P.L. Erdős, A. Frank, L.A. Székely, *Minimum multiway cuts in trees,* Discrete Applied Mathematics, 87, No. 1-3, (1998), pp. 67-76.

MINIMUM MULTIWAY CUTS IN TREES

Péter L. Erdős*
András Frank**
László Székely***

ABSTRACT We compare three lower bounds for the minimum cardinality of a multiway cut in a graph separating a given set S of terminals. The main result is a relatively short algorithmic proof for a simplified version of a min-max theorem of the first and the third authors asserting that the best of the three lower bounds is actually attainable if every circuit of the graph contains a terminal node.

0. INTRODUCTION

Let $G = (V, E)$ be a connected graph with no loops and S a specified subset of nodes. A family $\mathcal{P} := \{V_1, V_2, \ldots, V_t\}$ of pairwise disjoint non-empty subsets of V whose union is V is called a **partition** of V. $\mathcal{P}$ is said to **separate** S or to be S**-separating** if each member of $\mathcal{P}$ contains exactly one element of S. The **value** $e_G(\mathcal{P})$ of $\mathcal{P}$ is the number of edges connecting distinct parts. Clearly, $e_G(\mathcal{P}) = \sum_{X \in \mathcal{P}} d(X)/2$ where $d(X)$ denotes the number of edges leaving X. The set of edges connecting distinct members ofan S-separating partititon $\mathcal{P}$ is called a **multiway cut** (separating S). We are interested in the minimum cardinality π_S of a multiway cut, that is, in an S-separating partition of minimum value.

* Mathematical Institute of the Hungarian Academy of Sciences, Reáltanoda u. 13-15, Budapest, Hungary, H-1053, e-mail: `elp@math-inst.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T016358.

** Department of Operations Research, Eötvös University, Múzeum krt. 6-8, Budapest, Hungary, H-1088 and Ericsson Traffic Laboratory, Laborc u.1, Budapest, Hungary H-1037. e-mail: `frank@cs.elte.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T17580.

*** Department of Computer Science, Eötvös University, Múzeum krt. 6-8, Budapest, Hungary, H-1088. e-mail: `szekely@cs.elte.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T016358 and by NSF DMS 9701211.

Minimum multiway cuts have been subject of study before, see, e.g. [1], [2], [3]. The minimum multiway cut problem was shown to be NP-complete even for some restricted versions [4]. Paper [5] introduced a new lower bound for the minimum cardinality of a multiway cut and proved a min-max theorem in the special case when the nodes of S cover all circuits. For this special class, even the weighted multiway cut problem has been solved in [6].

In the first section we introduce a new lower bound $\vec{\nu}_S$ for the minimum multiway cut. This value is the same as the lower bound used in [5] but its definition is more transparent. We compare $\vec{\nu}_S$ with two known lower bounds and prove that, among these three, $\vec{\nu}_S$ is always the best (that is, the largest). Section 2 includes the main contribution of the paper: it is a relatively simple algorithmic proof for a simlified version of a min-max theorem of [5]. This result is about undirected graphs and the basic idea behind the present simplification is that we introduce orientations of the underlying undirected graph.

We need the following notions and notation. A **leaf** of a tree is a node of degree one. A **star** is a tree whose all but possibly one nodes are leaves. We call a directed tree $\vec{T}$ an **arborescence** if every node is reachable by a directed path from a special node, called the root of $\vec{T}$.

