Geometric Methods in Discrete Optimization

A. Schrijver
Department of Econometrics Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
and
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

Classically, there exists a strong connection between optimization and geometry. Often, a set of options ('feasible solutions') can be represented by vectors in euclidean space, and a search process for an optimal option ('option solution') can be seen as a trip in space. The geometric nature of optimization methods like the simplex method, the gradient method, the ellipsoid method, the cutting plane method, is suggested already by their names. Thus optimization illustrates once more that Descartes' idea of analytic geometry can be used in turn to study analytic problems geometrically.

The above being well-known for linear and nonlinear optimization, where the feasible solutions generally give a continuous, sometimes even convex, region in space, the purpose of this paper is to show the geometric character also of several methods and results in combinatorial optimization, where the feasible solutions, in the first instance, yield a discrete, discontinuous set.

Among the geometric methods and results used in combinatorial optimization we discuss are:

- the representation of combinatorial optimization problems by polyhedra;
- the ellipsoid method;
- the basis reduction method for lattices;
- the cutting plane method;
- the results of Tutte and Seymour on the representation and decomposition of geometric configurations in projective spaces over GF(2).

We illustrate these ideas by some applications - we focus on two problems (the matching problem and the coclique problem), but the methods have a much wider applicability (like to trees (directed · or undirected), one- and multicommodity flows, coverings, directed cuts, cliques, disjoint paths in graphs, the traveling salesman problem, the acyclic subgraph problem).

2. Representing combinatorial optimization problems by polyhedra

The idea of using polyhedra in combinatorial optimization is simple. Suppose we have a collection $\mathcal{F}$ of subsets of a finite set S. (For instance, $\mathcal{F}$ is the collection of matchings in a given graph $G=(V,E)$.) Moreover, a function $c:s\rightarrow\mathbb{Z}$ is given, and we wish to find

$$\max_{U\in\mathcal{F}}\sum_{s\in U}c(s), \tag{1}$$

being a generic form of a combinatorial optimization problem. (In the example above, it amounts to finding a matching of maximum 'weight'.) Usually, the collection $\mathcal{F}$ is too large to evaluate every U in $\mathcal{F}$ to determine the maximum. 'Too large' here means with respect to the data structure given (like the number of matchings in a graph is exponentially large in the size of the graph). One should find a method more efficient than this 'brute force' method.

We can represent each subset U of S by its characteristic vector χ^U in $\{0,1\}^S$, i.e., $(\chi^U)_s=1$ if $s\in U$, and 0 otherwise. Moreover, the function c can be considered as a vector in $\mathbb{R}^S$. Then problem (1) becomes:

$$\max\{c^T\chi^U|U\in\mathcal{F}\}. \tag{2}$$

Clearly, the maximum value in (2) is equal to

$$\max\{c^Tx|x\in\text{conv.hull }\{\chi^U|U\in\mathcal{F}\}\}, \tag{3}$$

where conv.hull denotes the convex hull in $\mathbb{R}^S$. Since conv.hull $\{\chi^U|U\in\mathcal{F}\}$ is a convex polytope, there exists a matrix A and a column vector b such that this polytope is equal to $\{x|Ax\leqslant b\}$ (where the columns of A are indexed by the elements of S). This implies that (3) is equal to

$$\max\{c^Tx|Ax\leqslant b\}. \tag{4}$$

This way we have transformed the combinatorial optimization problem (1) into a linear programming problem, and we can appeal to linear programming methods to solve the combinatorial problem. We could use the simplex method to solve (4) and hence (1) (note that the simplex method gives a vertex of $\{x|Ax\leqslant b\}$ as optimal solution, which corresponds to the optimal set in $\mathcal{F}$). Alternatively, one could apply the ellipsoid method for linear programming, which is not a practical method, but which can yield that (1) is solvable in polynomial time.

The mathematical problem now is to determine A and b, given $\mathcal{F}$. Although the system $Ax\leqslant b$ clearly always exists, there is the problem that in many cases the polytope conv.hull $\{\chi^U|U\in\mathcal{F}\}$ has an enormous number of facets, often too difficult to describe. The application of linear programming methods will be helpful only in case the system $Ax\leqslant b$ is decent enough - decent in the sense to be described in Section 3 below.

As we shall see also in Section 3, if we are interested in the polynomial-time solvability of combinatorial optimization problems of type (1) (and in fact, we are), the above approach of replacing $\mathfrak{F}$ by conv.hull $\{\chi^U | U \in \mathfrak{F}\}$ is, at least implicitly, unavoidable.

As a theoretical by-product, if we have written (1) as the LP-problem (4), we can apply the Duality theorem of linear programming to (4), saying:

$$\max\{c^T x | Ax \leqslant b\} = \min\{y^T b | y \geqslant 0, y^T A = c^T\}. \tag{5}$$

Therefore,

$$\max_{U \in \mathfrak{F}} \sum_{s \in U} c(s) = \min\{y^T b | y \geqslant 0, y^T A = c^T\}, \tag{6}$$

which is a *min-max relation* for the combinatorial problem. If we can prove that the minimum in (6) has an integer solution, we obtain a purely combinatorial min-max relation.

3. Application of the ellipsoid method

The ellipsoid method was shown by Khachiyan [15] to solve linear programming problems in polynomial time. In this section we discuss an application of the ellipsoid method to combinatorial optimization.

An *(undirected) graph* is a pair $G = (V,E)$, where V is a finite set and E is a collection of unordered pairs from V. The elements of V and E are called *vertices* and *edges,* respectively.

