DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 49, 1999

# FINDING MINIMUM WEIGHTED GENERATORS OF A PATH SYSTEM

ANDRÁS FRANK

ABSTRACT. In a recent paper the author [5] described a short algorithmic proof of a min-max theorem of E. Győri [8] on minimum generators of a system of subpaths of an underlying path. A. Frank and T. Jordán [4] generalized Győri's theorem in several ways; the underlying path was replaced by a circuit, generators were extended to weighted generators, and minimum cost generators were also treated for node-induced cost-functions. The original proof method however was not constructive and the only known algorithm so far to solve the optimization problems in question relied on the ellipsoid method. In this paper a constructive proof is described which gives rise to a purely combinatorial strongly polynomial algorithm.

## 1. INTRODUCTION

Let $P = (v_0, j_1, v_1, j_2, v_2, \ldots, j_n, v_n = v_0)$ be a directed circuit, that is, each directed edge $j_i$ has tail $v_{i-1}$ and head $v_i$, and the nodes $v_1, \ldots, v_n$ are distinct. Let $V := \{v_0, \ldots, v_n\}$ and $E := \{j_1, \ldots, j_n\}$ denote the node set and edge set of $P$, respectively. Let $E^* := \{uv : u, v \in V, u \neq v\}$ denote the set of all directed edges on the node set $V$. Suppose that a non-negative integer-valued weight-function $p : E \to \mathbf{Z}_+$ is given.

Let $\mathcal{P}$ be a system of distinct subpaths of $P$. We use the convention that the edge set of $P_i$ will be denoted by the same letter $P_i$. The node set of $P_i$ is denoted by $V(P_i)$. For a path $J$, let $f(J)$ and $l(J)$ denote the first and last nodes of $J$, respectively. Also, for a (directed) edge $e = uv$ let $f(e) := u$ and $l(e) := v$.

We say that a system $\mathcal{G}$ of subpaths of $P$ **generates** a path $J$ if $J$ is the union of some members of $\mathcal{G}$. $\mathcal{G}$ **generates** $\mathcal{P}$ or $\mathcal{G}$ is a **generator of** $\mathcal{P}$ if every member of $\mathcal{P}$ is generated by $\mathcal{G}$. Let $\gamma(\mathcal{P})$ denote the minimum cardinality of a generator of $\mathcal{P}$. More generally, a family $\mathcal{G}$ of (not-necessarily distinct) subpaths of $P$ is a **$p$-generator** or a **weighted generator** of $\mathcal{P}$ if, for every path $J \in \mathcal{P}$ and edge $j \in J$, $\mathcal{G}$ contains at least $p(j)$ paths containing $j$ and included in $J$. Let $\gamma_p(\mathcal{F})$ denote the minimum cardinality of a $p$-generator of $\mathcal{P}$. Clearly, for $p \equiv 1$, a $p$-generator is a generator. One of our main concerns will be to find algorithmically a minimum weighted generator.

To formulate a min-max theorem for $\gamma_p$ we need the following terminology. Let $J \in \mathcal{P}$ be a path and $j$ an edge of $J$. We say that the pair $(J, j)$ is a **path-edge pair.** Let $\mathcal{R}$ denote the set of all path-edge pairs $(J, j)$ for which $j \in J \in \mathcal{P}$. A

1991 *Mathematics Subject Classification.* 05C40, 90B10, 90C35.

*Key words and phrases.* combinatorial algorithms, path systems.

Research supported by the Hungarian National Foundation for Scientific Research Grant, OTKA T17580.

member $(J,j)$ of $\mathcal{R}$ is called **essential,** or an **essential pair** if there is no member $J'$ $(\neq J)$ of $\mathcal{P}$ for which $j \in J' \subset J$. Let $\mathcal{E}$ denote the set of essential pairs of $\mathcal{R}$. We will say that an essential pair $(J,j)$ is **sitting** at $j$. Weight $p(j)$ is called the **weight** of $(J,j)$ and denoted by $p(J,j)$. (Hopefully, no confusion may arise from this kind of notation that symbol $p$ is used for a weighting on $E$ and for a weighting on the set of path-edge pairs, as well.)

Let $(J,j)^-$ denote the set of nodes of $J$ preceding edge $j$, and let $(J,j)^+$ denote the set of nodes of $J$ following $j$. That is, $(J,j)^-$ and $(J,j)^+$ form a partition of the node-set $V(J)$ of $J$. For any edge $e = uv \in E^*$, $e$ **covers** $(J,j)$ if $u \in (J,j)^-$ and $v \in (J,j)^+$. For a non-negative integer vector $z : E^* \to \mathbf{Z}_+$, let $\varrho_z(J,j) := \sum(z(e) : e \text{ covers } (J,j))$. The **deficiency** of a path-edge pair $(J,j)$ is defined by $h_z(J,j) := \max(0, p(j) - \varrho_z(J,j))$. The **deficiency** of $z$ is $h(z) := \sum(h_z(J,j) : (J,j) \in \mathcal{E})$.

