COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **21** (1) (2001) 145–150

NOTE

UNCROSSING A FAMILY OF SET-PAIRS

TAMÁS FLEINER[1]

Received November 9, 1998
Revised October 18, 1999

Frank and Jordán [1] proved an important min-max result on covering a crossing family of set-pairs. As an application, among others they can solve the unweighted node-connectivity augmentation problem for directed graphs in polynomial time. In this paper, we show how to solve the dual packing problem in polynomial time. To decompose a fractional dual optimum as a convex combination of integer vertices, besides the ellipsoid method, we use a polynomial-time algorithm for uncrossing a family of set-pairs. Our main result is this uncrossing algorithm.

1. Introduction

Let A and B be fixed finite sets. By a *pair* we mean an ordered pair (T,H) of sets, such that $T\subseteq A$, $H\subseteq B$. In accordance with Frank and Jordán [1], for a pair (T,H) we call T the *tail* and H the *head* of the pair. By a *family of pairs* we mean a function $\mathcal{F}:\mathcal{P}(A)\times\mathcal{P}(B)\to\mathbb{Q}_+$ where $\mathbb{Q}_+:=\{x\in\mathbb{Q}:x\geq 0\}$. If $\mathcal{F}(T,H)>0$ then we say that $(T,H)\in\mathcal{F}$, and the *multiplicity* of this pair is $\mathcal{F}(T,H)$.

Our task is to determine from a given input $\mathcal{F}$, another family of pairs $\mathcal{F}^*$ which can be obtained from $\mathcal{F}$ by a sequence of elementary uncrossing steps. Family $\mathcal{F}^*$ should be *cross-free*, that is, it cannot be changed any more by performing an elementary uncrossing step. In an *elementary uncrossing step*, we decrease $\mathcal{F}$ on crossing pairs $(T_1,H_1),(T_2,H_2)$ by ε and increase

[1] Supported by the Netherlands Organization for Scientific Research (NWO) and OTKA project 029772.

Mathematics Subject Classification (1991): 05C85, 05A05 it also by ε on pairs $(T_1\cap T_2, H_1\cup H_2)$ and $(T_1\cup T_2, H_1\cap H_2)$ where $\varepsilon := \min\{\mathcal{F}(T_1,H_1),\mathcal{F}(T_2,H_2)\}$. Pairs (T_1,H_1) and (T_2,H_2) are *crossing* if $T_1\cap T_2\neq\emptyset\neq H_1\cap H_2$ and the above two pairs $(T_1\cap T_2, H_1\cup H_2)$ and $(T_1\cup T_2, H_1\cap H_2)$ differ from the original pairs (T_1,H_1) and (T_2,H_2). Pairs (T_1,H_1) and (T_2,H_2) are are *half-disjoint* if $T_1\cap T_2=\emptyset$ or $H_1\cap H_2=\emptyset$. We call the elements of $A\times B$ *edges*, and we say that edge ab *covers* pair (T,H) if $a\in T$ and $b\in H$. For subsets A' of A and B' of B let $\mathcal{F}_{A'}^{B'}(T,H)=\mathcal{F}(T,H)$ if $A'\subset T$ and $B'\subset H$ else $\mathcal{F}_{A'}^{B'}(T,H)=0$; let $\mathcal{F}_{\overline{A'}}^{\overline{B'}}(T,H)=\mathcal{F}(T,H)$ if $A'\cap T=\emptyset=B'\cap H$ else $\mathcal{F}_{\overline{A'}}^{\overline{B'}}(T,H)=0$; and $\mathcal{F}_{\underline{A'}}^{\underline{B'}}(T,H)=\mathcal{F}(T,H)$ if $A'\cap T\neq\emptyset\neq B'\cap H$ else $\mathcal{F}_{\underline{A'}}^{\underline{B'}}(T,H)=0$.

When it does not cause ambiguity, we might use only elements without brackets to denote a set, so $\mathcal{F}_a^b$ simply stands for $\mathcal{F}_{\{a\}}^{\{b\}}$. On $\mathcal{P}(A)\times\mathcal{P}(B)$ we introduce a *partial order*: $(T_1,H_1)\preceq(T_2,H_2)$ if $T_1\supseteq T_2$ and $H_1\subseteq H_2$. We may suppose that $A=\{a_i: 1\leq i\leq n\}$, $B=\{b_i: 1\leq i\leq n\}$, and let $m=|\{(T,H):(T,H)\in\mathcal{F}\}|$. If $f,g:X\to\mathbb{R}$ are functions and $Y\subseteq X\ni x$ then $f(Y):=\sum_{y\in Y}f(y)$, $fg(x):=f(x)\cdot g(x)$ and $f\cdot g:=fg(X)$.