Suppose we are given, for each graph $G = (V,E)$, a collection $\mathfrak{F}_G$ of subsets of E. For example:

(i) $\mathfrak{F}_G$ is the collection of matchings in G (a *matching* is a collection of pairwise disjoint edges); (7)

(ii) $\mathfrak{F}_G$ is the collection of trees in G (a *tree* is a connected set of edges not containing a circuit);

(iii) $\mathfrak{F}_G$ is the collection of Hamilton circuits in G (a *Hamilton circuit* is a circuit containing each vertex of G exactly once).

With the family $(\mathfrak{F}_G | G$ graph) we can associate the following problem:

Optimization problem: Given a graph $G = (V,E)$ and $c \in \mathbb{Q}^E$, find $E' \in \mathfrak{F}_G$ maximizing $\Sigma_{e \in E'} c(e)$. (8)

So if $(\mathfrak{F}_G | G$ graph) is as in (i), (ii) and (iii), problem (8) amounts to the problems of finding a maximum weighted matching, a maximum weighted tree, and a maximum weighted Hamilton circuit, respectively. The last problem is the well-known traveling salesman problem (note that by replacing c by $-c$, (8) becomes a minimization problem).

Clearly, for each collection $(\mathfrak{F}_G | G$ graph), problem (8) forms a class of problems of type (1).

Especially, we are interested for which families $(\mathfrak{F}_G | G$ graph) problem (8) is solvable in *polynomial time* (or *polynomially solvable*), i.e., solvable by an algorithm whose running time is bounded by a polynomial in the input size, which is

$$|V| + |E| + \text{size}(c).$$

Here $\text{size}(c) := \Sigma_{e \in E}\ \text{size}(c(e))$, where the size of a rational number p/q $\log_2(|p|+1)+\log_2(|q|)$. So size (c) is about the space needed to specify c binary notation.

If $(\mathcal{F}_G|G$ graph) is as in (7) (i) or (ii), problem (8) is polynomially solvable it is as in (iii) no polynomial-time algorithm has been found, and it is a gen belief that no such algorithm exists (see also the Remark below).

It has been shown by GRÖTSCHEL, LOVÁSZ and SCHRIJVER [11] that, for fixed family $(\mathcal{F}_G|G$ graph), (8) is polynomially solvable, if and only if the lowing problem is solvable in polynomial time:

> *Separation problem*: Given a graph $G=(V,E)$ and $x \in \mathbb{Q}^E$, determine if x belongs to conv.hull $\{\chi^F|F \in \mathcal{F}_G\}$, and if not, find a separating hyperplane. (

Again 'polynomial-time' means: with running time bounded by a polynon in $|V|+|E|+\Sigma_{e \in E}\ \text{size}(x(e))$.

THEOREM 1. *For any collection $(\mathcal{F}_G|G$ graph), the optimization problem* (8) *polynomially solvable, if and only if the separation problem* (10) *is polynomi solvable.*

The theorem implies that with respect to the question of polynomial-time s vability, the approach described in Section 2 (studying the convex hull) is m or less essential: a combinatorial optimization problem is polynomially sc able if and only if the corresponding convex hull can be decently described the sense of the polynomial solvability of the separation problem. This also be used in the negative: if a combinatorial optimization problem is polynomially solvable (maybe the traveling salesman problem), then corresponding polytopes have no decent description.

Theorem 1 is proved with the help of the ellipsoid method, for which refer to the books by GRÖTSCHEL, LOVÁSZ and SCHRIJVER [12] and SCHRIJ [22]. The ellipsoid method does not give practical algorithms, but in some ca with Theorem 1 the polynomial solvability of a combinatorial optimizat problem was proved, which next formed a motivation for finding a practi polynomial-time algorithm for the problem.

There are several variations of Theorem 1. For instance, a similar res holds if we consider collections $\mathcal{F}_G$ of subsets of the vertex set V, instead subsets of the edge set E. E.g., we could take:

> $\mathcal{F}_G$ is the collection of all cocliques of G (a *coclique* is a set of vertices which are pairwise not adjacent). (

Moreover, a similar theorem holds if we consider classes $(\mathcal{F}_G|G \in \mathcal{G})$, where $\mathcal{G}$ a subcollection of the set of all graphs. Similarly, we can consider *direc* graphs.

REMARK. The question $NP = P$? amounts to the following. Call a class $(\mathcal{F}_G | G$ graph) *polynomially recognizable* if there exists a polynomial-time algorithm for the following problem:

$$\text{given } G = (V,E) \text{ and } F \subseteq E, \text{ decide if } F \text{ belongs to } \mathcal{F}_G. \tag{12}$$

It is not difficult to see that each of the examples in (7) gives a polynomially recognizable class.

Now one has:

(13) $NP = P$, if and only if for each polynomially recognizable class $(\mathcal{F}_G | G$ graph) the optimization problem is polynomially solvable.

There seems no reason to believe that for every polynomially recognizable class the optimization problem is polynomially solvable. However, no counterexample has been found. It has been shown by COOK [4] and KARP [14] that the traveling salesman problem (and several other classical combinatorial optimization problems) is *NP-complete*. It implies: if the traveling salesman problem is polynomially solvable, then for every polynomially recognizable class the optimization problem is polynomially solvable. This is one of the reasons why a lot of research has been spent on the traveling salesman problem.

