(AD) = \frac{\alpha+\delta-\alpha\delta}{2}$, $w(AE) = \frac{\alpha+\varepsilon-\alpha\varepsilon}{2}$,
$w(BC) = \frac{\beta+\gamma-\beta\gamma}{2}$, $w(BD) = \frac{\beta+\delta-\beta\delta}{2}$, $w(BE) = \frac{\beta+\varepsilon-\beta\varepsilon}{2}$,
$w(CD) = \frac{\gamma+\delta-\gamma\delta}{2}$, $w(CE) = \frac{\gamma+\varepsilon-\gamma\varepsilon}{2}$, $w(DE) = \frac{\delta+\varepsilon-\delta\varepsilon}{2}$,

$w(ZA) = \Delta_A = \frac{(\alpha\beta+\alpha\gamma+\alpha\delta+\alpha\varepsilon)-(2\alpha+\beta+\gamma+\delta+\varepsilon)}{2}$, $w(ZB) = \Delta_B = \frac{(\alpha\beta+\beta\gamma+\beta\delta+\beta\varepsilon)-(2\beta+\alpha+\gamma+\delta+\varepsilon)}{2}$,
$w(ZC) = \Delta_C = \frac{(\alpha\gamma+\beta\gamma+\gamma\delta+\gamma\varepsilon)-(2\gamma+\alpha+\beta+\delta+\varepsilon)}{2}$, $w(ZD) = \Delta_D = \frac{(\alpha\delta+\beta\delta+\gamma\delta+\delta\varepsilon)-(2\delta+\alpha+\beta+\gamma+\varepsilon)}{2}$, *and*
$w(ZE) = \Delta_E = \frac{(\alpha\varepsilon+\beta\varepsilon+\gamma\varepsilon+\delta\varepsilon)-(2\varepsilon+\alpha+\beta+\gamma+\delta)}{2}$.

Proof. As before, the capacities of the edges not incident on Z are all nonnegative as it was shown in the proof of Theorem 1. Hence, it remains to show that $\Delta_A \geq 0$, $\Delta_B \geq 0$, $\Delta_C \geq 0$, $\Delta_D \geq 0$, and $\Delta_E \geq 0$. The first inequality follows immediately by Lemma 3 with $M = \{\alpha\beta, \alpha\gamma, \alpha\delta, \alpha\epsilon\}$ and $N = \{\alpha, \alpha, \beta, \gamma, \delta, \epsilon\}$. The rest of the inequalities can be proved similarly.

Now, we have to satisfy the conditions of Lemma 1. Observe that $w(A \mid BCDEZ) = \alpha$ (as required) and $w(AZ \mid BCDE) = \alpha - \Delta_A + \Delta_B + \Delta_C + \Delta_D + \Delta_E$. Since it must hold that $w(AZ \mid BCDE) \geq w(A \mid BCDEZ)$, we have to show that

$$\Delta_B + \Delta_C + \Delta_D + \Delta_E \geq \Delta_A \quad (1)$$

Also, notice that $w(AB \mid CDEZ) = \alpha\beta$ (as required) and $w(ABZ \mid CDE) = \alpha\beta + (\Delta_C + \Delta_D + \Delta_E) - (\Delta_A + \Delta_B)$. We have to guarantee that $w(ABZ \mid CDE) \geq w(AB \mid CDEZ)$, i.e., we have to show that

$$\Delta_C + \Delta_D + \Delta_E \geq \Delta_A + \Delta_B \quad (2)$$

Note that (2) implies (1). By considering the rest of the cuts, we get a total of 10 inequalities symmetric to (2) (and which can be all proved in a similar way). Hence, to complete the proof of the theorem, it suffices to show that (2) holds. As before, this follows by Lemma 3 with $M = \{\gamma\delta, \gamma\epsilon, \delta\epsilon\}$ and $N = \{\alpha\beta, \gamma, \delta, \epsilon\}$. ∎

Let $\mathcal{M}$ be a vector with 15 values $[a, \ldots, e, ab, \ldots, ae, bc, \ldots, be, cd, ce, de]$. It can be easily verified that these values are realizable min-cut values of a 5-terminal network iff the following inequalities are satisfied:

$$\begin{array}{rclcrcl}
a+b & \geq & ab & \quad & \Delta'_C + \Delta'_D + \Delta'_E & \geq & \Delta'_A + \Delta'_B \\
a+c & \geq & ac & & \Delta'_B + \Delta'_D + \Delta'_E & \geq & \Delta'_A + \Delta'_C \\
a+d & \geq & ad & & \Delta'_B + \Delta'_C + \Delta'_E & \geq & \Delta'_A + \Delta'_D \\
a+e & \geq & ae & & & \cdot & \\
b+c & \geq & bc & & & \cdot & \\
b+d & \geq & bd & & & \cdot & \\
b+e & \geq & be & & & \cdot & \\
c+d & \geq & cd & & & \cdot & \\
c+e & \geq & ce & & & \cdot & \\
d+e & \geq & de & & \Delta'_A + \Delta'_B + \Delta'_C & \geq & \Delta'_D + \Delta'_E \\
\multicolumn{7}{l}{\Delta'_A \geq 0, \Delta'_B \geq 0, \Delta'_C \geq 0, \Delta'_D \geq 0, \Delta'_E \geq 0}
\end{array}$$

where $\Delta'_A = \frac{(ab+ac+ad+ae)-(2a+b+c+d+e)}{2}$, ... , and $\Delta'_E = \frac{(ae+be+ce+de)-(2e+a+b+c+d)}{2}$.