The motivation for solving the above particular uncrossing problem is to give a polynomial-time algorithm for the dual problem of covering a crossing family of set-pairs. To describe this problem let us suppose we have a *crossing bisupermodular* demand-function $p:\mathcal{P}(A)\times\mathcal{P}(B)\to\mathbb{N}$, i.e. if $(T_1,H_1),(T_2,H_2)$ are crossing pairs and $p(T_1,H_1)\cdot p(T_2,H_2)\neq 0$ then $p(T_1,H_1)+p(T_2,H_2)\leq p(T_1\cap T_2,H_1\cup H_2)+p(T_1\cup T_2,H_1\cap H_2)$. In this model we are instructed to choose some edges in such a way, that every pair is covered with at least as high multiplicity as its demand. That is, we are looking for a function $z:A\times B\to\mathbb{N}$ such that $z(T\times H)\geq p(T,H)$ for every $\emptyset\neq T\subseteq A$, $\emptyset\neq H\subseteq B$, or in other words, z *covers* p. We should also do it with the minimum number of edges, i.e. we would like to compute $\tau_p:=\min\{z(A\times B): z \text{ covers } p\}$.

In this model we might want to optimize the maximum demand of a subfamily $\mathcal{F}$ of $\mathcal{P}(A)\times\mathcal{P}(B)$ of pairwise half-disjoint pairs (a *half-disjoint subfamily*), i.e. we would like to compute $\nu_p:=\max\{p(\mathcal{F}):\mathcal{F}\subseteq\mathcal{P}(A)\times\mathcal{P}(B) \text{ is half-disjoint}\}$. Theorem 2.3 in [1] of Frank and Jordán states the following:

Theorem 1.1. (Frank and Jordán [1]) *If A,B are finite sets and $p:\mathcal{P}(A)\times\mathcal{P}(B)\to\mathbb{N}$ is a crossing bisupermodular function, then $\nu_p=\tau_p$.* ∎

For the sake of applications of Theorem 1.1 (e.g. the directed node-connectivity augmentation, or an alternative (non-combinatorial) method to construct a minimum cardinality generator of a path-system), it would be handy to have an efficient algorithm that provides both a half-disjoint family with maximum demand, and an optimal covering of p. The characteristic vector of a half-disjoint family of maximum demand is an optimum of the integer counterpart of linear program

$$(1) \qquad \max\{y \cdot p : y \geq 0,\ y(e) \leq 1 \text{ for every edge } e\},$$

where $y : \mathcal{P}(A) \times \mathcal{P}(B) \to \mathbb{R}$ and $y(e) = \sum\{y(T,H) : e \text{ covers } (T,H)\}$. This program is just the dual of the one that solves the covering problem via an integer optimum z:

$$(2)\ \min\{z(A \times B) : z \geq 0,\ z(T \times H) \geq p(T,H) \text{ for every pair } (T,H)\}\ .$$

Theorem 1.1 can be interpreted such that for a crossing bisupermodular function p, linear programs (1) and (2) have integer optima. The authors of [1] also indicate a possible way to find an integer optimum z^* of (2) in polynomial time. What they do is that they compute a fractional primal optimum z using the ellipsoid method and with the help of this optimum they reduce the problem to another one where the bisupermodular function p' is "small". In the small problem they can find "reducing" edges one by one, and this is sufficient for the efficient construction of some optimal covering z^*.

With our algorithm we can uncross the fractional dual optimum $\mathcal{F} := y$ of (1) found by the ellipsoid method. As the size of $\mathcal{F}$ is polynomial in the size of the original problem, by using our uncrossing algorithm, we can also obtain a cross-free optimum $y^* := \mathcal{F}^*$ in polynomial time. From y^*, in polynomial time, we are able to compute a maximum p-weight $\prec$-antichain of $\operatorname{supp}(y^*) := \{(T,H) : y^*(T,H) > 0\}$, using the well-known bipartite matching algorithm for Dilworth's theorem. This provides an optimum half-disjoint family. Indeed, y^* is a rational optimum solution of (1), as performing a single elementary uncrossing step on y results in another rational optimum. So there is a $c \in \mathbb{N}$ such that cy^* is integer. From cross-freeness, a $\prec$-antichain of $\operatorname{supp}(y^*)$ corresponds to a half-disjoint family, and by definition any $\prec$-chain of $\operatorname{supp}(y^*)$ has (cy^*)-weight at most c. By the dual version of Dilworth's theorem there are c $\prec$-antichains $A_1, \ldots, A_c$ such that each pair (T,H) is contained in at least $cy^*(T,H)$ of these $\prec$-antichains. Hence the maximum p-weight of a $\prec$-antichain of $\operatorname{supp}(y^*)$ is certainly not less than $\frac{1}{c} cy^* \cdot p = y \cdot p$, which is the maximum demand of a half-disjoint family. (Note, that from the dual version of Dilworth's theorem it already follows that the set of $\prec$-minima of $\operatorname{supp}(y*)$ corresponds to a half-disjoint family of maximum demand.)