4. LATTICE BASIS REDUCTION, SIMULTANEOUS DIOPHANTINE APPROXIMATION AND STRONGLY POLYNOMIAL ALGORITHMS

The basis reduction method for lattices was given by LENSTRA, LENSTRA and LOVÁSZ [16]. It solves the following problem:

(14) given a nonsingular rational $n \times n$-matrix A, find a basis $b_1, \ldots, b_n$ for the lattice generated by the columns of A satisfying

$$\|b_1\| \cdot \ldots \cdot \|b_n\| \leq 2^{\frac{1}{4}n(n-1)} |\det A|,$$

in time bounded by a polynomial in $\text{size}(A) := \Sigma_{i,j} \text{size}(a_{ij})$. Here the *lattice generated by* $a_1, \ldots, a_n$ is the set of vectors $\lambda_1 a_1 + \ldots + \lambda_n a_n$ with $\lambda_1, \ldots, \lambda_n \in \mathbb{Z}$. Any linearly independent set of vectors generating the lattice is called a *basis* for the lattice.

The basis reduction method has several applications in linear and integer programming, in number theory and in cryptography. One consequence is a polynomial-time algorithm for *simultaneous diophantine approximation:*

THEOREM 2. *There exists a polynomial-time algorithm which for given vector $a \in \mathbb{Q}^n$ and rational ϵ with $0 < \epsilon < 1$, finds an integer vector p and an integer q satisfying*

$$\|a - \frac{1}{q}p\| < \frac{\epsilon}{q} \quad \text{and} \quad 1 \leq q \leq 2^{\frac{1}{4}n(n+1)} \epsilon^{-n}. \tag{15}$$

This can be seen by applying the basis reduction method to the matrix

$$A := \left[\begin{array}{cccc|c} 1 & & & 0 & \alpha_1 \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ 0 & & & 1 & \alpha_n \\ \hline 0 & \ldots\ldots & & 0 & 2^{-\frac{1}{4}n(n+1)}\epsilon^{n+1} \end{array}\right], \tag{16}$$

denoting $a =: (\alpha_1, \ldots, \alpha_n)^T$.

Frank and Tardos [6] showed that this simultaneous diophantine approximation method yields so-called strongly polynomial algorithms. The ellipsoid method discussed in Section 3 can derive a polynomial-time algorithm for the optimization problem (8) from a polynomial-time algorithm for the separation problem (10), and vice versa. The polynomial-time algorithms for (8) derived perform a number of arithmetic operations, which number is bounded by a polynomial in (9). (*Arithmetic operations* here are: addition, subtraction, multiplication, division and comparison of numbers.) Although this does not conflict the definition of 'polynomial-time', it would be preferable if the size of the 'cost' function c only influences the size of the numbers occurring when executing the algorithm, but not the number of arithmetic operations. Therefore, an algorithm for the optimization problem (8) is called *strongly polynomial* if it consists of a number of arithmetic operations, bounded by a polynomial in $|V|+|E|$, on numbers of size bounded by a polynomial in $|V|+|E|+\text{size}(c)$.

Frank and Tardos now showed however the equivalence of the two concepts when applied to (8):

Theorem 3. *For any class $(\mathcal{G}_G|G$ graph), there exists a polynomial-time algorithm for the optimization problem* (8), *if and only if there exists a strongly polynomial algorithm for* (8).

Proof. The 'if' part being trivial, we sketch a proof of the 'only if' part. Suppose (8) is polynomially solvable for a certain class $(\mathcal{G}_G|G$ graph). Let $G=(V,E)$ and $c \in \mathbb{Q}^E$ be given as input for (8). Determine vectors $c_1, c_2, \ldots$ successively as follows. $c_1 := c$. Suppose $c_1, \ldots, c_i$ has been found. If $c_i \neq \mathbf{0}$, let

$$v := 2^{-5n^2} \left\lfloor \frac{2^{5n^2}}{\|c_i\|_\infty} c_i \right\rfloor, \tag{17}$$

where $n := |E|$, and where $\lfloor\ \rfloor$ denotes component-wise lower integer parts. By the method of Theorem 2 we can find $u_i \in \mathbb{Z}^n$ and $q_i \in \mathbb{Z}$ such that

$$\|v - \frac{1}{q_i} u_i\|_\infty < \frac{1}{q_i} \cdot \frac{1}{2n} \quad \text{and} \quad 1 \leq q_i \leq 2^{\frac{1}{4}n(n+1)} (2n)^n \tag{18}$$

(taking $\epsilon := 1/2n$). Note that

$$\|u_i\|_\infty \leqslant q_i \leqslant 2^{n^2}(2n)^n, \tag{19}$$

since $\|v\|_\infty = 1$. Let

$$c_{i+1} := c_i - \frac{\|c_i\|_\infty}{q_i} u_i. \tag{20}$$

If $c_{i+1} = \mathbf{0}$, stop. Otherwise, repeat with i replaced by $i+1$.

Since c_{i+1} has at least one 0-component more than c_i has, as one easily derives from (17), (18) and (20), the algorithm stops after $k \leqslant n$ iterations. Let $c_1, \ldots, c_k$ be the sequence generated. Note that by (20),

$$c = \frac{\|c_1\|_\infty}{q_1} u_1 + \frac{\|c_2\|_\infty}{q_2} u_2 + \ldots + \frac{\|c_k\|_\infty}{q_k} u_k. \tag{21}$$

Now define:

$$\tilde{c} := 2^{5n^2(k-1)} u_1 + 2^{5n^2(k-2)} u_2 + \ldots + 2^{5n^2} u_{k-1} + u_k. \tag{22}$$

The vector $\tilde{c}$ has the following property:

$$\text{for each vector } x \in \{0, \pm 1\}^n\text{: if } c^T x < 0 \text{ then } \tilde{c}^T x < 0. \tag{23}$$