4.4 Optimality

We now show that the preceding constructions of mimicking networks are optimal. All we have to show is the next lemma.

Lemma 4 *There exists a k-terminal network for which the mimicking network must have at least one vertex in addition to the k terminals, for any $k > 3$.*

Proof. Assume that the network G to be mimicked is a star (i.e., a tree of depth one) where all vertices but the center Z (root of the tree) are the $k > 3$ terminals. Let Q denote the terminal set and let all k edges have capacity 1. Hence, every min-cut in G has a non-zero value.

Now assume contrary to the lemma that there is a mimicking network G' for G *without* additional vertices. Since there are no additional vertices in G', we must have that G' contains exactly k vertices, where each vertex is a terminal. Take any two vertices (terminals) in G', say A and B. Then, $w(AB \mid Q \setminus \{A,B\}) = w(A \mid Q \setminus \{A\}) + w(B \mid Q \setminus \{B\}) - 2w(AB)$, and since G' is a mimicking network, this gives that $\alpha\beta = \alpha + \beta - 2w(AB)$, or $w(AB) = \frac{\alpha+\beta-\alpha\beta}{2}$. Since $k > 3$, $\alpha\beta = w(AZ) + w(BZ) = 2$. Hence, $w(AB) = \frac{1+1-2}{2} = 0$.

But since A and B are any two terminals, this implies that every edge of the mimicking network has capacity zero, which, in turn, implies that the value of every cut in G' is also zero; this contradicts the fact that all min-cuts in G have non-zero value. ∎

5 Bounded Treewidth Networks

Let G be a network of bounded treewidth and (X,T) its augmented binary tree decomposition. For a subtree T' of T, we define the subgraph G' *spanned by* T', as follows. The vertices of G' are the vertices in the sets associated with the vertices of T', i.e., $V(G') = \cup_{i \in V(T')} X_i$. The edges of G' are those edges that the augmenting function maps to vertices in T', i.e., $E(G') = \{e \in E(G) : h(e) \in V(T')\}$. It is easy to check that vertex-disjoint subtrees span edge-disjoint subgraphs. (In fact it is only to ensure this property that we introduce the augmenting function.)

Theorem 4 *Let G be a n-vertex network of treewidth t. Let $Q \subseteq V(G)$ be the terminals of G and let $|Q| = k$. Then, a mimicking network for G at terminals Q has size at most $k2^{2^{3(t+1)}}$.*

Proof. We use induction on k. If $k \leq 3(t+1)$, then the theorem follows from the result in [10]. Assume the theorem holds for $k' < k$. We will show that it holds for networks with k terminals.

Let (X,T) be an augmented binary tree decomposition of G. Root T at some arbitrary vertex of degree less than 3. We define the following notation. For a vertex $v \in V(T)$, let T_v^- denote the subtree of T rooted at v, including the vertex v, and let T_v^+ denote the subtree $T \backslash T_v^-$. Define G_v^- and G_v^+ to be the subgraphs spanned by T_v^- and T_v^+ respectively. Let w be the parent of v. Define $Q_v^- = Q \cap V(G_v^-)$ and $Q_v^+ = Q \cap V(G_v^+)$.

Suppose we have a vertex, $v \in V(T)$, such that $|Q_v^+| < k$ and $|Q_v^-| \leq 3(t+1)$. Let w be the parent of v. The continuity condition of tree decompositions ensures that $X_v \cap X_w$ is a separator. *Split* the graph at $X_v \cap X_w$ into G_v^+ and G_v^-. By induction, we construct a mimicking network of size at most $(k-1)2^{2^{3(t+1)}}$ for G_v^+ at terminals Q_v^+. Using the algorithm of [10], we construct a mimicking network of size at most $2^{2^{3(t+1)}}$ for G_v^- at terminals Q_v^-. *Glueing* the two networks together yields a mimicking network for G at terminals Q of the required size.