As a main application of Theorem 1.1 of [1], Frank and Jordán obtained a min-max formula for the unweighted directed node-connectivity augmentation problem. More precisely, they proved that the minimum number of arcs one needs to augment a given directed graph to be k-node-connected equals the maximum total demand of a certain independent family. Here independent means that any edge can decrease the demand of at most one member of the family. By solving the above primal problem (2) for this case the authors can find an optimum augmentation. A dual optimum which can be constructed via our uncrossing algorithm provides a witnessing independent family.

2. The algorithm

In this section we describe our main result, the uncrossing algorithm for set-pairs. It is well known, that the uncrossing procedure is finite, as the nonnegative amount $\sum_{a\in A}\sum_{b\in B}\sum_{a\in T\subseteq A}\sum_{b\in H\subseteq B}\mathcal{F}(T,H)$ is decreasing at every step with at least the reciprocal of the common denominator of the $\mathcal{F}$-values.

We begin with examining a cross-free family $\mathcal{F}^*$. If two pairs of $\mathcal{F}^*$ are covered by the same edge, then they must be $\prec$-comparable, as otherwise an elementary uncrossing step along them would change the family. On the other hand, if for every edge ab, family $\mathcal{F}_a^b$ is a $\prec$-chain (that is, the family of the heads and the family of the tails of these pairs are chains of increasing sets, and they are paired in an opposite way), then no elementary uncrossing step can change $\mathcal{F}$ anymore.

The input of a *routine step* of our uncrossing algorithm is a family $\mathcal{F}$ and an edge ab. Its output is a family $\mathcal{F}[a,b]$, which is obtained from $\mathcal{F}$ by uncrossing its subfamily $\mathcal{F}_a^b$. This can be done efficiently, because all the pairs emerging during the uncrossing process are covered by ab, thus the uncrossing procedure on $\mathcal{F}_a^b$ must result in a $\prec$-chain. Moreover, for $a'\in A$ and $b'\in B$ an elementary uncrossing step does not change $\sum\mathcal{F}_{a,a'}^b :=$ $\sum\{\mathcal{F}_{a,a'}^b(T,H):T\subset A,H\subset B\}$ and $\sum\mathcal{F}_a^{b,b'} := \sum\{\mathcal{F}_a^{b,b'}(T,H):T\subset A,H\subset B\}$. So after calculating all values $\sum\mathcal{F}_{a,x}^b$ for $x\in A$ and all values $\sum\mathcal{F}_a^{b,x}$ for $x\in B$ we simply pair the level-sets in an opposite way with corresponding $\mathcal{F}[a,b]$-values.

We remark, that here we only proved that if $\mathcal{F}$ is rational, then the level-set pairing is a quick way of uncrossing the subfamily $\mathcal{F}_a^b$. Although, at this point it is not clear whether for *real* $\mathcal{F}$'s there exist a *finite* uncrossing procedure that turns $\mathcal{F}_a^b$ into the oppositely paired level-sets. Based on the algorithm described in [2] it is relatively easy to construct such a method, showing that our result can be extended to the real case.

Unfortunately, in a routine step, we may ruin the previously ordered structure of the actual family along another edge. The key idea is, that we can keep control of the damage done if we execute our routine steps in the lexicographic order of the edges. To prove the following observations, we assume that rather than using the above efficient level-set pairing, we execute each routine step as a sequence of elementary uncrossing steps.

Lemma. 2.1. *If $\mathcal{F}_a^b$ is cross-free then $(\mathcal{F}[a,b'])_a^b$ is cross-free.*

Proof. Family $(\mathcal{F}[a,b'])_a^{b,\overline{b'}} = \mathcal{F}_a^{b,\overline{b'}}$ is cross-free by assumption, family $(\mathcal{F}[a,b'])_a^{b,b'}$ is cross-free because of the routine step. In the beginning every pair of $\mathcal{F}_a^{b,\overline{b'}}$ was $\prec$-less than any pair of $\mathcal{F}_a^{b,b'}$, and this property is preserved by an elementary uncrossing step along edge ab'. ∎

For family $\mathcal{F}$ and element a_i of A let $\mathcal{F}[a_i]$ denote the system that we get after executing the routine steps along all edges incident with a_i (in order $a_ib_1, a_ib_2, \dots, a_ib_n$). We call the execution of these routine steps the *ith phase of our algorithm.* Let $A(i) := \{a_j : 1 \le j \le i\}$.

