

On weighted multiway cuts in trees

Péter L. Erdős^{*,a}, László A. Székely^{**,b}

^a*Centrum voor Wiskunde en Informatica, 1098 SJ Amsterdam, Netherlands*
Mathematical Institute of the Hungarian Academy of Sciences, H-1055 Budapest, Hungary

^b*Department of Computer Science, Eötvös University, H-1088 Budapest, Hungary*
Department of Mathematics, University of New Mexico, Albuquerque, NM 87131, USA

Received 11 September 1991; revised manuscript received 1 April 1993

Abstract

A min–max theorem is developed for the multiway cut problem of edge-weighted trees. We present a polynomial time algorithm to construct an optimal dual solution, if edge weights come in unary representation. Applications to biology also require some more complex edge weights. We describe a dynamic programming type algorithm for this more general problem from biology and show that our min–max theorem does not apply to it.

AMS 1991 Subject Classifications: 05C05, 05C70, 90C27

Keywords: Multiway cut; Menger's theorem; Tree; Duality in linear programming; Dynamic programming

1. Introduction

Let $G = (V, E)$ be a simple graph, $C = \{1, 2, \dots, r\}$ be a set of colours. For $N \subseteq V(G)$, a map $\chi: N \rightarrow C$ is a *partial colouration*. We usually think of a given partial colouration. A map $\bar{\chi}: V(G) \rightarrow C$ is a *colouration* if $\chi(v) = \bar{\chi}(v)$ holds for all $v \in N$.

A *colour dependent weight function* assigns to every edge (p, q) and colours i, j a natural number $w(p, q; i, j)$, which tells the weight of the edge (p, q) in a colouration $\bar{\chi}$, in which $\bar{\chi}(p) = i$, $\bar{\chi}(q) = j$. We assume that $w(p, q; i, i) = 0$ and $w(p, q; i, j) = w(q, p; j, i)$. We say that w is *colour independent*, if for any (p, q) , $i_1 \neq j_1$, $i_2 \neq j_2$, we have $w(p, q; i_1, j_1) = w(p, q; i_2, j_2)$. We say that w is *edge independent*, if for any $(p_1, q_1) \in E$ and $(p_2, q_2) \in E$, and

*Corresponding author.

**Research of the author was supported by the A. v. Humboldt-Stiftung and the U.S. Office of Naval Research under the contract N-0014-91-J-1385.

$i, j \in C$, we have $w(p_1, q_1; i, j) = w(p_2, q_2; i, j)$. (Hence, any edge independent weight function satisfies $w(p, q; i, j) = w(p, q; j, i)$.) We say that w is *constant*, if it is colour and edge independent.

An edge (p, q) is *colour-changing* in the colouration $\bar{\chi}$, if $\bar{\chi}(p) \neq \bar{\chi}(q)$. The *changing number* of the colouration $\bar{\chi}$ is the sum of weights of the colour-changing edges in $\bar{\chi}$, i.e.:

$$\text{change}(G, \bar{\chi}) = \sum_{(p, q) \in E(G)} w(p, q; \bar{\chi}(p), \bar{\chi}(q)).$$

A partial colouration χ defines a partition of N by $N_i = \{v \in N: \chi(v) = i\}$. A set of edges that separates every N_i from all the other N_j 's is termed a *multiway cut* [1]. Observe that the set of colour-changing edges of a colouration $\bar{\chi}$ forms a multiway cut and every multiway cut is represented in this way.

The *length* of the pair (G, χ) is the minimum weight of a multiway cut, in formula:

$$l(G, \chi) = \min\{\text{change}(G, \bar{\chi}): \bar{\chi} \text{ colouration}\}.$$

An *optimal colouration* is a colouration $\bar{\chi}$ such that $\text{change}(G, \bar{\chi}) = l(G, \chi)$.

The multiway cut problem for colour independent weight functions has been extensively studied in combinatorial optimization (e.g. [1–3]). As Dahlhaus et al. pointed out [3], this problem is NP-hard, even for $|N| = 3$, $|N_i| = 1$ and constant weight.

On the other hand, if we restrict ourselves to planar graphs, a fixed number of colours, and constant weight, then the problem becomes solvable in polynomial time [3]. A well-known specialization of the multiway cut problem, which is solvable in polynomial time, is $r = 2$, which is considered in the undirected edge version of Menger's theorem [8].

Although it is less known in the operations research community, some instances of the multiway cut problem have great importance in biomathematics. In fact, the notions of the changing number and the length came from genetics and we follow the terminology used there. For the case of constant weight function, Fitch [6] and Hartigan [7] developed a polynomial time algorithm to determine the length of a given tree. Sankoff and Cedergren [13], and Williamson and Fitch [12] studied edge independent weight functions and made polynomial time algorithms to find the length. Some explanation of the significance of the multiway cut problem in biology is given in [4, 5].