To complete the proof, we only have to describe how to find such a vertex, v. Let w be a vertex in T, with children l and r, such that $|Q_w^-| > 3(t+1)$ (such a vertex must exist, since $k > 3(t+1)$). Let $x = |Q \cap (V(G_l^-) \backslash V(G_l^+))|$, that is, x is the number of terminals of G that appear in G_l^- but not in G_l^+. Similarly, define $y = |Q \cap (V(G_r^-) \backslash V(G_r^+))|$. Without loss of generality, assume $x \geq y$. Observe that $x + y + |X_w| \geq |Q_w^-| > 3(t+1)$. Since $|X_w| \leq t+1$, and $x \geq y$, we have $x > t+1$. This implies that $|Q_l^+| < k$, since $|Q_l^+| \leq k - x + |X_l| \leq k - x + t + 1$. Hence, if $|Q_l^-| \leq 3(t+1)$, then l is the vertex v that we are seeking. Otherwise, we just repeat the argument with l and its children. This defines an algorithm to find the required vertex v. ∎

The above construction holds for both directed and undirected bounded treewidth networks.

6 Outerplanar networks

Outerplanar graphs have treewidth 2. Theorem 4 promises a mimicking network of size $2^{2^7}k$ for an outerplanar network with k terminals. In this section, by arguing more carefully for outerplanar networks, we give a different construction for mimicking networks of much smaller size.

6.1 The general case

For simplicity, we start with the case of undirected outerplanar networks. The basic idea is as follows. Consider an embedding of the outerplanar network. Then, the endpoints of any edge not adjacent to the outer face are a separator. The proof shows how to *split* the network at a number of such separators to obtain $O(k)$ subgraphs with the property that each has either 3 or 4 terminals (including its separator vertices). The size upper bound follows by *glueing* together the mimicking networks for these subgraphs.

Theorem 5 *Let G be an n-vertex undirected outerplanar network with terminal set Q, where $|Q| = k$. Then, in $O(n)$ time we can construct: (i) a mimicking network of G at Q with at most $10k - 6$ vertices; (ii) an outerplanar mimicking network of G at Q with at most $31k - 27$ vertices.*

Proof. We start with the case where G is biconnected and treat later the non-biconnected case. Fix an outerplanar embedding of G and triangulate its interior faces by adding edges of zero capacity. Associate each edge of G with a single face of the embedding that is not the outer face. One way to do this is to number the faces, giving the outer face the highest number and associate each edge with the smaller numbered of the two faces adjacent to it. Let D be the dual graph of this embedding. Deleting the vertex in D corresponding to the outer face yields a binary tree (since each interior face is a triangle), which we call T. A subtree S, of T, defines a subgraph $G(S)$ of G in a natural way; $G(S)$ is the union of the edges associated with the faces that are the duals of the vertices of S.

Let (u, v) be an edge of T, and let T_u and T_v be the subtrees obtained by deleting (u, v). We say (u, v) is an *effective edge* if, $\forall x \in \{u, v\}$, it holds that $G(T_x)$ contains a terminal of G that $G(T \backslash T_x)$ does not contain. For a vertex v of T, define its *effective degree* to be the number of effective edges it is adjacent to. As an example, consider Figure 1(a) which shows an outerplanar network with 5 terminals (black vertices) and its associated binary tree T (the edges of T are shown with dashed lines). The number next to a vertex in T denotes its effective degree.

We form groups of maximal connected subtrees of T that do not contain any vertices of effective degree 3. Shrinking each group into a single vertex by contracting edges yields a binary tree we call T'. We will show that in T': (i) there are at most k leaves; and (ii) no two vertices of degree less than 3 are adjacent.

To see (i), notice that a leaf, l, of T', corresponds, in T, to a maximal subtree with effective degree less than 3. Call this subtree T_l. The parent of l must have effective degree 3, which implies that $G(T_l)$ contains a terminal of G not contained in $G(T \backslash T_l)$. Since the number of terminals of G is k, (i) follows. The condition (ii) follows directly from the maximality of the groups.

Conditions (i) and (ii) ensure that T' has at most $k - 2$ degree 3 vertices, at most $k - 2 - 1$ degree 2 vertices and at most k leaves.

For S, a subtree of T', let $F(S)$ denote the subtree of T corresponding to the vertices of S. Define $H(S) = G(F(S))$. Figure 1(b) shows the tree T' for the tree T of Figure 1(a). The dotted lines denote the subgraphs $H(u)$ and $H(z)$ of G which include terminals q_5 and q_1 respectively.