If the deficiency of $(J,j)$ is zero, $z$ is said to **$p$-cover** $(J,j)$. Otherwise $(J,j)$ is called **deficient.** We say that $z$ is a **$p$-covering** or a **weighted covering** of $\mathcal{R}$ if $\varrho_z(J,j) \geq p(j)$ holds for every path-edge pair $(J,j) \in \mathcal{R}$. The **size** of a $p$-covering $z$ of $\mathcal{R}$ is $z(E^*) := \sum(z(e) : e \in E^*)$.

There is a natural correspondence between $p$-generators of $\mathcal{P}$ and $p$-coverings of $\mathcal{R}$. Namely, from a $p$-covering $z$ we can obtain a $p$-generator by associating with each edge $uv \in E^*$ $z(e)$ copies of the subpath of $P$ from $u$ to $v$, and conversely, from a $p$-generator $\mathcal{G}$ we obtain a $p$-covering $z$ by letting $z(uv)$ be the number of paths in $\mathcal{G}$ from $u$ to $v$. By this correspondence, $z(E^*) = |\mathcal{G}|$ and hence $\gamma_p(\mathcal{P})$ is the minimum size of $p$-covers of the set $\mathcal{R}$ of path-edge pairs.

Two path-edge pairs $(I,i), (J,j)$ are called **independent** if $(I,i)^- \cap (J,j)^- = \emptyset$ or $(I,i)^+ \cap (J,j)^+ = \emptyset$ which is equivalent to saying that no edge in $E^*$ can cover both of them. A set of represented paths is called **independent** if its members are pairwise independent. Let $\sigma_p(\mathcal{P})$ denote the maximum total weight of an independent subset of $\mathcal{R}$. For $p \equiv 1$, $\sigma_p$ is abbreviated by $\sigma$. Because two independent path-edge pairs cannot be covered by one edge, $z(E^*) \geq \sigma_p(\mathcal{P})$ holds for every $p$-covering $z$ of $\mathcal{R}$, or equivalently, every $p$-generator of $\mathcal{P}$ has at least $\sigma_p(\mathcal{P})$ members, and hence $\sigma_p(\mathcal{P}) \leq \gamma_p(\mathcal{P})$.

Actually, here we always have equality:

**Theorem 1.1.** *For a set $\mathcal{P}$ of subpaths of a directed circuit $(V,E)$ and for a weight function $p : E \to \mathbf{Z}_+$,*

$$\sigma_p(\mathcal{P}) = \gamma_p(\mathcal{P}),$$

*that is, the maximum total weight of an independent subset of $\mathcal{R}$ is equal to the minimum size of a $p$-covering of $\mathcal{R}$.*

This result was proved by Frank and Jordán [4] in a more abstract form concerning coverings of bi-supermodular functions. The special case of Theorem 1.1 when no member of $\mathcal{P}$ contains edge $j_n$ of $P$, that is, when $\mathcal{P}$ may be considered as a set of subpaths of an underlying directed path $P - j_n$ was proved by A. Lubiw [10]. When $p \equiv 1$, Lubiw's theorem specializes to a theorem of Győri [8] which has actually been the starting point of this area.

Győri's original proof for his theorem is not constructive but its ideas could be used to develop a combinatorial algorithm. That was achieved by D.S. Franzblau and D.J. Kleitman [7]. For a clear explanation and simplification of their algorithm, along with a computer code, see [9]. The algorithm of Franzblau and Kleitman was extended by A. Lubiw [10] to the weighted case. A completely different approach,

relying on Dilworth's theorem, was introduced in [5] to obtain a short algorithmic proof of Győri's theorem.

The proof of Theorem 1.1 in [4] is not constructive. Although the same paper describes a polynomial time algorithm to compute the two extrema in question, that algorithm relies on the theorem itself and uses the ellipsoid method. The main purpose of the present work is to extend ideas of [5] to develop an alternative proof of Theorem 1.1 which is constructive and gives rise to a strongly polynomial (purely combinatorial) algorithm to compute $\sigma_p(\mathcal{P})$ and $\gamma_p(\mathcal{P})$. A slight modification of the algorithm will allow us to solve a min-cost version of the weighted generator problem, as well.

The main min-max theorem of [4] on covering bi-supermudular functions by a minimum number of directed edges (whose special case is Theorem 1.1) remains algorithmically unsolved (apart from an ellipsoid-method based algorithm). In particular, no combinatorial polynomial time (exact) algorithm is known for the other important special case of the general model when the objective is adding a minimum number of new edges to make a given directed graph $k$-connected. However, for the even more special problem when the starting digraph is $(k-1)$-connected, that is, when the connectivity of the digraph is required to be increased by one by adding a minimum number of new edges, it was possible to develop such an algorithm [6].

Throughout we use the notational convention for any given function $g : S \to \mathbf{R}$ and subset $X \subseteq S$ to denote $\sum(g(x) : x \in X)$ by $g(X)$.