The goal of the present paper is to study the multiway cut problem. In Section 2 we give a new lower bound for the length of a multiway cut. Section 3 provides a dynamic programming type algorithm to find the length of a tree with an arbitrary weight function. Section 4 uses the algorithm of Section 3 to establish a min–max theorem for the multiway cut problem of trees, in the case of colour independent weight functions. All the results can be extended to any graph G , in which N intersects every cycle. Section 5 describes our results in terms of linear programming.

A preliminary version of the present paper has already appeared [5]. We are indebted to the anonymous referees for their helpful observations that we use in this presentation.

2. Lower bound for the weight of a multiway cut

Let G be a simple graph, $N \subseteq V(G)$ and $\chi: N \rightarrow C$ be a partial colouration. Let w be a colour dependent weight function.

Definition. An oriented path P in G starting at $s(P) \in N$ and terminating at $t(P) \in N$ is a *colour-changing path*, if $\chi(s(P)) \neq \chi(t(P))$ and P has no internal vertex in N . (From now on path means oriented path, unless we explicitly say the opposite.) Let us fix a family \mathcal{P} of colour-changing paths and let $e = (p, q) \in E(G)$. Define

$$n_i(e, \mathcal{P}) = \#\{P \in \mathcal{P}: (p, q) \in P \text{ and } \chi(t(P)) = i\}.$$

The notation $(p, q) \in P$ means that P enters the edge (p, q) at p and leaves at q .

Definition. Let $\chi: N \rightarrow C$ be a partial colouration and $\bar{\chi}$ be a colouration on G . A family \mathcal{P} of colour-changing paths is a *path packing*, if all pairs of colours $i \neq j$ and all edges (p, q) satisfy

$$n_i((p, q), \mathcal{P}) + n_j((q, p), \mathcal{P}) \leq w(p, q; j, i).$$

The maximum cardinality of a path packing is denoted by $p(G, \chi)$.

Theorem 1. For any graph G and partial colouration χ , we have

$$l(G, \chi) \geq p(G, \chi).$$

Proof. Let \mathcal{P} be a path packing and $\bar{\chi}: V(G) \rightarrow C$ be an optimal colouration. Define a map $f: \mathcal{P} \rightarrow E(G)$ as follows: let $f(P) = e$ if e is the last colour-changing edge in P in $\bar{\chi}$. For any colour changing edge $e = (p, q)$, $\bar{\chi}(p) = j$ and $\bar{\chi}(q) = i$ ($i \neq j$ since e is colour changing), we have

$$\#\{P \in \mathcal{P}: f(P) = e\} \leq n_i((p, q), \mathcal{P}) + n_j((q, p), \mathcal{P}) \leq w(p, q; j, i).$$

Therefore,

$$|\mathcal{P}| \leq \text{change}(G, \bar{\chi}) = l(G, \chi). \quad \square$$

3. An algorithm to find optimal colourations

Now we focus on the multiway cut problem of trees. Let T be a tree and $\chi: N \rightarrow C$ be a partial colouration, and let $L(T)$ denote the set of leaves, i.e. vertices of degree 1. We assume $N = L(T)$. (It is obvious that the solution of the multiway cut problem of trees with $N = L(T)$ easily generalizes to the solution of the multiway cut problem of trees with arbitrary N .) Let w be a colour dependent weight function. In this section we give a polynomial time algorithm to determine all optimal colouration of T for the weight w .

Let us fix an arbitrary non-leaf vertex, the **root** of T . Let (u, v) be an edge and let v be closer to the **root** than u , then we say $v = \mathbf{Father}(u)$. ($\mathbf{Father}(\mathbf{root})$ is **NIL**.) We denote the set of all u for which $v = \mathbf{Father}(u)$ by $\mathbf{Son}(v)$.

Our colouring algorithm has two phases. Starting from the leaves and approaching the **root** we determine a *penalty function* of every vertex v recursively, and subsequently we determine a suitable colouration $\bar{\chi}$ starting from the *root* and spreading to the leaves.

Definition. The vector-valued *penalty function* is a map

$$\mathbf{pen}: V(T) \rightarrow (\mathbb{N} \cup \{\infty\})^r,$$

such that $\mathbf{pen}_i(v)$ means the length of the subtree separated by v from the **root**, if the colour of v has to be i .