Since T' is obtained from T by shrinking edges, there is a correspondence between the edges of T' and a subset of the edges of T, i.e., those that were not contracted. We split G into the

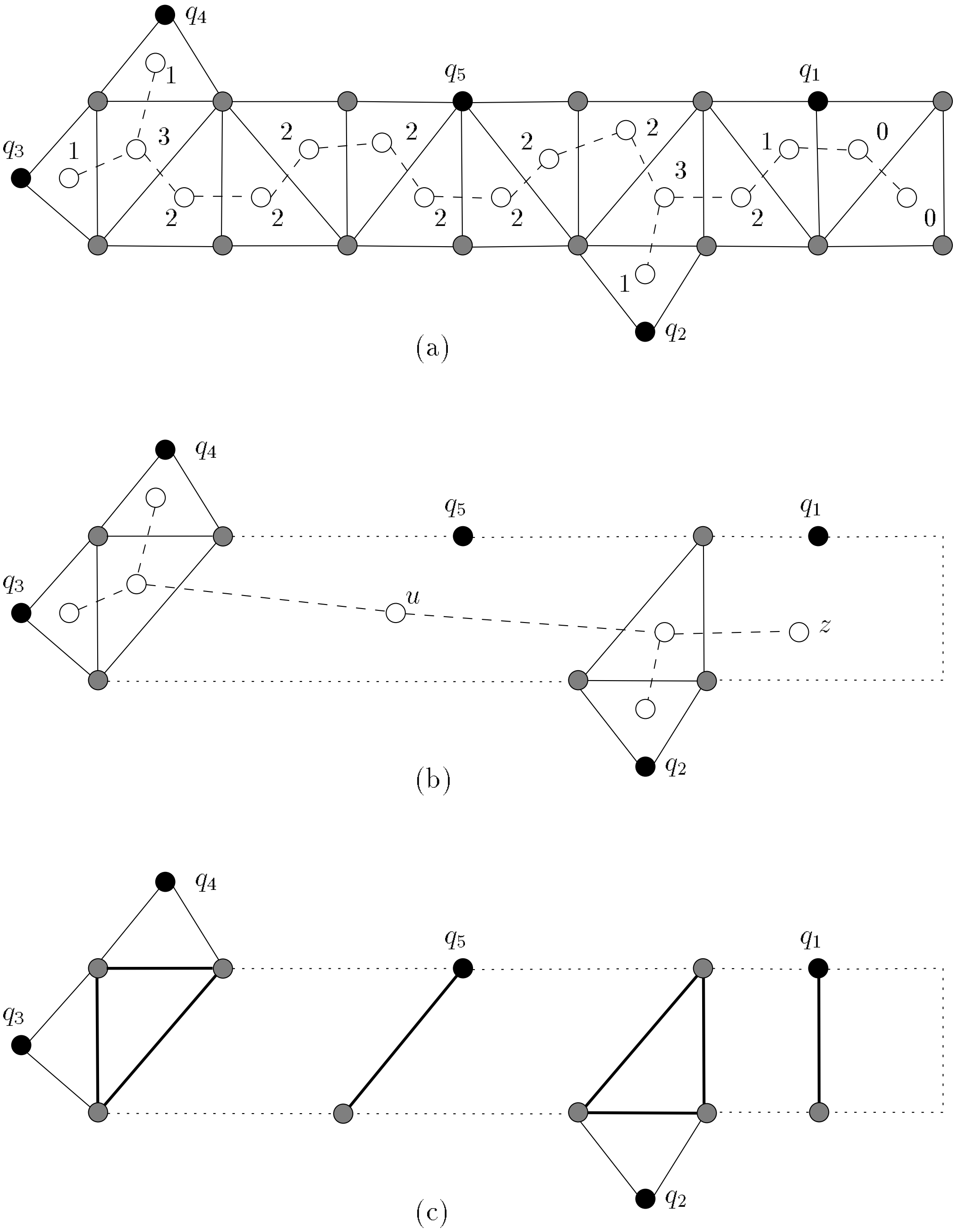

Figure 1: Computing a mimicking network for an outerplanar network.

following subgraphs: $H(v)$, $\forall v \in V(T')$. For a subgraph, $H(v)$, we define the terminals to be the endpoints of the duals of the edges adjacent to v. We consider two cases: if $F(v)$ is a vertex of T with effective degree three, then $H(v)$ is a face of G with three terminals. Otherwise, we will split $H(v)$ further to ensure that each resulting subgraph has at most four terminals. Let m be the number of terminals of G contained in $H(v)$ that are not contained in $H(T \backslash v)$. Then, we will split $H(v)$ into at most $m+1$ subgraphs.

If $m = 0$, then there is nothing to be done. Hence assume m is at least one. Construct a new subgraph as follows. $F(v)$ is connected to the rest of T by one or two effective edges, by construction. Start at the endpoint of one of these edges, e. The effective edges of $F(v)$ form a path, since each vertex of $F(v)$ has at most two such edges adjacent to it. Walk along this path until you encounter an edge f such that one of the endpoints of its dual is a terminal of G and is different from the two endpoints of the dual of e. Let T_1 denote the subtree in between e and f (not including either edge). Define $G(T_1)$ to be a new subgraph whose terminals are the endpoints of the duals of e and f, which ensures that the subgraph has at most 4 terminals. Start the process again with the next vertex, i.e., the other endpoint of f. Notice that we create a new subgraph only when we encounter a new terminal, or when we exhaust the subtree $F(v)$. Since we can encounter at most m new terminals, by hypothesis, we create at most $m+1$ subgraphs.