## 2. PREPARATIONS

Let $(\mathcal{K}, \leq)$ be a partially ordered set and $p : \mathcal{K} \to \mathbf{Z}_+$ a weight function. Two elements $K, L$ of $\mathcal{K}$ are called **comparable** if $K \leq L$ or $L \leq K$, otherwise they are **uncomparable.** By a **chain** (respectively, **antichain)** we mean a subset consisting of pairwise comparable (resp., uncomparable) elements. A family $\mathbf{C}$ of (not necessarily distinct) chains is said to **cover** $(\mathcal{K}, p)$ or that $\mathbf{C}$ is a **chain-covering** of $(\mathcal{K}, p)$ if every element $K \in \mathcal{K}$ belongs to at least $p(K)$ members of $\mathbf{C}$. The number $|\mathbf{C}|$ of chains is called the **size of the chain-covering** of $(\mathcal{K}, p)$. The **weight** $p(\mathcal{A})$ of a subset $\mathcal{A} \subseteq \mathcal{K}$ is $\sum(p(K) : K \in \mathcal{A})$. In the algorithm we are going to describe, a weighted version of Dilworth's theorem plays a fundamental role.

**Theorem 2.1** (Weighted Dilworth). *In a partial order $(\mathcal{K}, \leq)$ with a weight function $p$, the minimum size of a chain-covering of $(\mathcal{K}, p)$ is equal to the maximum weight of an antichain. Moreover, there is a set $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ of $k \leq |\mathcal{K}|^2$ distinct chains along with positive integer coefficients $\gamma_1, \ldots, \gamma_k$ so that the family consisting of $\gamma_i$ copies of every chain $\mathcal{C}_i$ $(i = 1, \ldots, k)$ is a minimum size chain-covering of $p$.* □

This form immediately follows from the original version of Dilworth's theorem (where $p \equiv 1$) by replacing each element $K$ by $p(K)$ pairwise uncomparable elements. Ford and Fulkerson [3] described an elegant way to deduce Dilworth's theorem from that of Kőnig. An analogous reduction can be used to formulate the weighted Dilworth problem as a maximum flow problem and therefore both a minimum chain-covering of $p$ and a maximum weight antichain can be computed in (strongly) polynomial time via a max-flow min-cut algorithm. We will refer to this algorithm as the **weighted Dilworth algorithm** (see the Appendix). Note that the second half of the theorem follows from this max-flow approach.

With the help of this (rather standard) technique even the following minimum cost version becomes algorithmically tractable. Suppose that, in addition to the weight function $p$, we are given two cost functions $d_f$ and $d_l$ on $\mathcal{K}$. The **cost** of a chain with minimal element $K$ and maximal element $L$ is defined by $d_f(K)+d_l(L)$. We will refer to the algorithm computing a minimum cost chain-covering of $p$ as the **min-cost weighted Dilworth algorithm.** (See the Appendix).

We will work with a particular partial ordered set $(\mathcal{E}, \preceq)$ which is defined, as follows. For two essential pairs $(I,i)$ and $(J,j)$ let $(I,i) \preceq (J,j)$ if $(I,i)^- \subseteq (J,j)^-$ and $(I,i)^+ \supseteq (J,j)^+$. If $(I,i) \preceq (J,j)$ but $(I,i) \neq (J,j)$, we write $(I,i) \prec (J,j)$. Two essential pairs are **crossing** if they are neither comparable nor independent. (That is, two uncomparable essential pairs are either independent or crossing). Note that the essential pairs sitting at the same edge form a chain. A subset $\mathcal{K}$ of $\mathcal{E}$ is called **cross-free** if it contains no two crossing members.

**Claim 2.2.** *If a function $z : E^* \to \mathbf{Z}_+$ $p$-covers $\mathcal{E}$, then $z$ $p$-covers $\mathcal{R}$, as well.*

*Proof.* It is enough to prove that if $(J,j)$ is a path-edge pair not $p$-covered by $z$ so that $|J|$ is minimum, then $(J,j)$ is essential. If there is a path $J' \in \mathcal{P}$ with $j \in J' \subset J$, then, by the minimality of $J$, $\varrho_z(J',j) \geq p(j)$. Since every edge covering $(J',j)$ covers $(J,j)$, as well, we obtain that $\varrho_z(J,j) \geq \varrho_z(J',j) \geq p(j)$, contradicting the assumption that $z$ does not $p$-cover $(J,j)$, that is, $(J,j)$ is indeed essential. □

The claim implies that in order to $p$-cover $\mathcal{R}$ it suffices to $p$-cover $\mathcal{E}$.

**Claim 2.3.** *The members of a chain $\mathcal{C}$ of $(\mathcal{E}, \preceq)$ can be covered by one edge. If $\mathbf{C}$ is a family of chains and $p'(K,k)$ denotes the number of chains in $\mathbf{C}$ containing $(K,k)$ ($(K,k) \in \mathcal{E}$), then there is an integer vector $z : E^* \to \mathbf{Z}_+$ so that $\varrho_z(K,k) \geq p'(K,k)$ for every $(K,k) \in \mathcal{E}$ and so that the size $z(E^*)$ of $z$ is $|\mathbf{C}|$.*

*Proof.* If $(K,k)$ and $(J,j)$ are the minimal and maximal members of $\mathcal{C}$, respectively, then any edge $uv$ with $u \in (K,k)^-$, $v \in (J,j)^+$ covers each member of the chain $\mathcal{C}$. The second half follows easily from the first. Namely, for each chain $\mathcal{C}$ in $\mathbf{C}$ choose an edge $e_\mathcal{C}$ covering the members of $\mathcal{C}$, define $z_\mathcal{C} : E^* \to \mathbf{Z}_+$ by $z_\mathcal{C}(e_\mathcal{C}) = 1$ and $z_\mathcal{C}(e) = 0$ otherwise, and finally define $z := \sum(z_\mathcal{C} : \mathcal{C} \in \mathbf{C})$. This function $z$ satisfies the requirements. □

The following claim is staightforward.