Indeed, let $x \in \{0, \pm 1\}^n$ with $c^T x < 0$. Choose the smallest i with $u_i^T x \neq 0$ (i exists by (21)). Then

$$c_i^T x = (c - \frac{\|c_1\|_\infty}{q_1} u_1 - \ldots - \frac{\|c_{i-1}\|_\infty}{q_{i-1}} u_{i-1})^T x = c^T x < 0. \tag{24}$$

Hence

$$\begin{aligned} u_i^T x &= (u_i - q_i v)^T x + q_i (v - c_i)^T x + q_i c_i^T x \\ &< \|u_i - q_i v\|_\infty \cdot \|x\|_1 + q_i \cdot \|v - c_i\|_\infty \cdot \|x\|_1 \\ &\leqslant \frac{1}{2n} n + 2^{n^2}(2n)^n 2^{-5n^2} n \leqslant 1, \end{aligned} \tag{25}$$

implying $u_i^T x \leqslant -1$ (as u_i is integral and $u_i^T x \neq 0$). Therefore,

$$\begin{aligned} \tilde{c}^T x &= 2^{5n^2(k-i)} u_i^T x + 2^{5n^2(k-i-1)} u_{i+1}^T x + \ldots + u_k^T x \\ &\leqslant -2^{5n^2(k-i)} + n \cdot 2^{5n^2(k-i-1)} \cdot 2^{4n^2} \\ &= 2^{5n^2(k-i)}(-1 + n \cdot 2^{-n^2}) < 0 \end{aligned} \tag{26}$$

(using $u_j^T x \leqslant \|u_j\|_\infty \cdot \|x\|_1 \leqslant 2^{n^2}(2n)^n n \leqslant 2^{4n^2}$ - cf. (19)). This proves (23).

Having determined $\tilde{c}$, give the input $G = (V, E)$ and $\tilde{c} \in \mathbb{Z}^E$ to the polynomial-time algorithm for (8). It gives a set F in $\mathscr{F}_G$ maximizing $\Sigma_{e \in F} \tilde{c}(e)$. Then F also maximizes $\Sigma_{e \in F} c(e)$. For suppose $\Sigma_{e \in F'} c(e) > \Sigma_{e \in F} c(e)$ for some $F' \in \mathscr{F}_G$. Then $c^T(\chi^F - \chi^{F'}) < 0$. By (23), $\tilde{c}^T(\chi^F - \chi^{F'}) < 0$, contradicting the fact that F maximizes $\Sigma_{e \in F} \tilde{c}(e)$.

The whole procedure consists of a number of arithmetic operations bounded by a polynomial in $|V| + |E|$. Indeed, v in (17) can be determined by binary search by $5n^2 + 1$ comparisons (for each coordinate). The method of Theorem

2 applied to v and $\epsilon = 1/2n$ takes time bounded by a polynomial in size$(v) = \Theta(n^3)$ and size$(\epsilon) = \Theta(\log n)$. Finally, the algorithm for the optimization problem applied to G and $\tilde{c}$ takes time bounded by a polynomial in $|V| + |E|$ and size$(\tilde{c}) = \Theta(n^6)$ (by (19) and (21)). Concluding, we have a strongly polynomial algorithm for (8). □

A similar result holds for the separation problem (10).

5. Totally unimodular matrices and bipartite graphs

We now come to some concrete examples of polyhedral characterizations. A prime technique in deriving polyhedral results is based on 'total unimodularity' of matrices. A matrix is called *totally unimodular* if each subdeterminant belongs to $\{0, +1, -1\}$. In particular, each entry in a totally unimodular matrix belongs to $\{0, +1, -1\}$.

The following is not difficult to see.

THEOREM 4. *Let A be a totally unimodular $m \times n$-matrix, and let b be an integral column vector in $\mathbb{R}^m$. Then each vertex of the polyhedron $\{x | Ax \leq b\}$ is integral.*

PROOF. Let x^* be a vertex of $\{x | Ax \leq b\}$. Then there exists a nonsingular $m \times m$-submatrix A' of A, with corresponding part b' of b, so that $A'x^* = b'$. Hence $x^* = (A')^{-1}b'$. As $\det A' = \pm 1$, it follows that x^* is integral. □

This theorem and extensions characterizing total unimodularity were given by HOFFMAN and KRUSKAL [13].

Let $G = (V,E)$ be a *bipartite graph*, i.e., an undirected graph whose vertex set V can be split into two classes V' and V'' so that each edge consists of a vertex in V' and a vertex in V''. Let A be the *incidence matrix* of G, i.e., A is the $V \times E$- matrix with 1 in position (v,e) if $v \in e$, and 0 otherwise.

THEOREM 5. *The incidence matrix of a bipartite graph is totally unimodular.*

PROOF. Let B be an $m \times m$-submatrix of A. We show $\det B \in \{0, \pm 1\}$ by induction on m, the case $m = 1$ being trivial. If B contains an all-zero column, then $\det B = 0$. If B contains a column with exactly one 1, we can expand $\det B$ by this column, yielding $\det B = \pm \det B'$ for some $(m-1) \times (m-1)$-submatrix B' of B. Then by induction $\det B \in \{0, \pm 1\}$. If each column of B contains exactly two 1's, we can decompose B as $\begin{bmatrix} B' \\ B'' \end{bmatrix}$ so that each column of B' has exactly one 1, and similarly for B'' (possibly after permuting rows of B). Then $(1, \ldots, 1, -1, \ldots, -1) \begin{bmatrix} B' \\ B'' \end{bmatrix} = \mathbf{0}$, and hence $\det B = 0$. □

Theorems 4 and 5 have some direct consequences. For any graph $G = (V,E)$, the polytope conv.hull $\{\chi^M | M \text{ matching}\}$ is called the *matching polytope* of G.