Let us illustrate this process on the tree T' of Figure 1(b). There are two vertices with effective degree 3 resulting into a face of G with three terminals; there are three leaves (whose dual faces contain the terminals q_2, q_3 and q_4), all of which have $m = 0$; and there are two vertices u and z such that both $H(u)$ and $H(z)$ contain a terminal, and consequently have to be split once. Figure 1(c) shows the resulting subnetworks after splitting. The bold edges show where the splits occurred and their endpoints are the terminals of the subnetworks.

When we perform this process on all degree 2 vertices and leaves of T', we increase the number of subgraphs by at most k. Hence the total number of subgraphs that we split G into is at most : $k-2$ subgraphs (that are faces) with 3 terminals, for the effective degree 3 vertices, and $k-2-1+k+k = 3(k-1)$ subgraphs with at most 4 terminals each for the other vertices. (In our example of Figure 1 we get a total of 9 subnetworks: 2 from the two vertices with effective degree 3, 3 from the leaves, and 4 from the splitting of $H(u)$ and $H(z)$; cf. Figure 1(c).) Construct mimicking networks for each subgraph with 3 terminals using Theorem 1 and for each subgraph with 4 terminals using either Theorem 2, or Lemma 5 (Section 6.2) which states that such a special 4-terminal subgraph has an outerplanar mimicking network of at most 12 vertices. Glueing the different mimicking networks together yields the desired mimicking network.

To bound the number of vertices in the resulting network, think of the glueing process sequentially. That is, we start with one mimicking network at glue to it another that shares separator terminals with it. We continue in this manner. Then, since each separator has two vertices, every time we glue a new mimicking network with x vertices, we increase the number of vertices by $x-2$. The first network has at most 5 vertices, after which we add one vertex for each 3-terminal mimicking network and (a) either three vertices for every 4-terminal mimicking network (Theorem 2), or (b) 10 vertices for every 4-terminal mimicking network (Lemma 5, Section 6.2). This yields a total of $5+(k-2)+3(3k-3) = 10k-6$ for the former case, and a total of $5+(k-2)+10(3k-3) = 31k-27$ for the latter case. It is easy to verify that the whole process takes $O(n)$ time.

We now discuss the case where G is not biconnected. First identify the articulation vertices of G (the vertices shared by two biconnected components). Let v be an articulation vertex. If v is a terminal, then do nothing. If v is not a terminal, then replace v by two vertices v_1 and v_2 connected by an edge of infinite capacity, and appropriately divide the edges incident on v between v_1 and v_2 such that outerplanarity is preserved. Now, proceed as before, i.e., start by finding the dual graph of G excluding the outer face. Since there may be still articulation vertices in G (which are terminals), T is a forest. In this case, work with every tree of T and find a mimicking network for the biconnected component it represents. Finally, glue together these mimicking networks at the articulation vertices which are terminals to get a mimicking network for G. Clearly, neither the size of the resulting mimicking network is affected, nor the time. ∎

In the case of directed outerplanar networks, we proceed in a similar fashion except that now we have to use the directed version of Lemma 5 (Section 6.2), and a new 4-vertex mimicking network for a 3-terminal directed network. The need for the latter is shown by the counterexample below.

As before, let A, B and C be the terminals of a 3-terminal directed network. Observe that there are six distinct min-cuts whose values are $\alpha, \beta, \gamma, \alpha\beta, \alpha\gamma$ and $\beta\gamma$.

Counterexample: Assume that a 3-terminal directed network has as a mimicking network the complete (directed) graph on vertices $\{A, B, C\}$. Then, we must have that $\alpha + \beta + \gamma = w(AB) + w(AC) + w(BA) + w(BC) + w(CA) + w(CB) = \alpha\beta + \alpha\gamma + \beta\gamma$. Let the input 3-terminal directed network be a 4-vertex network with vertex set $\{A, B, C, Z\}$ and edge set $\{AZ, BZ, CZ, ZA, ZB, ZC\}$ with capacities $w(AZ) = w(BZ) = w(CZ) = 2$ and $w(ZA) = w(ZB) = w(ZC) = 3$. Then, $\alpha + \beta + \gamma = 6 \neq 9 = \alpha\beta + \alpha\gamma + \beta\gamma$.

Hence, a mimicking network for a 3-terminal directed network must have at least 4 vertices. Such a 4-vertex network is described next (it seems that this construction was known before [9], but never published). It has vertex set $\{A, B, C, Z\}$, where Z is an additional vertex, and edge set $\{AZ, BZ, CZ, ZA, ZB, ZC\}$ with capacities: $w(AZ) = \alpha$, $w(BZ) = \beta$, $w(CZ) = \gamma$, $w(ZA) = \beta\gamma$, $w(ZB) = \alpha\gamma$, and $w(ZC) = \alpha\beta$. It can be easily verified (using Lemma 1) that this is indeed the required mimicking network.