Phase I. For every leaf $v \in L(T)$ let

$$\mathbf{pen}_i(v) = \begin{cases} 0 & \text{if } v \in N_i, \\ \infty & \text{otherwise,} \end{cases}$$

where in an actual computation ∞ may be substituted by a sufficiently large number. Take a vertex v , such that $\mathbf{pen}(v)$ is not computed yet for the vertex v , but $\mathbf{pen}(u)$ is already known for every vertex $u \in \mathbf{Son}(v)$. Then compute

$$\mathbf{pen}_i(v) = \sum_{u \in \mathbf{Son}(v)} \min_{j=1, \dots, r} \{w(u, v; j, i) + \mathbf{pen}_j(u)\}.$$

Phase II. Now we determine an optimal colouration $\bar{\chi}$ of T . First, let $\bar{\chi}(\mathbf{root})$ be a colour i , which minimizes the value $\mathbf{pen}_i(\mathbf{root})$. Furthermore, for a vertex v for which $\bar{\chi}(v)$ is not settled yet, but $\bar{\chi}(\mathbf{Father}(v))$ is already determined, let $\bar{\chi}(v)$ be a colour i , which minimizes the expression

$$w(v, \mathbf{Father}(v); i, \bar{\chi}(\mathbf{Father}(v))) + \mathbf{pen}_i(v).$$

It is easy to see, that every leaf $v \in N_i$ satisfies $\bar{\chi}(v) = i = \chi(v)$, for $i = 1, \dots, r$.

The correctness of this algorithm is almost self-explanatory. Assume the positive integer edge weights are given in unary representation. Then, the time complexity is $O(n \cdot r^2 \cdot (\max \text{weight}))$, since at each step we calculate r^2 sums, take the minimum, and roughly $2n$ steps are necessary because T has n vertices and $n - 1$ edges. You may change $\max \text{weight}$ for $\log(\max \text{weight})$, if the edge weights come in binary representation.

In the rest of this section we focus on colour independent weight functions, since we can develop a slightly more efficient version of this algorithm, which also can determine all optimal colourations. Biologists may need all optimal colourations; the saving in running time comes from avoiding the second minimization in Phase II. Also, case (A2) in the proof of Theorem 2 will need the modified algorithm. For the sake of simplicity, for the rest of this section the weight function is a map $w: E(T) \rightarrow \mathbb{N}$ for colour changing edges

and the weight of any edge not changing colour is 0. We use the usual *Kronecker delta* notation.

Phase I'. For every leaf v , set

$$M_1(v) = M_2(v) = \{i: \text{pen}_i(v) = 0\}.$$

If $\text{pen}(v)$ is not computed yet for the vertex v but $\text{pen}(u)$ is already known for every vertex $u \in \text{Son}(v)$, then set

$$\text{pen}_i(v) = \sum_{u \in \text{Son}(v)} \min_{j=1, i, r} \{(1 - \delta_{ij})w(u, v) + \text{pen}_j(u)\}.$$

Let $p(v) = \min_i \text{pen}_i(v)$, and

$$M_1(v) = \{i \in \{1, \dots, r\}: \text{pen}_i(v) = p(v)\},$$

$$M_2(v) = \{i \in \{1, \dots, r\}: \text{pen}_i(v) < p(v) + w(v, \text{Father}(v))\}.$$

It is obvious that $M_1(v) \subseteq M_2(v)$.

Phase II'. For $\bar{\chi}(\text{root})$, take an arbitrary element of $M_1(\text{root})$. If $\bar{\chi}(v)$ is not settled yet for a vertex v , but $\bar{\chi}(\text{Father}(v))$ is already determined, take

$$\bar{\chi}(v) = \begin{cases} \bar{\chi}(\text{Father}(v)) & \text{if } \bar{\chi}(\text{Father}(v)) \in M_2(v), \\ \text{an arbitrary element of } M_1(v) & \text{otherwise.} \end{cases}$$

It is easy to see, that every vertex $v \in N_i$ satisfies $\bar{\chi}(v) = i = \chi(v)$, for $i = 1, \dots, r$. This algorithm is obviously correct and permitting some extra freedom at certain steps, any optimal colouration can be obtained by the modified algorithm. For this purpose we introduce a third set of colours at Phase I':

$$M_3(v) = \{i \in \{1, \dots, r\}: \text{pen}_i(v) = p(v) + w(v, \text{Father}(v))\}.$$

If in Phase II' we also allow to give the colour of $\bar{\chi}(\text{Father}(v))$ to v , if $\bar{\chi}(\text{Father}(v)) \in M_3(v)$, then the algorithm still yields an optimal colouration. Moreover, one can prove that running this algorithm in all possible ways yields all optimal colourations. (We leave the proof to the reader.) The complexity of this revised algorithm is better by a constant multiplicative factor than that of the original, but to get every optimal colouration may take exponential time, since M.A. Steel exhibited trees with exponentially many optimal colourations [11].

4. A min-max theorem

In this section we assume that the weight function is *colour-independent* and we prove that the lower bound of Theorem 1 is tight for leaf-coloured trees, and then even for a larger class of graphs.