THEOREM 6. *Let $G=(V,E)$ be a graph. Then the matching polytope of G is equal to the set of all vectors $x \in \mathbb{R}^E$ satisfying*

$$\begin{aligned} &\text{(i)} \quad x_e \geqslant 0 \qquad (e \in E), \\ &\text{(ii)} \quad \sum_{e \ni v} x_e \leqslant 1 \quad (v \in V), \end{aligned} \tag{27}$$

if and only if G is bipartite.

PROOF. 'if': If G is bipartite, its incidence matrix A is totally unimodular, and hence also the matrix $\begin{bmatrix} -I \\ A \end{bmatrix}$ is totally unimodular. Since the system (27) is the same as $\begin{bmatrix} -I \\ A \end{bmatrix} x \leqslant \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$, we know that the polytope P defined by (27) has integral vertices only (Theorem 4). Since the integral vectors satisfying (27) are exactly the vectors χ^M for matchings M, we know that P is equal to the matching polytope of G.

'only if': If G is not bipartite, it has an odd circuit C. Let $x \in \mathbb{R}^E$ be defined by $x_e = \frac{1}{2}$ if e belongs to C, and $x_e = 0$ otherwise. Then x satisfies (27), but x does not belong to the matching polytope of G, as one easily checks. □

This theorem immediately yields a strongly polynomial algorithm for finding a maximum-weighted matching in a bipartite graph $G=(V,E)$ (which problem is one of the variants of the *optimal assignment problem*): given a weight function $c \in \mathbb{Z}^E$, a matching M in G maximizing $\Sigma_{e \in M} c(e)$ can be found by solving the linear program of maximizing $c^T x$ over (27).

This can be derived also from Theorems 1 and 3, since solving the separation problem for matching polytopes of bipartite graphs just means testing if a given vector x satisfies (27); these constraints can be checked one by one in polynomial time (there are $|V|+|E|$ constraints). This does not reflect the full power of Theorem 1 - we shall see a better illustration in the next section.

A similar result holds for the *coclique polytope* of a graph G, being conv.hull $\{\chi^C | C \text{ coclique}\}$.

THEOREM 7. *Let $G=(V,E)$ be a graph. Then the coclique polytope of G is equal to the set of all vectors $x \in \mathbb{R}^E$ satisfying*

$$\begin{aligned} &\text{(i)} \quad x_v \geqslant 0 \qquad (v \in V), \\ &\text{(ii)} \quad \sum_{v \in e} x_v \leqslant 1 \quad (e \in E), \end{aligned} \tag{28}$$

if and only if G is bipartite.

PROOF. Similarly to the proof of Theorem 6 (note that clearly also the transpose A^T of the incidence matrix of a bipartite graph is totally unimodular). $\square$

Again, one can derive from this that for bipartite graphs a maximum weighted coclique can be found in polynomial time.

The following two related results are classical. The *perfect matching polytope* of a graph is the polytope conv.hull$\{\chi^M | M$ perfect matching$\}$. A *perfect* matching is a matching covering all vertices of the graph exactly once.

THEOREM 8 (BIRKHOFF-VON NEUMANN THEOREM). *Let $G=(V,E)$ be a bipartite graph. Then the perfect matching polytope is equal to the set of all vectors x in $\mathbb{R}^E$ satisfying*

$$\begin{aligned} &\text{(i)} \quad x_e \geqslant 0 \qquad (e \in E), \\ &\text{(ii)} \quad \sum_{e \ni v} x_e = 1 \quad (v \in V). \end{aligned} \tag{29}$$

PROOF. The theorem follows from the total unimodularity of the matrix

$$\begin{bmatrix} -I \\ A \\ -A \end{bmatrix}, \tag{30}$$

where A is the incidence matrix of G. $\square$

This theorem is better known in the following equivalent formulation: each doubly stochastic matrix is a convex combination of permutation matrices. (A matrix is *doubly stochastic* if it is nonnegative and if each row sum and each column sum is equal to 1.)

Theorem 8 yields the polynomial-time solvability of the problem of finding a maximum (and similarly, a minimum) weighted perfect matching in a bipartite graph.

THEOREM 9 (KÖNIG-EGERVÁRY THEOREM). *Let $G=(V,E)$ be a bipartite graph. Then*

$$\text{(i)} \max\{|M| \,|M \text{matching}\} = \min\{|W| \,|W \subseteq V; \forall e \in E: \exists v \in W: v \in e\}; \tag{31}$$

$$\text{(ii)} \max\{|C| \,|C \text{ coclique}\} = \min\{|F| \,|F \subseteq E; \forall v \in V: \exists e \in F: v \in e\}.$$

PROOF. Let A be the incidence matrix of G. Then by the total unimodularity of A:

$$\begin{aligned} \max\{|M| \,|M \text{matching}\} &= \max\{\mathbf{1}^T x | x \geqslant 0; Ax \leqslant \mathbf{1};\ x \text{ integral}\} \\ &= \max\{\mathbf{1}^T x | x \geqslant 0; Ax \leqslant \mathbf{1}\} = \min\{y^T \mathbf{1} | y \geqslant 0; y^T A \geqslant \mathbf{1}^T\} \\ &= \min\{y^T \mathbf{1} | y \geqslant 0; y^T A \geqslant \mathbf{1}^T; y \text{ integral}\} \end{aligned} \tag{32}$$

$$= \min\{\,|W|\ |W \subseteq V; \forall e \in E: \exists v \in W: v \in e\}.$$

This shows (i). Equation (ii) is shown similarly. $\square$

REMARK. Similar results can be derived for flows in directed graphs (like the max-flow min-cut theorem), using the fact that any $\{0,+1,-1\}$-matrix with in each column at most one 1 and at most one -1, is totally unimodular.