Given a hypergraph $\mathcal{L}$, the **degree** of a node u is the number of members of $\mathcal{L}$ containing u. For a subset Z of nodes of a graph $G = (V, E)$, the set of edges connecting Z and $V - Z$ is called a **cut.** It is denoted by $[Z, V - Z]$ and its cardinality by $d(Z) = d_G(Z)$. For a digraph $\vec{G}$ let $\varrho(Z) = \varrho_{\vec{G}}(Z)$ denote the number of edges entering Z. For two disjoint subsets A, B of V, let $\lambda(A, B; \vec{G})$ denote the maximum number of edge-disjont (directed) paths with starting node in A and end node in B. By Menger's theorem $\lambda(A, B; \vec{G}) = \min(\varrho(X) : B \subseteq X \subseteq V - A)$. For $s \in S$ let $\lambda(S - s, s; G)$ denote the maximum number of edge-disjoint paths from $S - s$ to s. If $\vec{G}$ is a directed graph we use the notation $\lambda(S - s, s; \vec{G})$ for the maximum number of edge-disjoint directed paths from $S - s$ to s. Note that via the Max-flow Min-cut algorithm both $\lambda(S - s, s; G)$ and $\lambda(S - s, s; \vec{G})$ are computable in polynomial time.

1. LOWER BOUNDS

First, we try to find some lower bounds for π_S. Let $\tau_S^* := \sum_{s \in S} \lambda(S - s, s; G)/2$. The quantity τ_S^* was introduced by Lovász [7]. He proved that τ_S^* is equal to the maximum value of a fractional packing of S-paths and also to the minimum value of a fractional edge-covering of S-paths, where an S-path is a path connecting two distinct elements of S. For τ_S^* one has $\tau_S^* = \sum_{s \in S} \lambda(S - s, s; G)/2 \leq \sum_{s \in S} d(V_s)/2 = e_G(\mathcal{P})$ for any S-separating partition $\mathcal{P} = \{V_s : s \in S\}$, from which $\tau_S^* \leq \pi_S$ follows. Therefore τ_S^* is a polynomially computable lower bound for π_S. It is not a very good one though as is shown by a star with k leaves where S consists of the k leaves. For such a star $\tau_S^* = k/2$ and $\pi_S = k-1$.

In order to obtain better bounds we introduce two other parameters. By the **value** $\mathrm{val}(T)$ of a sub-tree T of G we mean the number of its leaves belonging to S minus one. In particular, the value of a path connecting two elements of S is 1. Let ν_S^{tree} denote the maximum sum of values of edge-disjoint trees of G. Let $\vec{\nu}_S := \max(\sum_{s\in S} \lambda(S-s, s; \vec{G}))$ where the maximum is taken over all orientations $\vec{G}$ of G.

THEOREM 1.1 $\tau_S^* \leq \nu_S^{tree} \leq \vec{\nu}_S \leq \pi_S$.

Proof. To see the last inequality, suppose that $\vec{G}$ is an orientation of G for which $\vec{\nu}_S = \sum_{s\in S} \lambda(S-s, s; \vec{G})$ and that $\mathcal{P} := \{V_s : s \in S\}$ an S-separating partition for which $e_G(\mathcal{P}) = \pi_S$. Then

$$\vec{\nu}_S = \sum_{s\in S} \lambda(S-s, s; \vec{G}) \leq \sum_{s\in S} \varrho(V_s) = \sum_{s\in S} d(V_s)/2 = e_G(\mathcal{P}) = \pi_S,$$

as required.

The middle inequality is also straightforward. Indeed, let $T_1, T_2, \ldots, T_k$ be the members of an optimal packing of trees. Orient the edges of each T_i as follows. Choose arbitrarily a leaf of T_i in S and orient each edge of T_i so as to obtain an arborescence with this root. The edges not in any T_i may be oriented arbitrarily. In such an orientation $\vec{G}$ of G the value $\lambda(S-s, s; \vec{G})$ is at least as large as the number of trees containing s whose chosen root is different from s. Therefore the sum $\sum_{s\in S} \lambda(S-s, s; \vec{G})$ is at least the sum of the values of the trees. We obtain, that $\vec{\nu}_S \geq \sum_{s\in S} \lambda(S-s, s; \vec{G}) \geq \nu_S^{tree}$.