Theorem 2. *Let T be an arbitrary tree with colour-independent weight function $w: E(T) \rightarrow \mathbb{N}$ and with leaf-colouration $\chi: L(T) \rightarrow C$. Then*

$$l(T, \chi) = p(T, \chi) .$$

We already know from Theorem 1 that the LHS is greater or equal than the RHS. We have to prove the other inequality. For this end we construct the desired optimal path packing in a recursive manner. At first, we explicitly construct optimal path packings for stars, i.e. for trees with 1 branching vertex. Then, for a tree T with at least 2 branching vertices and with

$$W(T) = \sum_{f \in E(T)} w(f)$$

sum of weights, we define a ‘smaller’ tree T' for which we can trace back the problem of the construction of an optimal path packing, such that we can ‘lift up’ the path packing from T' to T to get the solution. We may have at most $W(T)$ ‘lift up’ steps. Here we give the details.

For convenience, we want to use the functions **Son** and **Father**, therefore we fix, as in Section 3, a **root** of T . In the complexity issues we assume that our tree is represented by the vertices v and the sets **Son**(v) and **Father**(v), furthermore every element of **Son**(v) and **Father**(v) (which represents edges) also contains the weight of the edge. The paths under construction will be represented as double-linked lists, therefore, due to Theorem 1, the space complexity of the representation is $O(l(T, \chi) \cdot n)$.

Definition. We say that a vertex v is of order 1 if every element of **Son**(v) is a leaf.

Notice that every tree with at least 2 branching vertices has a non-root vertex of order 1. Before starting the main body of the proof we need the following lemma.

Lemma 1. *One can assume that no vertex of order 1 has two sons with the same colour.*

Let v be a vertex of order 1, such that **Son**(v) contains at least 2 leaves with identical colour. Let $\Sigma(T)$ denote the tree obtained from T by identification of the elements of **Son**(v) with identical colour and adding up their edge weights, respectively. Now one can easily construct an optimal path packing for T from an optimal path packing of $\Sigma(T)$. Anyhow, we give a formal proof, otherwise, the base case of our recursive algorithm would not be complete.

Proof. Define the tree $\Sigma(T)$ formally as follows: let the tree T' be a star with midpoint v and with leaves $\{l_i: \exists u \in \text{Son}(v) \text{ with } \chi(u) = i\}$ and let $\Sigma(T)$ be the tree made of the trees $T \setminus \text{Son}(v)$ and T' by identification of their common v . The leaf-colouration and weight function of $\Sigma(T)$ are as follows:

$$\chi'(u) = \begin{cases} \chi(u) & \text{if } u \in L \setminus \text{Son}(v) , \\ i & \text{if } u = l_i , \end{cases}$$

$$w'(f) = \begin{cases} \sum_{\substack{u \in \text{Son}(v) \\ \chi(u)=i}} w((u, v)) & \text{if } f = (l_i, v), \\ w(f) & \text{otherwise.} \end{cases}$$

Notice that $l(\Sigma(T), \chi') = l(T, \chi)$.

Claim. If $l(\Sigma(T), \chi') = p(\Sigma(T), \chi')$ then $l(T, \chi) = p(T, \chi)$.

Proof. Let $\text{Son}(v)$ contain d different colours. We apply induction on $|\text{Son}(v)|$.

Base case: if $|\text{Son}(v)| = d$, then $\Sigma(T) = T$, $\chi = \chi'$, and we have nothing to prove.

Inductive step: Suppose that we know Lemma 1 for all $|\text{Son}(v)| < k$. Assume now $|\text{Son}(v)| = k$ and for some fixed $z_1, z_2 \in \text{Son}(v)$, let $\chi(z_1) = \chi(z_2)$. Join z_1 and z_2 into z . In the new tree T^* obtained by identification, define the leaf colouration and the weight function as follows:

$$\chi^*(u) = \begin{cases} \chi(u) & \text{if } u \neq z_1, z_2, \\ \chi(z_1) & \text{if } u = z, \end{cases}$$

$$w^*(f) = \begin{cases} w(f) & \text{if } f \neq (v, z_i), \\ w(v, z_1) + w(v, z_2) & \text{if } f = (v, z). \end{cases}$$

Now we have $\Sigma(T) = \Sigma(T^*)$, therefore $l(\Sigma(T)) = l(\Sigma(T^*))$. By the hypothesis there exists a path packing \mathcal{P}^* in the tree T^* satisfying $|\mathcal{P}^*| = l(T^*)$. It is easy to divide the paths of \mathcal{P}^* adjacent to vertex z into two groups, such that the members of one group are adjacent to z_1 and the members of the other are adjacent to z_2 and both groups obey the weight restriction on the edge adjacent to z_i . In this way we obtain a path packing of $l(T)$ members in T . This proves the Claim as well as Lemma 1. \square

The time complexity of this algorithm is $O(\sum_{u \in \text{Son}(v)} w(u, v))$ so the time complexity of all applications of Lemma 1 altogether is $O(W(T))$.

We return to the main body of the proof; we assume that any two sons of an arbitrary vertex of order 1 have different colours. Our algorithm is given in a recursive form in the variables $b(T)$ and $W(T)$, where $b(T)$ is the number of branching (non-leaf) vertices of T .