**Claim 2.4.** *Let $(K,k)$ and $(J,j)$ be two crossing essential pairs. Then both $(K,j)$ and $(J,k)$ are essential.* □

**Lemma 2.5.** *Let $(K,k)$ and $(J,j)$ be two crossing essential pairs. If an essential pair $(X,x)$ crosses neither of them, then $(X,x)$ crosses neither $(K,j)$ nor $(J,k)$.*

*Proof.* First assume that $(X,x)$ is comparable with both $(K,k)$ and $(J,j)$. It is not possible that $(K,k) \preceq (X,x) \preceq (J,j)$ or $(J,j) \preceq (X,x) \preceq (K,k)$ since then $(K,k)$ and $(J,j)$ would be comparable and not crossing. Therefore either $(X,x) \succeq (K,k),(J,j)$ or $(X,x) \preceq (K,k),(J,j)$. By symmetry we may assume that $(X,x) \succeq (K,k),(J,j)$. But in this case $(X,x) \succeq (K,j)$ and $(X,x) \succeq (J,k)$, that is, $(X,x)$ does not cross $(K,j)$ and $(J,k)$.

Second, assume that $(X,x)$ is independent from both $(K,k)$ and $(J,j)$. Then $x \notin K \cup J$ and hence $(X,x)$ is independent from both $(K,j)$ and $(J,k)$.

In the third case $(X,x)$ is independent from one of $(K,k)$ and $(J,j)$, say from $(K,k)$, and comparable with the other, $(J,j)$. Then $x \notin K$ and hence $(X,x)$ is

independent of $(K,j)$. If $(X,x) \preceq (J,j)$, then $(X,x) \preceq (J,k)$. If $(X,x) \succeq (J,j)$, then $(X,x) \succeq (J,k)$. □

## 3. ALGORITHMIC PROOF OF THEOREM 1.1

We have seen that $\sigma_p(\mathcal{P}) \leq \gamma_p(\mathcal{P})$. In order to prove equality, it suffices to find a $p$-covering $z$ of $\mathcal{R}$ and an independent subset $\mathcal{A}$ of $\mathcal{E}$ for which $z(E^*) = p(\mathcal{A})$ $(:= \sum(p(A) : A \in \mathcal{A}))$. Below we describe a constructive proof of the existence of such $z$ and $\mathcal{A}$, and indicate how this proof can be turned into a strongly polynomial time algorithm.

**PHASE 1.** Compute a cross-free subset $\mathcal{K}$ of $\mathcal{E}$, as follows. At the beginning, let $\mathcal{K}$ be empty. Choose a total ordering of the elements of $\mathcal{E}$ so that $(K,k)$ precedes $(J,j)$ whenever $p(K,k) > p(J,j)$. Consider the elements of $\mathcal{E}$ in this order and put one into $\mathcal{K}$ if it does not cross any previously chosen member of $\mathcal{K}$.

Let $(K_1,k_1),(K_2,k_2),\ldots,(K_N,k_N)$ denote the members of the final $\mathcal{K}$ where the indices reflect the order in which the members have been put into $\mathcal{K}$. For each $(J,j) \in \mathcal{E} - \mathcal{K}$ there is a member of $\mathcal{K}$ crossing $(J,j)$, and let $t(J,j)$ denote the least index $i$ for which $(K_i,k_i) \in \mathcal{K}$ crosses $(J,j)$.

**PHASE 2.** With the help of the weighted Dilworth algorithm determine a maximum weight antichain $\mathcal{A}$ of $\mathcal{K}$ and a minimum cardinality chain-covering $\mathbf{C}$ of $(\mathcal{K},p)$ (for which $|\mathbf{C}| = p(\mathcal{A})$). By Claim 2.3, there is an integer-valued $p$-covering $z : E^* \to \mathbf{Z}_+$ of $\mathcal{K}$ with size $|\mathbf{C}|$. If $z$ $p$-covers $\mathcal{E}$ as well, then by Claim 2.2 we have found a required $p$-covering of $\mathcal{R}$ and the algorithm (as well as the proof of the theorem) terminates.

So suppose that there are deficient essential pairs. We are going to improve the current $p$-covering $z$ of $\mathcal{K}$ step by step without changing its size so as to obtain a final $p$-covering of $\mathcal{K}$ which $p$-covers the whole $\mathcal{E}_p$, as well, (and hence $\mathcal{R}_p$, too).