Finally, we prove the first inequality

$$\tau_S^*(G) \leq \nu_S^{tree}(G). \tag{1.1}$$

By induction we assume that

($*$) the inequality (1.1) holds for any graph $G' = (V', E')$ for which $|V'| + |E'| < |V| + |E|$.

We may assume that the deletion of any edge e decreases τ_S^*. Indeed, if the deletion of e leaves τ_S^* unchanged, then by ($*$) we have $\tau_S^*(G) = \tau_S^*(G-e) \leq \nu_S^{tree}(G-e) \leq \nu_S^{tree}(G)$, as required. We also may assume that there is no edge e connecting two elements of S. Indeed, leaving out such an edge decreases both τ_S^* and ν_S^{tree} by one and hence $(*)$ implies again (1.1).

Case 1. *There is a set Z of nodes for which $|Z| \geq 2, Z \cap S = \{s\}$ for some $s \in S$ and $\lambda(S-s, s; G) = d_G(Z)$.*

Contract Z into one node denoted by s_Z. In the contracted graph G' let $S' := S - s + s_Z$. Using $(*)$ and the fact that contraction does not decrease any value $\lambda(S-x, x; G)(x \in S)$, we have $\nu_S^{tree}(G') \geq \tau_{S'}^*(G') \geq \tau_{S'}^*(G)$. Therefore there is a family $\mathcal{T}'$ of edge-disjoint trees in G' so that $\sum(\text{val}(T) : T \in \mathcal{T}') \geq \tau_S^*(G)$.

We assume that $|\mathcal{T}'|$ is as large as possible. In this case we claim that each terminal node $x \in S'$ belonging to a tree $T \in \mathcal{T}'$ is a leaf of T. For otherwise we could split T at x into $d_T(x)$ subtrees. Then the total value of the new family of trees is unchanged, contradicting the maximality of $|\mathcal{T}'|$, and the claim follows.

Since $\lambda(S-s, s; G) = d_G(Z)$, there is a family of $d_G(Z)$ edge-disjoint paths in G connecting s and $S-s$. For an edge e in the cut $[Z, V-Z]$ let P_e denote the path in this family containing e and let P_e' be the subpath of P_e whose first node is s and last edge is e. If a tree $T' \in \mathcal{T}'$ uses an edge e' of G' corresponding to an edge $e \in [Z, V-Z]$ of G, then $T := T' - e' + P'(e)$ is a tree of G for which val(T) =val(T'). In $\mathcal{T}'$ replace each such T' by T.

Every tree in $\mathcal{T}'$ not containing s_Z corresponds to a tree T of G (disjoint from Z) whose value is the same. Therefore we have obtained a family $\mathcal{T}$ of edge-disjoint trees of G for which $\nu_S^{tree}(G) \geq \sum(\text{val}(T) : T \in \mathcal{T}) = \sum(\text{val}(T') : T' \in \mathcal{T}') \geq \tau_S^*(G') \geq \tau_S^*(G)$, as required.

Case 2.

$$\lambda(S-s, s; G) = d_G(s) < d_G(Z) \tag{1.2}$$

holds whenever $s \in S$, $Z \cap S = \{s\}$ and $|Z| \geq 2$.

By Menger's theorem, the deletion of an edge e decreases τ_S^* if and only if e belongs to a (minimum) cut $[Z, V-Z]$ for which $Z \cap S = \{s\}$ and $d_G(s) = \lambda(S-s, s; G)$ for some $s \in S$. Therefore every edge e of G has exactly one end-node in S, that is, G is bipartite.