Base case: let $b(T) = 1$ and $W(T)$ be arbitrary. Then T is a star; let v denote the midpoint of it. Due to Lemma 1 we may assume that $|L(T)| = r$ (i.e. every colour occurs once). Assume that the edge (v, u) has maximum weight over all edges. Orient paths from u to every other leaf $z \in L(T) \setminus \{u\}$ with multiplicity $w(v, z)$. This path system is obviously a path packing and has $l(T)$ members. This case requires $O(W(T))$ steps.

Recursive step: For any tree T with at least 2 branching vertices we shall find ‘smaller’ tree T' with fewer branching vertices ($b(T') < b(T)$) or with smaller total weights

($b(T') = b(T)$ and $W(T') < W(T)$) such that an optimal path packing of T' can be lifted up to an optimal path packing of T . Define

$$s(v) = \max_{u \in \text{Son}(v)} w(v, u) .$$

We distinguish two cases:

(A) There is a vertex v of order 1 such that $s(v) \neq w(v, \mathbf{Father}(v))$.

(B) $s(v) = w(v, \mathbf{Father}(v))$ for every vertex v of order 1.

Case (A). Let $\bar{\chi}$ be an optimal colouration of T such that v is the first branching vertex for which the colour sets M_i were determined. We have two subcases; in (A1) we have $s(v) > w(v, \mathbf{Father}(v))$, in (A2) we have $s(v) < w(v, \mathbf{Father}(v))$.

Case (A1). Let T'' be the tree with the same vertex set, edge set and leaf colouration as the tree T was, and let the new weight function $w' : E(T) \rightarrow \mathbb{N}$ such that

$$w'(f) = \begin{cases} w(f) - 1 & \text{if } f = (v, u) \text{ where } u \in \text{Son}(v) , \\ w(f) & \text{if otherwise .} \end{cases}$$

If $w'(f) = 0$, then cancel this edge and its leaf endpoint from the tree T'' to obtain the tree T' . Due to our colouring algorithm, colouration $\bar{\chi}$ is also optimal for the tree T' , therefore

$$l(T') + (|\text{Son}(v)| - 1) = l(T) .$$

The total weight of tree T' is less than of T . Assume now that we have an optimal path packing \mathcal{P}' of $l(T', \chi)$ elements in T' . Denote by ΔT the star of $v \cup \text{Son}(v)$ with weight function $w \equiv 1$ and with the original leaf colouration. Let $\Delta \mathcal{P}$ be optimal path packing in ΔT (use the base case). Now the path system $\mathcal{P} = \mathcal{P}' \cup \Delta \mathcal{P}$ is obviously optimal path packing in the tree T .

We can construct T' and the path packings $\Delta \mathcal{P}$ and \mathcal{P} from the given tree T and path packing \mathcal{P}' in $O(r \cdot \sum_{u \in \text{Son}(v)} w(v, u))$ time, so that the total time complexity of the case (A1) is $O(rW(T))$.

Case (A2). Now we have $s(v) < w(v, \mathbf{Father}(v))$. Let the tree T' be identical with the tree T with the same leaf-colouration and with the weight function

$$w'(f) = \begin{cases} s(v) & \text{if } f = (v, \mathbf{Father}(v)) , \\ w(f) & \text{otherwise .} \end{cases}$$

Now it is easy to see that there exists an optimal colouration $\bar{\chi}$ of T' satisfying $\bar{\chi}(v) = \bar{\chi}(\mathbf{Father}(v))$ which is also optimal in T . (The only problem that can occur is that $\bar{\chi}(\mathbf{Father}(v)) \in M_2(v)$ but $\bar{\chi}(\mathbf{Father}(v)) \in M'_3(v)$. In that case we can apply the extended Phase II'.) Therefore, we have $l(T) = l(T')$ and $W(T') < W(T)$. Now we can easily 'lift up' any optimal path packing \mathcal{P} of T' to the tree T , namely \mathcal{P} itself is obviously path packing in T .

This operation takes $O(1)$ time, so the total time complexity of case (A2) is $O(n)$.

Case (B). From now on we assume that every vertex z of order 1 satisfies the condition $s(z) = w(z, \mathbf{Father}(z))$. For the rest of (B), we fix a vertex v ; if the diameter of T is 3, then

let v be the **root**, otherwise, let v be a non-root vertex such that $\text{Son}(v) \not\subset L(T)$ and every non-leaf son is a vertex of order 1 (the existence of such a v is obvious). Let the non-leaf sons of v be the vertices z_1, \dots, z_k .