Let $(J,j)$ be a deficient essential pair which is **latest** in the sense that $t(J,j)$ is maximum. Let $(K,k) := (K_{t(J,j)}, k_{t(J,j)})$. By Claim 2.4, $(J,k)$ and $(K,j)$ are essential.

**Claim 3.1.** *Both $(J,k)$ and $(K,j)$ are $p$-covered by $z$.*

*Proof.* Suppose indirectly that $(J,k)$ is not covered by $z$. (The proof for $(K,j)$ is analogous). Since every member of $\mathcal{K}$ is $p$-covered, we have $(J,k) \notin \mathcal{K}$. Since $(J,j)$ is latest, $i' := t(J,k) \leq t(J,j)$. Here we cannot have equality because $(K,k)$ and $(J,k)$ are comparable and not crossing. Now $(K_{i'},k_{i'})$ crosses neither $(K,k)$ (since both are in $\mathcal{K}$) nor $(J,j)$ (by the definition of $t(J,j)$) but $(K_{i'},k_{i'})$ does cross $(J,k)$, contradicting Lemma 2.5. □

By reversing the edges, if necessary, (and thereby interchanging $(X,x)^-$ and $(X,x)^+$ for every $(X,x) \in \mathcal{R}$), we may assume that $f(k) \in (J,j)^-$ and $l(k) \in (J,j)^-$.

**Claim 3.2.** *There is an edge $e_1 = u_1v_1 \in E^*$ with $u_1 \in (J,j)^- \cap (K,k)^-$, $v_1 \in (J,j)^- \cap (K,k)^+$) so that $z(e_1) > 0$. There is an edge $e_2 = u_2v_2$ with $u_2 \in (K,k)^- - (J,j)^-, v_2 \in (J,j)^+ \cap (K,k)^+$ so that $z(e_2) > 0$.*

*Proof.* We claim that $p(k) \geq p(j)$ for otherwise $(J,j)$ would precede $(K,k)$ in the total ordering of $\mathcal{E}$ introduced in Phase 1, that is, during the construction of $\mathcal{K}$ pair $(J,j)$ would have been considered earlier than $(K,k)$ and then, by the definition of $i = t(J,j)$, $(J,j)$ would have been chosen to belong to $\mathcal{K}$.

By $p(k) \geq p(j)$ we have $p(J,k) = p(K,k) \geq p(J,j)$. Since $z$ does not $p$-cover $(J,j)$ but, by Claim 3.1, it $p$-covers $(J,k)$, there must be an edge $e_1$ required by the Claim. Similarly, since $z$ $p$-covers $(K,j)$, an edge $e_2$ required by the Claim must exist. □

*EXTREME CHOICE OF* $e_1$ *AND* $e_2$. Choose edges $e_1$ and $e_2$ ensured by the previous claim so that $e_1$ is minimal and $e_2$ is maximal in the sense that the subpath of $P$ from $u_1$ to $v_1$ is as small as possible and the subpath of $P$ from $u_2$ to $v_2$ is as large as possible. (This particular choice of $e_1$ and $e_2$ will not be used in the proof of the theorem. It will play a role in getting a strongly polynomial upper bound for the number of steps of the algorithm.)

*FLIPPING OPERATION.* Let $\delta := \min(z(e_1), z(e_2))$ and let $e_1' := u_1v_2, e_2' := u_2v_1$. Revise $z$ by letting $z'(e_1) := z(e_1) - \delta, z'(e_2) := z(e_2) - \delta, z'(e_1') := z(e_1') + \delta, z'(e_2') := z(e_2) + \delta$, and let $z'(e) := z(e)$ otherwise. Note that $z'(E^*) = z(E^*)$, that is, the size of $z'$ and $z$ is the same. We call this revision of $z$ a **flipping operation** or, in short, a **flipping** at $e_1$ and $e_2$.

*FIXING A LATEST DEFICIENT PAIR* $(J,j)$. As long as $(J,j)$ is deficient, choose edges $e_1$ and $e_2$ as described in *extreme choice*, and apply a flipping operation at $e_1$ and $e_2$.

**PHASE 3.** As long as there are deficient pairs, choose a latest deficient pair $(J,j)$ and fix it.

In order to conclude the proof of the theorem, it suffices to show that the deficiency of $z'$ obtained by applying one flipping is smaller than that of $z$, that is, $h(z') < h(z)$. (We will prove this by assuming only the properties of $e_1$ and $e_2$ given in Claim 3.2 and not using the extreme choice of $e_1, e_2$.)

**Claim 3.3.** *$\varrho_{z'}(X,x) \geq \varrho_z(X,x)$ for every essential pair $(X,x)$ and $\varrho_{z'}(J,j) > \varrho_z(J,j)$.*

*Proof.* As the second part is straightforward, we prove only the first. Obviously, if $e_2$ covers $(X,x)$, then $e_1'$ or $e_2'$ covers $(X,x)$. It is also trivially seen that if both $e_1$ and $e_2$ cover $(X,x)$, then so do $e_1'$ and $e_2'$. Finally, if $(X,x)$ is covered by $e_1$, then $(X,x)$ must be covered by at least one of $e_1'$ and $e_2'$ for otherwise we have $k \in X \subset K$ contradicting that $(K,k)$ is essential. □

By this claim $z'$ is also a $p$-covering of $\mathcal{K}$ for which $h(z') < h(z)$. Therefore, after a finite number of flippings, a vector $z_0$ is obtained with $h(z_0) = 0$, that is, $z_0$ is a $p$-covering of $\mathcal{E}$ and its size is equal to the total weight of an independent set $\mathcal{A}$ of essential pairs, as required for the theorem. □

By now we have finished the description of the steps of the algorithm and proved its finiteness. Our next goal is to get a polynomial bound on the number of steps. To this end first we show that Phase 3 terminates after at most $n^4$ flippings.