For each $s \in S, v \in V-S$ let $c(sv)$ denote the number of parallel edges between s and v. For $v \in V-s$ let $\alpha(v) := \max(c(sv) : s \in S)$. We claim that

$$\alpha(v) < d_G(v)/2, \tag{1.3}$$

for otherwise $d_G(\{s, v\}) \leq d_G(s)$, contradicting (1.2). By (1.3) the set of edges incident to v can be partitioned into $\alpha(v)$ stars so that each contains at least two edges. The value of one such star is one less than the number of its edges and hence the total value of the $\alpha(v)$ trees is $d_G(v) - \alpha(v)$. Applying this way of partitioning to each $v \in V - S$, we obtain a family of trees whose total value is $\sum([d_G(v) - \alpha(v)] : v \in V - S) > \sum(d_G(v)/2 : v \in V - S) = |E|/2 = \sum(d_G(s)/2 : s \in S) = \sum(\lambda(S - s, s; G)/2 : s \in S) = \tau^*_S(G)$, from which (1.1) follows. •

In Theorem 1.1 strict inequality may occur at each place. That was shown already for the first inequality. In the graph in Figure 1 $\tau^*_S = 3 = \nu^{tree}_S$ and $\vec{\nu}_S = 4 = \pi_S$, that is the second inequality is strict.

Fig. 1

In the first graph in Figure 2 $\tau^*_S = 6, \nu^{tree}_S = 7 = \vec{\nu}_S, \pi_S = 8$ (A tree-packing of total value 7 is shown in the second of Figure 2. The fact that $\vec{\nu}_S < 8$ can be shown by case checking.)

Fig 2.

2. MIN-MAX THEOREM AND ALGORITHM

The example of Fig. 2 leaves little room to find classes of graphs for which $\vec{\nu}_S = \pi_S$. In what follows we prove that those graphs for which $G - S$ induces a tree form such a class. (The apparently more general case when $G - S$ induces a forest is easily seen to be equivalent to the tree case.) The theorem below is equivalent to the min-max theorem of [5], but formulated in simpler terms relying on the notion of orientations. This idea gave rise to a proof significantly simpler than the original one. (Actually, the result below extends to the case when $G - S$ may induce only 2-element circuits. This is equivalent to the weighted multiway cut problem in trees and was solved in [6]. We hope, though the details have not yet been worked out, that the present orientation method can be extended to the weighted case, as well.)

If one is interested only in computing a minimum multiway cut in the special case when $G - S$ induces a tree, then a simple greedy type algorithm is available in [5] whose proof of correctness is also very simple and does not need any kind of duality theory. The first part of the present algorithm is nothing but a reformulation of the greedy algorithm from [5]. The main novelty here lies in the second part of the algorithm where an optimal orientation is computed yielding a relatively simple proof of (2.1).

THEOREM 2.1 *Let $G = (V, E)$ be an undirected graph with a terminal set S for which $G - S$ induces a tree. Then $\vec{\nu}_S = \pi_S$, that is, the minimum cardinality of a multiway cut separating S is equal to the maximum of $\sum_{s \in S} \lambda(S - s, s; \vec{G})$ over all orientations $\vec{G}$ of G.*

Proof. We have seen that $\vec{\nu}_S \leq \pi_S$. In order to prove the equality, we are going to construct a partition $\mathcal{P}$ separating S and an orientation $\vec{G}$ of G so that

$$\varrho_{\vec{G}}(X) = \lambda(S - s, s; \vec{G}) \text{ whenever } s \in X \in \mathcal{P}. \tag{2.1}$$

Before proceeding to the proof, let us mention that there is another interpretation of the problem. Consider each element of S as a colour. Then the minimum multiway cut problem is equivalent to colouring the nodes with the available $|S|$ colours so as to minimize the number of bichromatic edges. For a given colouration we say that an edge uv is **bichromatic** if its two ends have different colours. The other edges are called **mono-chromatic.**

We may assume that S is a stable set. We also may assume without loss of generality that for every edge sv with $s \in S$ the degree of v is 2. If this is not the case, then subdivide the edge vs by a new node. Clearly, the theorem holds for the new graph if and only if it holds for the original.

Let $T = (U, F)$ denote the tree induced by $G - S$. By the assumptions we made, only the leaves of T have neighbours in S. We may furthermore assume that every leaf v actually has one neighbour in S for otherwise we may delete v without changing the problem.