Here we mention SEYMOUR's deep result [24] that each totally unimodular matrix can be decomposed, in a certain way, into matrices described in the previous sentence and into two certain totally unimodular 5×5-matrices. It yields a polynomial-time test for the total unimodularity of matrices (clearly, checking all subdeterminants would require exponential time).

6. THE MATCHING POLYTOPE OF AN ARBITRARY GRAPH

If G is not bipartite, the inequalities (27) are not enough to determine the matching polytope. A famous theorem of EDMONDS [5] gives the inequalities determining the matching polytope of a not-necessarily bipartite graph. Similarly, Edmonds characterized the perfect matching polytope, which we discuss first. We follow the proof of [21].

THEOREM 10 (EDMONDS' MATCHING POLYTOPE THEOREM). *For any graph $G=(V,E)$ the perfect matching polytope is equal to the set of vectors x in $\mathbb{R}^E$ satisfying*

$$\begin{aligned} &\text{(i)} && x_e \geqslant 0 && (e \in E),\\ &\text{(ii)} && \sum_{e \ni v} x_e = 1 && (v \in V),\\ &\text{(iii)} && x(\delta(U)) \geqslant 1 && (U \subseteq V,\ |U| \text{ odd}).\end{aligned} \tag{33}$$

Here $\delta(U)$ denotes the set of edges e in E with $|e \cap U| = 1$, and $x(\delta(U)) := \Sigma_{e\in\delta(U)} x_e$.

PROOF. Let P be the perfect matching polytope of G, and let Q be the set of vectors satisfying (33). As $\chi^M \in Q$ for each perfect matching M, it follows that $P \subseteq Q$ - the content of the theorem is the converse inclusion.

Let G be a smallest graph with $Q \not\subseteq P$ (that is, with $|V|+|E$ minimal), and let x be a vertex of Q not contained in P. Then $0<x_e<1$ for all e in E - otherwise we could delete e from G to obtain a smaller counterexample. Moreover, $|E|>|V|$ - otherwise, either G is disconnected (in which case one of the components of G will be a smaller counterexample), or G has a point v of degree one (in which case the edge e incident to v has $x_e = 1$), or G is an even circuit (for which the theorem trivially holds).

Since x is a vertex of Q, there are $|E|$ independent constraints among (33) satisfied by x with equality, and hence there exists a $U \subseteq V$ with $|U|$ odd, $|U| \geqslant 3, |V\setminus U| \geqslant 3$ and $x(\delta(U)) = 1$. Let G_1 and G_2 arise from G by contracting U and $V\setminus U$, respectively, and let x_1 and x_2 be the corresponding projections of x onto the edge sets of G_1 and G_2, respectively. Since x_1 and x_2 satisfy inequalities (33) for the smaller graphs G_1 and G_2, respectively, it follows that x_1 and x_2 can be decomposed as convex combinations of characteristic vectors of perfect matchings in G_1 and G_2, respectively. These decompositions can be easily glued together to form a decomposition of x as a convex combination of perfect matchings, contradicting our assumption.

(This glueing can be done, e.g., as follows. By the rationality of x (as it is a vertex of Q), there exists a natural number K such that, for $i=1,2$, Kx_i is the sum of the characteristic vectors of the perfect matchings $M_1^i, \ldots, M_K^i$ of G_i (possibly with repetitions). Since, for each e in $\delta(U)$, e is contained in $Kx(e)$ of the M_j^1 as well as in $Kx(e)$ of the M_j^2, we may assume that $M_j^1 \cap M_j^2 \neq \varnothing$ for $j=1,\ldots,K$. It follows that Kx is the sum of the characteristic vectors of the perfect matchings $M_1^1 \cup M_1^2, \ldots, M_K^1 \cup M_K^2$ of G, and hence that x itself is a convex combination of perfect matchings in G.) $\square$

Application of Theorem 1 now becomes more illustrative than in Section 5. By Theorem 1, we can find a maximum weighted perfect matching in a graph in polynomial time, if we can solve the separation problem for the perfect matching polytope in polynomial time. This last can be shown as follows (following PADBERG and RAO [18]). For a given $x \in \mathbb{Q}^E$ we have to test if x satisfies (33). Testing the inequalities in (i) and (ii) can be done easily by checking them one by one. If one of them is not satisfied, we know that x does not belong to the perfect matching polytope, and the violated constraint gives a separating hyperplane. So we may assume that x satisfies (33) (i) and (ii). If $|V|$ is odd, then clearly (33) (iii) is not satisfied for $U:=V$. So we may assume that $|V|$ is even. We cannot check the constraints in (iii) one by one in polynomial time, simply because there are exponentially many of them. Yet, there is a polynomial-time method of checking them. Indeed, first note that from Ford-Fulkerson's max-flow min-cut algorithm we can derive a polynomial-time algorithm having the following as in- and output:

(34)
input: subset W of V;
output: a subset T of V such that $W \cap T \neq \varnothing \neq W \setminus T$ and such that $x(\delta(T))$ is as small as possible.

To see this, consider x as a capacity function on E, and determine for each pair $r,s \in W$, a cut of minimum capacity separating r and s. That is, we find a subset $T_{r,s}$ of V so that $r \in T_{r,s}$, $s \notin T_{r,s}$ and such that $\text{cap}(\delta(T_{r,s})) := x(\delta(T_{r,s}))$ is minimal. Taking $T := T_{r,s}$ for that pair r,s for which $\text{cap}(\delta(T_{r,s}))$ is as small as possible, we obtain T as required.