By the definition of case (B) it is easy to see the existence of an optimal coloration $\bar{\chi}$ colouring v and every z_i to the same colour. Therefore if \bar{T} is the tree derived from the tree T by contracting every edge of form (v, z_i) (leaving the name of the new vertex v), which is endowed with the original leaf-colouration and weight function on the existing edges, then the restriction of the same colouration $\bar{\chi}$ is also optimal for \bar{T} and $l(\bar{T}) = l(T)$. On the other hand, the tree \bar{T} has less branching vertices than T .

Now due to our hypothesis we have an optimal path packing $\bar{\mathcal{P}}$ in the tree \bar{T} . Therefore

$$|\bar{\mathcal{P}}| = l(T).$$

Let us define the *lift up* $\mathcal{P} = \{\hat{P} : P \in \bar{\mathcal{P}}\}$ of the path packing $\bar{\mathcal{P}}$, where \hat{P} is identical with P if no leaf u of $\text{Son}(z_i)$ ($i = 1, \dots, k$) belongs to the path P , and \hat{P} comes from P by subdivision of the edge (v, u) with vertex z_i if $\text{endvertex}(P) = u \in \text{Son}(z_i)$ ($i = 1, \dots, k$). We have $l(T)$ many elements in \mathcal{P} .

Let $e_i = (v, z_i)$ (for every $i = 1, \dots, k$). For an edge $f = (p, q)$, we write $-f = (q, p)$. Now, by the definition of \mathcal{P} , the condition

$$n_i(f, \mathcal{P}) + n_j(-f, \mathcal{P}) \leq w(f)$$

holds for every edge $f \neq e_i$ ($i = 1, \dots, k$), but unfortunately this is not necessarily the case for the edges e_i .

We solve this problem in a slightly more general setting (Lemma 2). For this we introduce the following notations: Let $[x]^+$ denote x , if x is non-negative, 0, if x is non-positive. Define the *badness* of the colour changing path system \mathcal{P} by

$$\text{bad}(\mathcal{P}) = \sum_{\substack{(i, j) \in C \times C \\ i \neq j}} \sum_{e \in E(G)} [n_i(e, \mathcal{P}) + n_j(-e, \mathcal{P}) - w(e)]^+.$$

Call an edge *oversaturated* by the path system \mathcal{P} , if the contribution of the edge to the badness is positive. (We recall the definition $e_i = (v, z_i)$.)

Lemma 2. Let \mathcal{P} be a system of colour-changing paths on the tree T such that

- (i) for all i, j , $n_j(\pm e_i, \mathcal{P}) \leq w(e_i)$,
- (ii) \mathcal{P} does not oversaturate any edge from $E(T) \setminus \{e_1, \dots, e_k\}$.

Then there exists a path packing \mathcal{P}^* in T of the same size.

Proof. If $\text{bad}(\mathcal{P}) = 0$ then \mathcal{P} itself is a path packing. Suppose $\text{bad}(\mathcal{P}) > 0$, and, say, the edge e_1 is oversaturated with colours 1 and 2, i.e.

$$n_1(e_1, \mathcal{P}) + n_2(-e_1, \mathcal{P}) > w(e_1) .$$

Take a path $P_1 \in \mathcal{P}$ such that $e_1 \in P_1$ and $\chi(t(P_1)) = 1$ (where, say, $t(P_1) \in \text{Son}(z_1)$), and a path $P_2 \in \mathcal{P}$ such that $-e_1 \in P_2$ and $\chi(t(P_2)) = 2$ (where $t(P_2) \notin \text{Son}(z_1)$ and $s(P_2) \in \text{Son}(z_1)$). Now we distinguish the cases (BA) and (BB):

Case (BA). Suppose there is no $P_3 \in \mathcal{P}$ for which $-e_1 \in P_3$, $s(P_3) = s(P_2)$ and $\chi(t(P_3)) = 1$. In this case we define the following path system:

$$\mathcal{P}_1 = \mathcal{P} \cup \{P\} \setminus \{P_1\} ,$$

where the path P is $(s(P_2), z_1, t(P_1))$, oriented from left to right.

Claim A.

$$\text{bad}(\mathcal{P}_1) \leq \text{bad}(\mathcal{P}) - 1 .$$

Proof. It is easy to see that $n_i(\pm f, \mathcal{P}_1) \leq n_i(\pm f, \mathcal{P})$ for each $i = 1, \dots, k$ and for each $f \in E(T) \setminus \{e_1, (z_1, s(P_2))\}$, furthermore

$$n_i(-e_1, \mathcal{P}_1) = n_i(-e_1, \mathcal{P}), \quad i = 1, \dots, k ,$$

$$n_i(e_1, \mathcal{P}_1) = n_i(e_1, \mathcal{P}), \quad i = 2, \dots, k ,$$

$$n_1(e_1, \mathcal{P}_1) = n_1(e_1, \mathcal{P}) - 1 .$$

Finally, for the edge $f_2 = (z_1, s(P_2))$ we have

$$n_i(f_2, \mathcal{P}_1) = n_i(f_2, \mathcal{P}), \quad i = 1, \dots, k ,$$

$$n_i(-f_2, \mathcal{P}_1) = n_i(-f_2, \mathcal{P}), \quad i = 2, \dots, k ,$$

$$n_1(-f_2, \mathcal{P}_1) + n_i(f_2, \mathcal{P}_1) \leq w(f_2), \quad i = 1, \dots, k .$$