Lemma. 2.2. *If $\mathcal{F}_{\underline{A(i-1)}}$ is cross-free then $(\mathcal{F}[a_i])_{\underline{A(i)}}$ is cross-free.*

Proof. Lemma 2.1 implies that $(\mathcal{F}[a_i])_{a_i}^b$ is cross-free for any $b \in B$. We have to show that the same holds for $(\mathcal{F}[a_i])_a^b$ whenever $a \in A(i-1)$ and $b \in B$. By Lemma 2.1, both $(\mathcal{F}[a_i])_{a,a_i}^b$ and $(\mathcal{F}[a_i])_{a,\overline{a_i}}^b$ will be cross-free by themselves. We will show that during the ith phase for any intermediate family $\mathcal{F}'$, for any $a \in A(i-1)$ and any $b \in B$

(1) any element of ${\mathcal{F}'}_{a,a_i}^b$ is $\prec$-less than any element of ${\mathcal{F}'}_{a,\overline{a_i}}^b$.

By assumption, property (1) is true for $\mathcal{F}$.

If, indirectly, this is not the case at the end, then there is a first elementary uncrossing step that ruins property (1). Suppose that it is first violated for ab, after an elementary uncrossing step is taken along a_ib' ($b = b'$ is allowed), with pairs $(T_1, H_1), (T_2, H_2) \in {\mathcal{F}'}_{a_i}^{b'}$, for some intermediate family $\mathcal{F}'$. Clearly, exactly one of the two pairs (say (T_1, H_1)) belongs to ${\mathcal{F}'}_a^b$, that is, $a, a_i \in T_1$ and $b, b' \in H_1$. The other pair (T_2, H_2) must be covered by a_ib' and cannot be covered by ab. If $a \notin T_2$ and $b \notin H_2$, then property (1) remains true for $(\mathcal{F}'[a_i, b])_a^b$ as no new pair which is covered by ab can emerge. If $a \notin T_2$ but $b \in H_2$ then only $(T_1 \cup T_2, H_1 \cap H_2)$ is covered by ab, and as $(T_1, H_1) \prec (T_1 \cup T_2, H_1 \cap H_2)$, property (1) remains true after the uncrossing. The only nontrivial case is, when $a \in T_2$ and $b \notin H_2$.

From $(T_1, H_1) \in {\mathcal{F}'}_a^b$, property (1) of $\mathcal{F}'$ implies that ${\mathcal{F}'}_{a,\overline{a_i}}^b \le {\mathcal{F}'}_a^{b'}$, thus ${\mathcal{F}'}_{a,\overline{a_i}}^b \le {\mathcal{F}'}_{a,\overline{a_i}}^{b'}$. So if $(T, H) \in {\mathcal{F}'}_{a,\overline{a_i}}^b$ then $(T, H) \in {\mathcal{F}'}_{a,\overline{a_i}}^{b'}$. From property (1) of $\mathcal{F}'$ along ab' we get that $(T_1 \cup T_2, H_1 \cap H_2) \prec (T_1, H_1) \prec (T, H)$, justifying the heredity of property (1). ∎

Let $\mathcal{F}^{(0)} := \mathcal{F}$ and for $1 \le i \le n$ let $\mathcal{F}^{(i)} := \mathcal{F}^{(i-1)}[a_i]$.

Theorem 2.3. *The above described algorithm constructs a cross-free family* $\mathcal{F}^* := \mathcal{F}^{(n)}_{\underline{A_n}}$ *in time polynomial in* n *and* m.

Proof. By induction, using Lemma 2.2, $\mathcal{F}^{(i)}_{\underline{A_i}}$ is cross-free. To construct $\mathcal{F}^*$, we need n^2 routine steps. The time needed for a routine step is linear in the size of the actual family. As new pairs, emerging from a routine step form a $\prec$-chain, each routine step brings at most $2n$ new pairs into the family. Hence the size of the actual family is never more than $m + 2n^3$, thus the algorithm needs $O(mn^2 + n^5)$ time. ∎

Acknowledgment. I thank the anonymous referees for their suggestions.

References

[1] ANDRÁS FRANK, TIBOR JORDÁN: Minimal edge-coverings of pairs of sets, *Journal of Combinatorial Theory Series B*, **65** (1995), 73–110.

[2] C. A. J. HURKENS, L. LOVÁSZ, A. SCHRIJVER, É. TARDOS: How to tidy up your set-system? *Combinatorics* (Eger, 1987), Colloquia Mathematica Societatis János Bolyai, 52, North Holland, Amsterdam, 1988, 309–314.

Tamás Fleiner

CWI, Kruislaan 413,
1098 SJ, Amsterdam,
The Netherlands
tamas@cwi.nl

Typeset by TYPOTEX Ltd, Budapest
PRINTED IN HUNGARY
Akadémiai Nyomda, Martonvásár