We next describe recursively an algorithm with the following in- and output:

(35)
input: subset W of V with $|W|$ even;
output: subset U of V such that $|W \cap U|$ is odd and such that $x(\delta(U))$ is as small as possible.

First we find with the algorithm (34) a subset T of V with $W \cap T \neq \varnothing \neq W \setminus T$ and such that $x(\delta(T))$ is minimal. If $|W \cap T|$ is odd we are done. If $|W \cap T|$ is even, call, recursively, the algorithm (35) for the inputs $W \cap T$ and $W \cap \overline{T}$, respectively, where $\bar{T} := V \setminus T$. Let it yield a subset U' of V such that $|W \cap T \cap U'|$ is odd and $x(\delta(U'))$ is minimal, and a subset U'' of V such that $|W \cap \bar{T} \cap U''|$ is odd and $x(\delta(U''))$ is minimal. Without loss of generality, $W \setminus T \not\subseteq U'$ (otherwise replace U' by $V \setminus U'$) and $W \setminus \bar{T} \not\subseteq U''$ (otherwise replace U'' by $V \setminus U''$).

We claim that if $x(\delta(T \cap U')) \leqslant x(\delta(\bar{T} \cap U''))$ then $U := T \cap U'$ is output of (35) for input W, and otherwise $U := \bar{T} \cap U''$ is output of (35) for input W. To see that this output is justified, suppose to the contrary that there exists a subset Y of V such that $|W \cap Y|$ is odd and $x(\delta(Y)) < x(\delta(T \cap U'))$ and $x(\delta(Y)) < x(\delta(\bar{T} \cap U''))$. Then either $|W \cap Y \cap T|$ is odd or $|W \cap Y \cap \bar{T}|$ is odd (since $|W \cap T|$ is even). *Case 1*: $|W \cap Y \cap T|$ is odd. Then $x(\delta(Y)) \geqslant x(\delta(U'))$, since U' is output of (35) for input $W \cap T$. Moreover, $x(\delta(T \cup U')) \geqslant x(\delta(T))$, since T is output of (34) for input W, and since $W \cap (T \cup U') \neq \varnothing \neq W \setminus (T \cup U')$. Therefore, we have a contradiction:

$$\begin{aligned} x(\delta(Y)) &\geqslant x(\delta(U')) \geqslant x(\delta(T \cap U')) + x(\delta(T \cup U')) - x(\delta(T)) \\ &\geqslant x(\delta(T \cap U')) > x(\delta(Y)) \end{aligned} \tag{36}$$

(the second inequality follows since $x(\delta(A)) + x(\delta(B)) \geqslant x(\delta(A \cap B)) + x(\delta(A \cup B))$ for all subsets A and B of V). *Case 2*: $|W \cap Y \cap \bar{T}|$ is odd. Similarly.

Given the polynomiality of the algorithm for (34), it is not difficult to see that also the described algorithm for (35) has polynomially bounded running time.

As a consequence, we can test the inequalities (33) (iii) in polynomial time, which implies the polynomial-time solvability of the problem of finding a maximum weighted perfect matching. In fact, EDMONDS [5] gave a direct polynomial-time algorithm for this problem, yielding Theorem 10 as a by-product. We have followed the above line to illustrate the use of Theorem 1.

By a standard construction, Edmonds' characterization of the matching polytope can be derived from Theorem 10.

THEOREM 11. *For any graph $G = (V,E)$, the matching polytope is equal to the set of all vectors x in $\mathbf{R}^E$ satisfying*

$$\begin{array}{lll} \text{(i)} & x_e \geqslant 0 & (e \in E), \\ \text{(ii)} & \sum_{e \ni v} x_e \leqslant 1 & (v \in V), \\ \text{(iii)} & \sum_{e \subseteq U} x_e \leqslant \lfloor \tfrac{1}{2}|U| \rfloor & (U \subseteq V,\ |U| \text{ odd}). \end{array} \tag{37}$$

PROOF. Again it is clear that each vector in the matching polytope satisfies (37), as χ^M satisfies (37) for each matching M. To see that the inequalities (37) are enough, let $x \in \mathbf{R}^E$ satisfy (37). Let $G^* = (V^*, E^*)$ be a disjoint copy of G, where the copy of vertex v will be denoted by v^*, and the copy of edge

$e=\{v,w\}$ will be denoted by $e^*=\{v^*,w^*\}$. Let $\tilde{G}$ be the graph with vertex set $V\cup V^*$ and with edge set $E\cup E^*\cup\{\{v,v^*\}|v\in V\}$. Define $\tilde{x}(e):=\tilde{x}(e^*):=x(e)$ for e in E, and $\tilde{x}(\{v,v^*\}):=1-x(\delta(v))$ for v in V. Now conditions (33) are easily derived for $\tilde{x}$ with respect to G. Constraints (i) and (ii) are trivial. To prove (iii) in (33), we have to show, for $V_1,V_2\subseteq V$ with $|V_1|+|V_2|$ odd, that $\tilde{x}(\delta(V_1\cup V_2^*))\geqslant 1$. Indeed, we may assume, without loss of generality, that $|V_1\setminus V_2|$ is odd. Hence

$$\begin{aligned}\tilde{x}(\delta(V_1\cup V_2^*)) &= \tilde{x}(\delta(V_1\setminus V_2))+\tilde{x}(\delta(V_2^*\setminus V_1^*))\geqslant\tilde{x}(\delta(V_1\setminus V_2))\\ &=|V_1\setminus V_2|-2\cdot\sum_{e\subseteq V_1\setminus V_2}x_e\geqslant 1,\end{aligned}\tag{38}$$

by (37) (iii).