The last inequality is true, since otherwise $n_2(-f_2, \mathcal{P}) + n_i(f_2, \mathcal{P}) > w(f_2)$ would hold, contradicting the assumptions of Lemma 2. \square

Case (BB). Suppose there exists a path P_3 which was forbidden in (BA). Then let \mathcal{P}_1 be the following path system:

$$\mathcal{P}_1 = \mathcal{P} \cup \{P, P_3 \wedge P_1\} \setminus \{P_1, P_3\}$$

where $P_3 \wedge P_1$ denotes the (unique) path oriented from $s(P_3)$ to $t(P_1)$.

Claim B.

$$\text{bad}(\mathcal{P}_1) \leq \text{bad}(\mathcal{P}) - 1 .$$

Proof. Set

$$E_1 = \{e_1, (z_1, t(P_1)), (z_1, s(P_3))\} \quad \text{and} \quad E_2 = E(P_1) \cup E(P_2) \setminus E(P_3 \wedge P_1) .$$

Then for each edge $f \in E(T) \setminus (E_1 \cup E_2)$ the estimates of Claim A hold. Furthermore, for $f \in E_1$ we have

$$\begin{aligned} n_i(\pm f, \mathcal{P}_1) &= n_i(\pm f, \mathcal{P}), \quad i = 2, \dots, k, \\ n_1(\pm f, \mathcal{P}_1) &\leq n_1(\pm f, \mathcal{P}), \\ n_i(\pm(z_1, t(P_1)), \mathcal{P}_1) &= n_i(\pm(z_1, t(P_1)), \mathcal{P}), \quad i = 1, \dots, k, \\ n_i(\pm e_1, \mathcal{P}_1) &= n_i(\pm e_1, \mathcal{P}), \quad i = 2, \dots, k, \\ n_1(\pm e_1, \mathcal{P}_1) &= n_1(\pm e_1, \mathcal{P}) - 1, \\ n_i(\pm(z_1, s(P_3)), \mathcal{P}_1) &= n_i(\pm(z_1, s(P_3)), \mathcal{P}) \quad i = 1, \dots, k. \end{aligned}$$

The equalities and inequalities above prove Claim B. \square

The surgeries described in Case (BA) and Case (BB) obviously keep the conditions of Lemma 2, therefore they may be repeated until the badness drops to 0. Claims A and B guarantee, that we finally reach 0. Lemma 2 and Theorem 2 are proved. \square

The determination of the tree \bar{T} takes $O(n)$ steps, therefore the total time complexity of this procedure is $O(nb(T))$. To lift up the paths from $\bar{\mathcal{P}}$ to \mathcal{P} takes

$$O\left(r \sum_{z \in \text{Son}(v)} w(v, z)\right)$$

time, therefore the total time complexity of lift up operations is $O(rW(T))$. Finally, the badness at Lemma 2 is at most

$$\sum_{z \in \text{Son}(v)} w(v, z)$$

and every edge can occur at most one application of Lemma 2 so the total time complexity of Lemma 2 is $O(\max\{rW(T), n^2\})$.

The bookkeeping of (edge, path) incidences is necessary. A possible execution of this task is to build up lists for every edge to store these incidences and to maintain these lists at every ‘lift up’ step. The total time complexity of our recursive procedure is $O(\max\{rW(T), n^2\})$, so it is unary polynomial.

The following theorem is an easy consequence of Theorem 2.

Theorem 3. *Let G be a graph with a weight function $w: E(T) \rightarrow \mathbb{N}$ and with a partial colouration $\chi: N \rightarrow C$. Assume that N intersects every cycle of G . Then*

$$l(G, \chi) = p(G, \chi)$$

Proof. Obtain a forest by eliminating the vertices of N and making leaves from the edges that were adjacent to them. Give the colour of n to the leaves that substitute a former $n \in N$. Apply Theorem 2 for each and every tree in the forest. \square

5. The LP connection

One may consider the following linear programs related to the multiway cut problem with colour independent weight function. Note that this is something, which is different from the usual multiway cut polyhedron [1].