Let us choose an arbitrary non-leaf node r of T and call it a **root.** The **height** $h(u)$ of a node u of T is the length of the unique path from r to u. For an edge $e = uv$ with $h(u) = h(v) - 1$ we say that node v and edge e are **above** u and that u and e are **under** v. That is, the nodes above u are exactly those neighbours of u whose height is one bigger than that of u. Furthermore every node u but the root has exactly one node under u. There is no node under the root and above a leaf. (In the literature a node above u is called a child of u and a node under u is called a parent of u.)

With the help of a depth first search (say), determine an ordering of the elements of U, described by a one-to-one mapping $f : U \rightarrow \{1, 2, \ldots, |U|\}$, in such a way that $f(r) = 1$ and $f(u) < f(v)$ whenever uv is an edge of T with $h(u) = h(v) - 1$.

The algorithm consists of two parts. In the first one we determine a partition $\mathcal{P}$ of V separating S while the second part serves for computing the orientation.

PART 1. *Computing partition $\mathcal{P}$.*

The partition $\mathcal{P}$ will be given by a function $\sigma : V \to S$ such that $\sigma(s) := s$ for $s \in S$. (In other words $\sigma(u)$ will be the colour of u.) The σ-values of the tree are computed in two phases.

In the first phase a subset $L(u)$ of S will be assigned to every node u of T, as follows. According to the ordering f of U, consider the elements u of U in a reverse order (that is, root r is considered last). If u is a leaf whose unique neighbour in S is s, then let $L(u) := \{s\}$. Suppose that u is a node for which $L(v)$ has been computed for all nodes v above u. Let $L(u)$ consist of the nodes of maximum degree of the hypergraph $\{L(v) : v$ is above $u\}$. The first phase terminates when $L(r)$ (and hence every other $L(v)$) has been computed.

Intuitively, we think of $L(u)$ as the set of candidate colours from which the final colour of u will be chosen during the second phase of Part 1. The sets $L(u)$ are determined in a downward manner (toward the root) and $L(u)$ consists of those colours which appear most often in the (already determined) candidate colour-sets of nodes above u.

In the second phase we work upward, that is we consider the elements v of U in the (forward) ordering given by f. Start at the root r and define $\sigma(r)$ to be an arbitrary member of $L(r)$. In the general step when v is considered let uv denote the unique edge of T under v. By the choice of the ordering, u precedes v and hence $\sigma(u)$ has already been determined.

(a) If $\sigma(u) \in L(v)$, then let $\sigma(v) := \sigma(u)$,
(b) if $\sigma(u) \notin L(v)$, then let $\sigma(v)$ be an arbitrary member of $L(v)$.

Intuitively, this means that the colour of root r is an arbitrary member of the candidate colour-set $L(r)$. Furthermore, if the colour $\sigma(u)$ of a node of tree T has already been determined and $e = uv$ is an edge of T above u, then the colour $\sigma(v)$ of v is always chosen from the candidate colour-set $L(v)$ of v so as to make e mono-chromatic whenever this is possible. (That is, e becomes bi-chromatic if the final colour $\sigma(u)$ of u is not in the set $L(v)$ of candidate colours of v.) This way we have determined a colouration σ of the nodes of G, or, equivalently, a partition $\mathcal{P} := \{V_s : s \in S\}$ of V where $V_s := \{u \in V ; \sigma(v) = s)\}$.

PART 2. *Computing the orientation of T.*

In the second part of the algorithm we define the orientation of the edges of G in such a way that once the orientation of an edge has been determined, it will never be changed later. Let $e = uv$ be an edge of T with $h(u) = h(v) - 1$. The orientation of e will be specified by declaring that e is either an up-edge or a down-edge. e being an **up-edge** means that e is oriented from u to v while e being a **down-edge** means that e is oriented from v to u. We will call a node v distinct from the root an **up-node** if the (unique) edge under v is an up-edge. Node v is called a **down-node** if either $u = r$ or if the edge under u is a down-edge.