Now, to find a mimicking network for a directed outerplanar network we follow the method of Theorem 5, with the exception that when we need to glue together mimicking networks we use a 28-vertex mimicking network (directed version of Lemma 5) for a 4-terminal network, and a 4-vertex mimicking network for a 3-terminal one. This yields a mimicking network which is outerplanar and has a total of $5 + 2(k-2) + 26(3k-3) = 80k - 77$ vertices. We have established the following.

Theorem 6 *Let G be an n-vertex directed outerplanar network with terminal set Q, where $|Q| = k$. Then, in $O(n)$ time we can construct an outerplanar mimicking network of G at Q with at most $80k - 77$ vertices.*

6.2 A special case

To complete the proofs of Theorems 5 and 6, recall that we have to find mimicking networks of 4-terminal outerplanar networks that have a special structure. That is, two of the terminals, say A, D, are in the same interior face, they are connected by an edge, and one of them is of degree 2. Similar conditions hold for the other two terminals, B, C, too. Let G be such a network and let $Q = \{A, B, C, D\}$. Based on the approach of [2] and on a new property that we prove below, we can show the following.

Lemma 5 *Let G be the above defined special 4-terminal outerplanar network with n vertices. Then, in $O(n)$ time we can construct: (i) an outerplanar mimicking network for G of at most 12 vertices, if G is undirected; (ii) an outerplanar mimicking network for G of at most 28 vertices, if G is directed.*

Proof. For convenience, assume that the vertices of G are numbered $1, 2, ..., n$ in clockwise order along the boundary of the outer face f_o of G. (W.l.o.g., assume that $A \leq B \leq C \leq D$ in this numbering.) Then, with $[i, j]$ we shall denote the interval of vertices in clockwise order along f_o from vertex i to vertex j, i.e., $[i, j]$ denotes the set $\{i, i+1, \ldots, j\}$ of vertices, if $i \leq j$, and it denotes $\{i, i+1, \ldots, n, 1, \ldots, j\}$, if $i > j$. Let a *chain* be the set of vertices determined by some interval $[i, j]$. We shall say that an interval $[i_2, j_2]$ is *between* two intervals $[i_1, j_1]$ and $[i_3, j_3]$ for which $i_1 \leq j_3$, if $j_1 \leq i_2$ and $j_2 \leq i_3$. The case where $i_1 > j_3$ is defined similarly.

As observed in [2], any coloring of the vertices of G with green and red colors defines a cut, namely, the cut separating the green vertices from the red ones. For a subset $R \subseteq Q$ of terminals, let $(S_R, \overline{S_R})$ be a minimum R-separating cut. Color the vertices of S_R green and those of $\overline{S_R}$ red. A *green unit* is defined to be a maximal chain of green vertices, and a *red unit* is defined analogously. Define the *support* of a green unit to be a green terminal such that some (and therefore every) vertex in the unit has an undirected path, consisting only of green vertices, to this terminal. Similarly, define the support of a red unit. We say a green unit is *unsupported* if no vertex in the unit has an undirected path, consisting only of green vertices, to a green terminal. Define an unsupported red unit analogously. A collection of unsupported units is *connected* if there is an undirected path, not including a vertex from any supported unit, between any two units of the collection.

In [2] it is proved that the cut obtained by changing the color of any maximal monochromatic connected collection of unsupported units is also a minimum R-separating cut. Hence, these unsupported units can be "absorbed" by a neighboring monochromatic supported unit. To apply the linear-time method of [2], we need the following property.

Proposition 1 *If G does not contain any unsupported units, then the defining set S_R of any minimum R-separating cut induces at most 2 units in G.*

Proof. It suffices to prove that each terminal in S_R supports one unit in each of $[A, B]$ and $[C, D]$.

Assume on the contrary that there are two units U_1, U_2 in $[A, B]$ supported by a terminal in S_R. W.l.o.g., assume that this is terminal A and also assume that $A \in U_1$. This implies that there is a path p (consisting only of vertices in S_R) from some vertex $u_1 \in U_1$ to some vertex $u_2 \in U_2$. Since there are no unsupported units in G, there must be at least one unit U' between U_1 and U_2 that is supported by a terminal $X \in \overline{S_R}$, i.e., there is a path p' (consisting only of vertices in $\overline{S_R}$) from X to some vertex $u' \in U'$. However, the subgraph of G induced by the boundary of the outer face and the paths p and p' is homeomorphic to the complete graph on the 4 vertices u_1, u', u_2 and X, which contradicts the outerplanarity of G. Hence, there is at most one unit supported by A in $[A, B]$. Similarly, we can prove that there is at most one unit in $[C, D]$. ∎

Now, we construct the mimicking network of G using the above Proposition and the approach in [2]. Let m be the (distinct) minimum R-separating cuts in G, each one defining at most 2 units from the proposition above, and whose intersection defines a partition of the vertex set of V into $2m$ chains. Contract (repeatedly) the edges between two vertices in the same chain and replace multiple edges by a single edge of capacity equal to the sum of the capacities of the edges it replaces. The resulting network is clearly outerplanar and has at most $2m$ vertices. The construction can be done in time $O(mn)$, where n is the number of vertices in G. Note that if G is directed, then $m = 2^4 - 2 = 14$. If G is undirected, then $m = 2^{4-1} - 1 = 7$ which can be further reduced to 6, as we show next.