For every oriented edge (p, q) of G and every ordered pair of distinct colours ij define a variable $z_{pq,ij}$. If $q \in N$, then eliminate $z_{pq,ij}$ and $z_{qp,ji}$ for every $j \neq \chi(q)$. Introduce new quotient variables by identifying the surviving variables $z_{pq,ij}$ and $z_{qp,ji}$ in pairs. For convenience we use the same notation for the quotient variables. Then the primal linear program is:

$$z_{pq,ij} \geq 0;$$

for every colour-changing path P_{ab} ($a, b \in N$), have

$$\sum_{(p, q) \in P_{ab}} \sum_{i: i \neq \chi(b)} z_{pq, i\chi(b)} \geq 1;$$

$$\min \sum z_{pq,ij} w(p, q),$$

where the last sum is for all quotient variables. To describe the dual linear program, for every colour-changing path P_{ab} introduce a variable λ_{ab} , such that

$$\lambda_{ab} \geq 0;$$

for every quotient variable $z_{pq,ij}$, have

$$\sum_{\substack{\chi(b)=j \\ (p, q) \in P_{ab}}} \lambda_{ab} + \sum_{\substack{\chi(v)=i \\ (q, p) \in P_{uv}}} \lambda_{uv} \leq w(p, q);$$

$$\max \sum \lambda_{ab}.$$

We claim that these linear programs have integer optimal solutions. It is easy to see, that

$$\begin{aligned} p(G, \chi) &\leq \max \sum \lambda_{ab} : \lambda_{ab} \text{ integer} \leq \max \sum \lambda_{ab} = \min \sum z_{pq,ij} w(p, q) \\ &\leq \min \sum z_{pq,ij} w(p, q) : z_{pq,ij} \text{ integer} \leq l(G, \chi). \end{aligned}$$

Only the first and last inequalities require proofs from the chain of inequalities above. The first one holds, since any path packing provides a feasible integer solution for the second linear program. The last one holds, since we have an optimal colouration $\bar{\chi}$ with total weight

of the colour-changing edges of $l(G, \chi)$; define $z_{pq,ij} = 1$, iff (p, q) is a colour-changing edge in the optimal colouration $\bar{\chi}$ and $\bar{\chi}(p) = i$, $\bar{\chi}(q) = j$ hold, and $z_{pq,ij} = 0$ otherwise. If $l(G, \chi) = p(G, \chi)$, then equality holds everywhere in the chain.

It is a natural question whether these linear programs are totally dual integral [10], i.e., whether they have integer optimal solutions for colour dependent weight functions $w(p, q; i, j)$. Unfortunately, this is not the case, take for example the 3-star with center c and leaves x, y, z with colours $\chi(x) = 1$, $\chi(y) = 2$ and $\chi(z) = 3$; and the weight function $w(c, .; i, j) = {}_iW_j$ defined by the matrix

$$W = \begin{pmatrix} 0 & 1 & 3 \\ 3 & 0 & 1 \\ 1 & 3 & 0 \end{pmatrix}.$$

References

- [1] S. Chopra and M.R. Rao, "On the multiway cut polyhedron," *Networks* 21 (1991) 51–89.
- [2] W.H. Cunningham, "The optimal multiterminal cut problem," *DIMACS Series in Discrete Math.* 5 (1991) 105–120.
- [3] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P. Seymour and M. Yannakakis, "The complexity of multiway cuts," extended abstract (1983).
- [4] P.L. Erdős and L.A. Székely, "Evolutionary trees: an integer multicommodity max–flow–min–cut theorem," *Advances in Applied Mathematics* 13 (1992) 375–389.
- [5] P.L. Erdős and L.A. Székely, "Algorithms and min–max theorems for certain multiway cut," in: E. Balas, G. Cornuéjols and R. Kannan, eds., *Integer Programming and Combinatorial Optimization*, Proceedings of the Conference held at Carnegie Mellon University, May 25–27, 1992, by the Mathematical Programming Society (CMU Press, Pittsburgh, 1992) 334–345.
- [6] W.M. Fitch, "Towards defining the course of evolution. Minimum change for specific tree topology," *Systematic Zoology* 20 (1971) 406–416.
- [7] J.A. Hartigan, "Minimum mutation fits to a given tree," *Biometrics* 29 (1973) 53–65.
- [8] L. Lovász and M.D. Plummer, *Matching Theory* (North-Holland, Amsterdam, 1986).
- [9] K. Menger, "Zur allgemeinen Kurventheorie," *Fundamenta Mathematicae* 10 (1926) 96–115.
- [10] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (John Wiley & Sons, New York, 1988).
- [11] M. Steel, "Decompositions of leaf-coloured binary trees," *Advances in Applied Mathematics* 14 (1993) 1–24.
- [12] P.L. Williams and W.M. Fitch, "Finding the minimal change in a given tree," in: A. Dress and A. v. Haeseler, eds., *Trees and Hierarchical Structures*, Lecture Notes in Biomathematics 84 (1989) 75–91.
- [13] D. Sankoff and R.J. Cedergren, "Simultaneous comparison of three or more sequences related by a tree," in: D. Sankoff and J.B. Kruskal, eds., *Time Wraps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison* (Addison-Wesley, London, 1983) 253–263.