Let all bi-chromatic edges be up-edges. To determine the orientation of other edges, we consider the nodes u of T in the order of their height starting with root r and determine the orientation of all the mono-chromatic edges above u. Therefore, when a node u is considered, its status of being an up-node or a down-node has already been determined. Specifically, the orientation of the mono-chromatic edges above u is determined by the following rules.

RULE 1 *If u is a down-node, then let every mono-chromatic edge above u be a down-edge.*

RULE 2 *If u is an up-node, then choose arbitrarily a mono-chromatic edge uz above u (there is one!) and, apart from uz, let all mono-chromatic edges above u be down-edge. We will call uz a* **special** *edge.*

RULE 3 *If e is an edge connecting a leaf u of T and a node s in S, then orient e so that the in-degree (and the out-degree) of u be 1.*

In other words every bi-chromatic edge is an up-edge and every mono-chromatic edge above u, with one exception in case u is a down-node, is a down-edge.

Let $\vec{G}$ denote the resulting directed graph. Henceforth our main concern is to prove that the partition $\mathcal{P}$ and the orientation $\vec{G}$ satisfy (2.1). To this end let us consider an element $s \in S$ (that is, one of the colours) and the set $F_s := \{u_1v_1, \ldots, u_kv_k\}$ of bi-chromatic edges of $\vec{G}$ for which $\sigma(v_i) = s$. That is, F_s is the set of edges of $\vec{G}$ entering the part V_s of $\mathcal{P}$ containing s.

We are going to find k edge-disjoint paths from $S - s$ to s. The existence of such paths directly implies (2.1). It follows from Rule 2 that for any up-node v_i $(i = 1, \ldots, k)$ there is a unique path P_i' in $\vec{G}$ from v_i to s consisting of special edges. Since special edges are mono-chromatic these paths are inside V_s. They are edge-disjoint (and actually node-disjoint, except at s) since no two special edges enter the same node.

Therefore, all what we have to show is that there are k edge-disjoint paths P_i'' from $S-s$ to v_i $(i=1,\ldots,k)$. By glueing together paths P_i' and P_i'' we will obtain a path P_i from a node of $S-s$ to s that uses edge u_iv_i.

Let $\vec{G}_s=(V,E_s)$ be a subgraph of $\vec{G}$ where E_s consists of three types of edges. Recall that if (a directed edge) yv is a down-edge or vy is an up-edge, then y is above v.

TYPE A *A down-edge yv belongs to E_s if $s\notin L(y)$.*

TYPE B *An up-edge vy belongs to E_s if $s\in L(y)$ and $\sigma(v)\neq s$.*

TYPE C *An edge tu of $\vec{G}$ belongs to E_s if $t\in S-s$.*

Note that a down-edge $yv\in E_s$ is mono-chromatic and $\sigma(v)=\sigma(y)\neq s$. Hence $\sigma(v)\in L(y)$. For a non-special up-edge vy, $\sigma(v)\notin L(y)$.

Let $\varrho(u)$ (respectively, $\delta(u)$) denote the number of edges in E_s entering (leaving) u.

LEMMA 2.2 *$\varrho(u)\geq\delta(u)$ for every node u of T.*

Proof. By Rule 3, the lemma holds for leaves so suppose that u is not a leaf. If $\sigma(u)=s$, then, by Rules 1 and 2, $\delta(u)=0\leq\varrho(u)$. Therefore we will assume that $\sigma(u)\neq s$. Let $A:=\{y\in U: y \text{ is above } u, s\in L(y), \sigma(u)\notin L(y)\}$ and $B:=\{y\in U: y \text{ is above } u, s\notin L(y), \sigma(u)\in L(y)\}$. Let $\alpha:=|A|, \beta:=|B|$.