**Claim 3.4.** *The precedure of fixing a latest deficient pair $(J,j)$ terminates after at most $n(n-1)$ flippings.*

*Proof.* When a flipping is carried out at $e_1$ and $e_2$, at least one of the $z$-values $z(e_1)$ and $z(e_2)$ becomes zero. By the extreme choice of $e_1$ and $e_2$, their $z$-values cannot increase during the subsequent flippings as long as $(J,j)$ is deficient. Therefore after at most $|E^*| = n(n-1)$ flippings $(J,j)$ must become $p$-covered. □

**Claim 3.5.** $|\mathcal{E}| \leq n^2$.

*Proof.* Let $\mathcal{P}_i$ denote the set of paths in $P$ with first node $v_i$. For every edge $e \in P$ there is at most one member $X$ of $\mathcal{P}_i$ for which $e \in X$ and $(X, e)$ is essential. That is, there are at most $n$ essential pairs $(X, e)$ with $f(X) = v_i$. Hence $|\mathcal{E}| \leq n^2$. □

Combining Claims 3.4 and 3.5 we obtain that after at most $n^4$ flipping operations Phase 3 terminates with a $p$-covering $z_0$ of $\mathcal{R}$ for which $z_0(E^*) = p(\mathcal{A})$.

A. Benczúr, J. Förster, and Z. Király [2] proved that any cross-free set of essential pairs, and hence $\mathcal{K}$, can have at most $n \log n$ members. Since there are max-flow min-cut algorithms of complexity $O(p^3)$, where $p$ is the number of nodes of the network, Phase 2 will terminate in $O(nm + n^2\sqrt{n \log n})$ steps. Note that if one uses only the naive upper bound $|\mathcal{K}| \leq |\mathcal{E}| \leq n^2$, then the number of steps in Phase 2 could be estimated only by $O(n^6)$.

Since $|\mathcal{E}| \leq n^2$, Phase 1 can be carried out in $O(n^4)$ steps, we can conclude that the complexity of the whole algorithm is $O(n^4)$. Benczúr et al. [2] made further simplifications for the unweighted case ($p \equiv 1$) and obtained an algorithm of complexity $O(n^{2.5} \log^{3/2} n)$.

## 4. MINIMUM COST WEIGHTED $p$-COVERINGS

Our next goal is to show how a slight modification of the algorithm above helps us compute a minimum cost $p$-covering of $\mathcal{R}$ for a node-induced cost function. The only change in the whole algorithm occurs in Phase 2 where finding a minimum cost chain-covering of $(\mathcal{K}, p)$ will be demanded rather than a minimum size.

Suppose we are given two cost-functions $d_f : V \to \mathbf{R}_+$ and $d_l : V \to \mathbf{R}_+$. The cost $d(e)$ of an edge $e = uv \in E^*$ is defined by $d(e) := d_f(u) + d_l(v)$. (This is why we call such a cost-function node-induced). For a function $z : E^* \to \mathbf{Z}_+$ let the cost of $z$ be defined by $d(z) := \sum(z(uv)d(uv) : uv \in E^*)$. (When $d_f \equiv 1, d_l \equiv 0$, the cost $d(z)$ is the size of $z$.) Extend the domain of $d_f$ and $d_l$ to the set of path-edge pairs, as follows. Let $d_f(K, k) := \min(d_f(u) : u \in (K, k)^-)$ and $d_l(K, k) := \min(d_l(u) : u \in (K, k)^+)$. Note that $d_f$ is **monotonously decreasing** and $d_l$ is **monotonously increasing** in the sense that if $(J, j) \preceq (K, k)$, then $d_f(K, k) \leq d_f(J, j)$ and $d_l(K, k) \geq d_f(J, j)$. Define the cost of a chain $\mathcal{C}$ by $d(\mathcal{C}) := d_f(K, k) + d_l(L, l)$ where $(K, k)$ and $(L, l)$ are the minimal and maximal elements of $\mathcal{C}$, respectively.

Let $\mathcal{K}$ be the cross-free family determined by Phase 1. It follows from the definitions that for any edge $e \in E^*$ the members of $\mathcal{K}$ covered by $e$ form a chain of cost at most $d(e)$, and conversely (similarly to the proof of Claim 2.4) any chain $\mathcal{C}$ of $\mathcal{K}$ can be covered by an edge $e$ of cost $d(C)$. Therefore a minimum cost $p$-covering of $\mathcal{K}$ can be determined by computing a minimum cost chain-covering of $(\mathcal{K}, p)$ which, in turn, can be done in Phase 2 with the help of the min-cost weighted Dilworth algorithm (to be outlined in the Appendix). Hence we can find a minimum cost integer-valued $p$-covering $z$ of $\mathcal{K}$.