Define a cut $(S_R, \overline{S_R})$ to be *connected* in G, if both S_R and $\overline{S_R}$ are connected. Otherwise, $(S_R, \overline{S_R})$ is called *disconnected*. Call a minimum R-separating cut a *singleton cut*, if R is a singleton. Consider the 3 minimum R-separating cuts in G for which R contains 2 terminals. Observe that $(S_{AD}, \overline{S_{AD}})$ is always connected and $(S_{AC}, \overline{S_{AC}})$ is always disconnected. To prove that $m = 6$, it suffices to show that $(S_{AC}, \overline{S_{AC}})$ can be expressed in terms of two singleton cuts.

Assume w.l.o.g. that S_{AC} is disconnected. Let S^1_{AC} and S^2_{AC} be the two parts of S_{AC}, where the first contains A and the second contains C. Note that $(S^1_{AC}, \overline{S^1_{AC}})$ is a cut separating A from $\{B, C, D\}$. Hence, $w(S^1_{AC} | \overline{S^1_{AC}}) \geq \alpha = w(S_A | \overline{S_A})$. Similarly, $(S^2_{AC}, \overline{S^2_{AC}})$ is a cut separating

C from $\{A, B, D\}$. Hence, $w(S^2_{AC}|\overline{S^2_{AC}}) \geq \gamma = w(S_C|\overline{S_C})$. Therefore, $\alpha\gamma = w(S_{AC}|\overline{S_{AC}}) = w(S^1_{AC}|\overline{S^1_{AC}}) + w(S^2_{AC}|\overline{S^2_{AC}}) \geq \alpha + \gamma$.

On the other hand, we know that $\alpha\gamma \leq \alpha + \gamma$, since $(S_{AC}, \overline{S_{AC}})$ is a min-cut. Consequently, $\alpha\gamma = \alpha + \gamma$, which implies that $(S_{AC}, \overline{S_{AC}})$ consists of the two singleton min-cuts $(S_A, \overline{S_A})$ and $(S_C, \overline{S_C})$. This completes the proof of the lemma. ∎

7 Extensions of Our Results

Another important area of research is to mimic the cut properties of a hypergraph H by a graph G. This is called modeling of a hypergraph by a graph with the same min-cut properties [11, 12] and is of particular importance on partitioning circuits in layout problems. In this case, circuits are modeled as hypergraphs and the task is to partition them according to a minimum cut of the hypergraph. However, the existing partitioning algorithms are usually designed for (ordinary) graphs and become very inefficient when applied to hypergraphs [15], mainly because all interesting variants of the partitioning problem involving hypergraphs are NP-hard. Moreover, no approximation algorithm exists for any of the hypergraph partitioning problems. For this reason, an elegant and general method is to model hypergraphs by graphs with the same min-cut properties and then apply the graph partitioning algorithms to the resulting model [12, Sec. 6.1.5]. Note that in this case, it suffices to examine the existence of graphs that model a single hyperedge with capacity 1. For if this is true, then we can construct a model for an arbitrary hypergraph with positive hyperedge capacities by multiplying the edge capacities by appropriate factors and putting together the models for all the hyperedges.

A graph $G = (V \cup D, E)$, where $|V| = k$ and $|D| = d \geq 0$, is called a *min-cut model* for a hyperedge with capacity 1 and vertex set V', which is in 1-1 correspondence with V, if for every non-empty $U \subseteq V$ the capacity of any min-cut with defining subset $S \supseteq U$ is equal to 1. If $U = \emptyset$, then the capacity of the min-cut should be 0.

In [11] it is shown that there is no min-cut model for $k \geq 4$ if negative capacities on the edges of G are not allowed. But one can hope to find an approximate min-cut model with positive edge capacities that is as balanced as possible, i.e., the quotient between the capacity of the largest and the smallest min-cut that partitions V' is as small as possible. More formally:

A graph $G = (V \cup D, E)$, where $|V| = k$ and $|D| = d \geq 0$, with positive edge capacities is called a *b-approximate min-cut model* for a hyperedge with capacity 1 and vertex set V', which is in 1-1 correspondence with V, if the quotient between the capacity of the largest and the smallest min-cut separating a proper and non-empty subset U of V from $V \setminus U$ is b. The quotient b is called the *balance* of the model.