Since $\sigma(u)\in L(u)$, the definition of $L(u)$ implies that

$$\beta\geq\alpha \text{ and if } \beta=\alpha, \text{ then } s\in L(u). \tag{2.2}$$

Let x denote the node under u in case $u\neq r$.

CASE 1 *u is a down-node.*

Then the edges of E_s above u that leave u are precisely the edges from u to A. Hence $\delta(u)\leq\alpha+1$ and $\delta(u)=\alpha$ if u is the root. Each edge under an element of B is a down-edge and belongs to E_s from which $\varrho(u)\geq\beta$. If $u=r$, then $\varrho(u)\geq\beta=\alpha=\delta(u)$, as required. So suppose that $u\neq r$. If $\beta\geq\alpha+1$, then $\varrho(u)\geq\delta(u)$. If $\alpha=\beta$, then (2.2) implies that $ux\notin E_s$ and hence $\varrho(u)\geq\beta=\alpha=\delta(u)$.

CASE 2 *u is an up-node.*

If $\alpha=\beta$, then $s\in L(u)$ by (2.2). Now $\sigma(x)\neq s$, since $\sigma(x)=s$ would imply $\sigma(u)=s$ which is not the case. Therefore $xu\in E_s$.

Let uz denote the special edge above u. Now $\sigma(u) = \sigma(z) \in L(z)$ and hence $z \notin A$. We distinguish two cases.

If $z \in B$, then $s \notin L(z)$ and hence $uz \notin E_s$. Therefore the edges in E_s leaving u are the edges from u to A, that is, $\delta(u) = \alpha$. For every node $y \in B - z$ the edge under y is oriented toward u. If $\beta \geq \alpha + 1$, then $\varrho(u) \geq \beta - 1 \geq \alpha = \delta(u)$. If $\beta = \alpha$, then $xu \in E_s$ and hence $\varrho(u) \geq (\beta - 1) + 1 = \alpha = \delta(u)$.

If $z \notin B$, then $\delta(u) \leq \alpha + 1$. If $\beta \geq \alpha + 1$, then $\varrho(u) \geq \beta \geq \alpha + 1 \geq \delta(u)$. If $\beta = \alpha$, then $xu \in E_s$ and hence $\varrho(u) \geq \beta + 1 = \alpha + 1 \geq \delta(u)$. •

Now we can construct the paths P_i'' $(i = 1, \ldots, k)$ in a greedy way. Starting at u_1v_1 we can go backward in E_s as long as we arrive at a node of S which is distinct from s since the tail of every edge in E_s has got a colour distinct from s. That is, we have constructed a directed path P_1'' that starts at an element of $S - s$ and its last edge is u_1v_1. After leaving out the edges of this path the property of Lemma 2.2 continues to hold, so we can repeat the construction to obtain the required edge-disjoint paths $P_1'', P_2'', \ldots, P_k''$ • •

REFERENCES

MINIMUM MULTIWAY CUTS IN TREES

Péter L. Erdős*
András Frank**
László Székely***

(January, 1996, final: March, 1998)

ABSTRACT We compare three lower bounds for the minimum cardinality of a multiway cut in a graph separating a given set S of terminals. The main result is a relatively short algorithmic proof for a simplified version of a min-max theorem of the first and the third authors asserting that the best of the three lower bounds is actually attainable if every circuit of the graph contains a terminal node.

* Mathematical Institute of the Hungarian Academy of Sciences, Reáltanoda u. 13-15, Budapest, Hungary, H-1053, e-mail: `elp@math-inst.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T016358.
** Department of Operations Research, Eötvös University, Múzeum krt. 6-8, Budapest, Hungary, H-1088 and Ericsson Traffic Laboratory, Laborc u.1, Budapest, Hungary H-1037. e-mail: `frank@cs.elte.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T17580.
*** Department of Computer Science, Eötvös University, Múzeum krt. 6-8, Budapest, Hungary, H-1088. e-mail: `szekely@cs.elte.hu` Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T016358 and by NSF DMS 9701211.