Finally, apply Phase 3. The correctness of the algorithm (that is, that the final $p$-covering of $\mathcal{E}$ is of minimum cost) follows from the easy observation that if $z'$ denotes the function obtained from $z$ by a flipping operation, then $d(z') = d(z)$, that is, the cost of $z$ and $z'$ are the same. (This is the place where we exploit that $d$ is node-induced and not a general cost-function.)

We remark that the above cost function can be made slightly more general by allowing a specified subset $F \subset E^*$ of edges to be used freely, that is, with cost

zero. Everything above (proof and algorithm) remains unchanged if one works with the subset of represented pairs not covered by any element of $F$ (including that $\mathcal{E}$ and $\mathcal{K}$ are also restricted in this way.)

## APPENDIX: (MIN-COST) WEIGHTED DILWORTH ALGORITHM

Ford and Fulkerson [3] showed how to derive Dilworth's theorem from Kőnig's theorem on maximum matchings in bipartite graphs. Since a maximum matching can be efficiently computed, this reduction made it possible to compute in polynomial time a minimum chain decomposition of a partially ordered set along with a maximum antichain. In Section 2 we have formulated the weighted Dilworth theorem. Here we show how the reduction technique of Ford and Fulkerson extends to the weighted case. For the closely related problem of finding a *minimum* flow in a network satisfying lower bound requirements on the arc-flows, see [1].

Let $(\mathcal{K}, \leq)$ be a partially ordered set endowed with an integer-valued weight-function $p : \mathcal{K} \to \mathbf{Z}$. With $(\mathcal{K}, \leq)$ we associate a directed graph $D := (U, F)$ with edge-capacities, as follows. For each element $K \in \mathcal{K}$ let $k'$ and $k''$ be two corresponding elements. Let $s$ be a source node and $t$ a sink node and let $U := \{s, t\} \cup \{k' : K \in \mathcal{K}\} \cup \{k'' : K \in \mathcal{K}\}$. For every element $K \in \mathcal{K}$, let $sk'$ and $k''t$ belong to $F$ with capacities $g(sk') := g(k''t) := p(K)$. Furthermore, let $k'l''$ belong to $F$ if $K > L$ and define $g(k'l'') = \infty$. (Note that $k'k''$ does not belong to $F$.)

By a max-flow min-cut algorithm we can compute a maximum integer-valued feasible flow $x$ and a set $S$ for which $s \in S \subseteq U - t$ and $\delta_g(S)$ is minimum where $\delta_g(X) := \sum(g(f) : f \in F \text{ leaves } S)$. By the MFMC theorem the value of maximum flow $x$ (that is, by definition, $\delta_x(s)$) is equal to $\delta_g(S)$.

Define another directed graph $D' = (U, F')$, as follows. For every element $K \in \mathcal{K}$, let $sk'$, $k''t$, $k'k''$ belong to $F'$, and let $l''k'$ belong to $F'$ if $K > L$. Let us define $x' : F' \to \mathbf{Z}_+$, as follows. For $K \in \mathcal{K}$, let $x'(sk') := p(K) - x(sk'), x'(k''t) := p(K) - x(k''t)$. Let $x'(k'k'') := p(K)$, and in case $K > L$ let $x'(l''k') := x(k'l'')$. It is not difficult to see that $x'$ is a non-negative integer-valued flow in $D'$ whose value is $p(\mathcal{K}) - \delta_x(s)$.

It is well-known (and can be proved easily by induction) that in an acyclic digraph with $n'$ nodes and $m'$ edges, any flow can be decomposed into at most $m' - n' + 2$ path-flows. (A **path-flow** is a flow that is positive on the edges of a directed path from $s$ to $t$ and zero otherwise). Moreover the inner nodes of any directed path of $D'$ from $s$ to $t$ form a chain of $(\mathcal{K}, \leq)$. Let $N := |\mathcal{K}|$. Since $D'$ has $2N+2$ nodes and at most $3N + N(N-1)/2$ edges, we obtain that $x'$ can be decomposed into at most $3N + N(N-1)/2 - (2N+2) + 2 = N(N+1)/2 \leq N^2$ path-flows. Hence there is a set of chains $\mathcal{C}_1, \ldots, \mathcal{C}_k$ $(k \leq N^2)$ of $(\mathcal{K}, \leq)$ and positive integer-valued coefficients $\gamma_1, \ldots, \gamma_k$ so that the family $\mathbf{C}$ of chains consisting of $\gamma_i$ copies of $\mathcal{C}_i$ $(i = 1, .., k)$ is a chain-covering of $(\mathcal{K}, p)$ and $\sum \gamma_i = \delta_{x'}(s) = p(\mathcal{K}) - \delta_x(s) = p(\mathcal{K}) - \delta_g(S)$.

Let us now consider the minimum cut in $D$ determined by $S$.