Lengauer conjectures in [12, Sec. 6.1.5] (see also [11, Conjecture 6]) that cliques without additional vertices (i.e., $d = 0$) are the best balanced approximate min-cut models. In the next theorem we show that Lengauer's Conjecture is true.

Theorem 7 (Lengauer's Conjecture) *For a hyperedge with capacity 1 and k vertices, there is no b-approximate min-cut model having a balance b smaller than that achieved by the complete graph on k vertices and all edge capacities equal to* $1/(k-1)$.

Proof. Take a best approximate min-cut model, i.e., a graph with $k + d$ vertices and positive edge capacities such that the quotient between the capacity $w(L)$ of the largest min-cut and the capacity $w(S)$ of the smallest min-cut is minimum.

Let M be the set of $\frac{\binom{k}{k/2}}{2}$ min-cuts separating $k/2$ of the terminals from the rest. Let N be the multiset of the k min-cuts separating a single terminal from the rest, where each one of the min-cuts is taken with multiplicity $\frac{k\binom{k}{k/2}}{8(k-1)}$. Now, invoke Lemma 3 with these two sets. It is not hard to verify that the conditions of Lemma 3 are fulfilled.

This implies that the capacity of the $\frac{\binom{k}{k/2}}{2}$ min-cuts in M is greater than or equal to the capacity of the $\frac{k^2\binom{k}{k/2}}{8(k-1)}$ min-cuts in N. Hence, the capacity of the largest min-cut in M, say $w(m)$, must be at least

$$\frac{k^2\binom{k}{k/2}/(8(k-1))}{\binom{k}{k/2}/2} = \frac{k^2}{4(k-1)}$$

times larger than the capacity of the smallest min-cut in N, say $w(n)$. Consequently, $\frac{w(L)}{w(S)} \geq \frac{w(m)}{w(n)} \geq \frac{k^2}{4(k-1)}$. But it can be easily verified that this is exactly the quotient achieved by a clique on k vertices and all edge capacities equal to $1/(k-1)$, since the smallest min-cut in such a clique cuts $k-1$ edges and hence has capacity 1, while the largest min-cut cuts $\frac{k}{2} \times \frac{k}{2}$ edges and hence has capacity $\frac{k^2}{4(k-1)}$. ∎

Acknowledgements. We are grateful to the anonymous referees for carefully reading the manuscript and making several valuable comments and suggestions.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows* Prentice-Hall, 1993.

[2] S. Arikati, S. Chaudhuri, and C. Zaroliagis, All-Pairs Min-Cut in Sparse Networks, *Journal of Algorithms*, to appear; preliminary version in *Found. of Software Technology and Theor. Comp. Science* – FSTTCS'95, Lecture Notes in Computer Science Vol. 1026, Springer-Verlag, 1995, pp.363-376.

[3] S. Arnborg, Efficient Algorithms for Combinatorial Problems on Graphs with Bounded Decomposability - A Survey, *BIT*, **25** (1985), 2-23.

[4] H. Bodlaender, A Linear Time Algorithm for Finding Tree-decompositions of Small Treewidth, in *Proc. 25th ACM Symposium on Theory of Computing (STOC'93)*, ACM, 1993, pp.226-234.

[5] H. Bodlaender, A Tourist Guide through Treewidth, *Acta Cybernetica*, **11:1-2** (1993), 1-21.

[6] C.K. Cheng and T.C. Hu, Maximum Concurrent Flows and Minimum Cuts, *Algorithmica*, **8** (1992), 233-249.

[7] D. Gusfield, Very Simple Methods for All Pairs Network Flow Analysis, *SIAM Journal on Computing*, **19** (1990), 143-155.

[8] R.E. Gomory and T.C. Hu, Multi-terminal network flows, *Journal of SIAM*, **9** (1961), 551-570.

[9] T. Hagerup, private communication, 1997.

[10] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde, Characterizations of k-Terminal Flow Networks and Computing Network Flows in Partial k-Trees, in *Proc. 6th ACM-SIAM Symposium on Discrete Algorithms (SODA'95)* ACM-SIAM, 1995, pp. 641-649.

[11] E. Ihler, F. Wagner, and D. Wagner, Modeling hypergraphs by graphs with the same mincut properties, *Inform. Proc. Lett.*, **45** (1993), 171-175.

[12] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Teubner, Stuttgart and Wiley, Chichester, UK, 1990.

[13] N. Robertson and P. Seymour, Graph Minors II: Algorithmic Aspects of Treewidth, *Journal of Algorithms*, **7** (1986), 309-322.

[14] M.T. Shing and T.C. Hu, A Decomposition Algorithm for Multi-terminal Network Flows, *Discrete Applied Mathematics*, **13** (1986), 165-181.

[15] A. Vanneli and S.W. Hadley, A Gomory-Hu cut tree representation of a netlist partitioning problem, *IEEE Trans. Circuits and Systems*, **37** (1990), 1133-1139.