**Claim A.1.** *If $k'' \in S$, then $k' \in S$.*

*Proof.* If $k'' \in S$ and $k' \notin S$, then there must be an element $L_1 \in \mathcal{K}$ for which $l_1'' \notin S$ and $K > L_1$ for otherwise we would have $\delta_g(S + k') < \delta_g(S)$, contradicting the choice of $S$. Similarly, there must be an element $L_2 \in \mathcal{K}$ for which $l_2' \in S$ and $K < L_2$, for otherwise we would have $\delta_g(S - k'') < \delta_g(S)$. But now $L_2 > L_1$ and the edge $l_2' l_1''$ leaves $S$ and hence $\delta_g(S) = \infty$, contradicting the minimality of $\delta_g(S)$. □

Let $\mathcal{A} := \{K \in \mathcal{K} : k' \in S, k'' \notin S\}$. Then $\mathcal{A}$ is an antichain of $(\mathcal{K}, \leq)$ and $p(\mathcal{A}) = p(\mathcal{K}) - \delta_g(S)$. Therefore $p(\mathcal{A})$ is equal to the flow-value of $x'$, that is, $\mathcal{A}$ is a maximum $p$-weight antichain of $(\mathcal{K}, \leq)$ and $\mathbf{C}$ is a minimum chain-covering of $(\mathcal{K}, p)$.

In conclusion, both a minimum chain-covering $\mathbf{C}$ of $(\mathcal{K}, p)$ and an antichain $\mathcal{A}$ of maximum weight can be computed with the help of a max-flow min-cut algorithm.

We remark that the minimum chain-covering constructed this way contains *exactly* $p(K)$ chains containing $K$ for every element $K$ of $\mathcal{K}$. We call such a chain-covering a **chain-decomposition** of $(\mathcal{K}, p)$. The above construction shows that there is a one-to-one correspondence between the chain-decompositions of $(\mathcal{K}, p)$ and the integer-valued flows of the network $(D, g)$ defined above.

Suppose now that we have two cost-functions $d_f$ and $d_l$ defined on $\mathcal{K}$ so that they are monotonously decreasing and increasing, respectively. Define the cost of a chain $\mathcal{C}$ by $d(\mathcal{C}) := d_f(K) + d_l(L)$ where $K$ and $L$ are the minimal and the maximal elements of $\mathcal{C}$, respectively. It follows from the monotonicity of $d_f$ and $d_l$ that the cost function $d$ is monotonously increasing in the sense that $d(\mathcal{C}_1) \geq d(\mathcal{C}_2)$ holds for every two chains with $\mathcal{C}_1 \supseteq \mathcal{C}_2$.

Our goal is to show how to modify the algorithm above to find a mininum cost chain-decomposition of $(\mathcal{K}, p)$. Use the same digraph $D = (U, F)$ introduced above along with the capacity function $g$ and define the cost of edges $sk'$, $k''t$ by $c(sk') := -d_l(K)$ and $c(k''t) := -d_f(K)$ $(K \in \mathcal{K})$ while the cost of all other edges is zero. By a min-cost flow algorithm (see e.g. [1]) determine a maximum flow $x$ of minimum cost. By the above-mentioned correspondence between flows and chain-decompositions of $(\mathcal{K}, p)$, one can easily see that the chain-decomposition $\mathbf{C}$ assigned to a minimum cost maximum flow of $D$ will be of minimum cost.

We can conclude that the minimum cost chain-decomposition problem can solved with the help of a min-cost flow algorithm. There are strongly polynomial combinatorial algorithms for the latter (for a comprehensive account, see Ahuja et al. [1]) and hence we have found one for the minimum cost weighted generator problem.

It should be remarked however that we need the min-cost flow problem only for a special type of cost function, namely when there are non-zero costs only on the edges leaving the source-node $s$ or entering the sink-node $t$. It can be shown that in such a case there is no need for a general purpose min-cost flow algorithm. Instead, a modification of a max-flow min-cut algorithm will do by taking into consideration the edges of $D$ at $s$ and $t$ one by one according to the increasing order of their costs.

## REFERENCES


1. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows*, Prentice Hall, 1993.
2. A. Benczúr, J. Förster, and Z. Király, *Finding minimum generators for path systems of a cycle - implementation and analysis*, submitted (1997).
3. L.R. Ford, D.R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, Princeton NJ., 1962.
4. A. Frank and T. Jordán, *Minimal edge-coverings of pairs of sets*, Combinatorial Theory **65** (1995), 73-110.
5. A. Frank, *Finding minimum generators of path systems*, J. Combinatorial Theory **Ser B.** (1997), to appear.
6. A. Frank, *Finding minimum edge-coverings of pairs of sets*, in preparation (1997).
7. D.S. Franzblau and D.J. Kleitman, *An algorithm for covering polygons with rectangles*, Information and Control **63** (1984), 164-189.
8. E. Győri, *A minimax theorem on intervals*, J. Combinatorial Theory **Ser. B 37** (1984), 1-9.
9. D. E. Knuth, *Irredundant Intervals*, ACM Journal of Experimental Algorithmics **1** (1996), article 1, 19.