Hence $\tilde{x}$ is a convex combination of perfect matchings in $\tilde{G}$. By restriction to x and G it follows that x is a convex combination of matchings in G. □

In a way similar to above one can derive a polynomial-time algorithm finding a maximum weighted matching.

Related to Theorem 11 is the following min-max relation due to TUTTE [26] and BERGE [1].

THEOREM 12 (TUTTE-BERGE FORMULA). *For any graph* $G=(V,E)$

$$\max\{|M|\,|M\text{ matching}\}=\min_{U\subseteq V}\frac{|V|+|U|-\mathcal{O}(V\setminus U)}{2}\tag{39}$$

where $\mathcal{O}(V\setminus U)$ *denotes the number of odd components of the graph induced by* $V\setminus U$.

The minimum here can be easily seen to be equal to:

$$\min\{|U|+\sum_{i=1}^{t}\lfloor\tfrac{1}{2}|V_i|\rfloor\,|U,V_1,\ldots,V_t\subseteq V,\text{ so that each edge intersects } U \text{ or is contained in one of the } V_i\}.\tag{40}$$

The content of the Tutte-Berge formula is that when we write $\max\{|M|\,|M$ matching$\}$ equivalently as maximizing $\mathbf{1}^Tx$ over (37), we obtain a linear program with integral optimal primal *and* dual solutions.

7. CUTTING PLANES

Quite often the problem of characterizing the convex hull of certain $\{0,1\}$-vectors amounts to characterizing, for some polytope P, the polytope

$$P_I:=\text{conv.hull }\{x\in P|x\text{ integral}\}.\tag{41}$$

P_I is called the *integer hull* of P. E.g., if $G=(V,E)$ is a graph, and

$$P:=\{x\in\mathbb{R}^E|x_e\geqslant 0\ \ (e\in E),\sum_{e\ni v}x_e\leqslant 1\ \ (v\in V)\},\tag{42}$$

the integral vectors in P are exactly the characteristic vectors of matchings, and hence P_I is equal to the matching polytope of G. Similarly, for

$$P := \{x \in \mathbf{R}^V | x_v \geqslant 0 \ \ (v \in V), \sum_{v \in e} x_v \leqslant 1 \ \ (e \in E)\}, \tag{43}$$

P_I is the coclique polytope of G.

There is a way of deriving the inequalities determining P_I from those determining P - the *cutting plane method*. Its basics were given by GOMORY [10]. The following description is due to CHVÁTAL [2] and SCHRIJVER [20].

Clearly, if H is a *rational halfspace*, i.e., H is of form

$$H = \{x \in \mathbf{R}^n | a^T x \leqslant \beta\}, \tag{44}$$

where $a \in \mathbf{Q}^n$, $a \neq \mathbf{0}$, $\beta \in \mathbf{Q}$, we may assume without loss of generality that a is integral, and that the components of a are relatively prime. In that case:

$$H_I = \{x \in \mathbf{R}^n | a^T x \leqslant \lfloor \beta \rfloor \}. \tag{45}$$

H_I arises from H by shifting its bounding hyperplane until it contains integral vectors.

Now define for any set P:

$$P' := \bigcap_{H \supseteq P} H_I, \tag{46}$$

where H ranges over all rational halfspaces containing P. Since $H \supseteq P$ implies $H_I \supseteq P_I$, it follows that $P_I \subseteq P'$. It can be shown that if P is a rational polyhedron (i.e., a polyhedron determined by linear inequalities with rational coefficients), then P' is a polyhedron again.

To P' we can apply this operation again, yielding P''. Generally $P'' \neq P'$ - consider e.g. the following example.

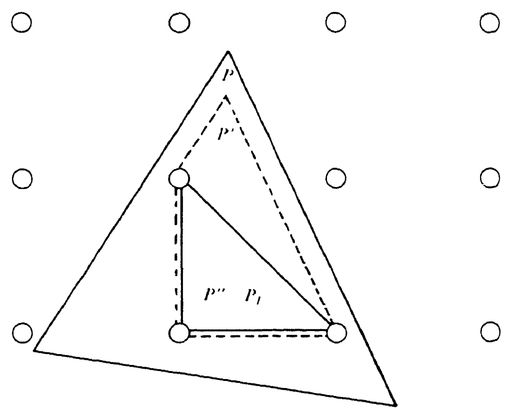


FIGURE 1

So there is a sequence of polyhedra containing P_I:

$$P \supseteq P' \supseteq P'' \supseteq P''' \supseteq \ldots \supseteq P_I. \tag{47}$$

Denoting the $(t+1)$-th set in this sequence by $P^{(t)}$, the following can be shown.

THEOREM 13. *For each rational polyhedron P there exists a number t such that* $P^{(t)}=P_I$.

The theorem is the theoretical essence of the cutting plane method of Gomory. The equation $ax=\lfloor\beta\rfloor$ defining H_I, or more strictly the hyperplane $\{x|ax=\lfloor\beta\rfloor\}$, is called a *cutting plane.*

The smallest t for which $P^{(t)}=P_I$ can be considered as a measure for the complexity of P_I relative to that of P. In a sense, P' is conceptually near to P, P'' to P', etc.

Let us study some specific polyhedra related to graphs. Let $G=(V,E)$ be an undirected graph, and let $P\subseteq\mathbf{R}^E$ be the polytope determined by the inequalities

$$\begin{aligned}&\text{(i)}\ \ x_e\geqslant 0 \qquad (e\in E),\\ &\text{(ii)}\ \sum_{e\ni v}x_e\leqslant 1 \quad (v\in V).\end{aligned}\tag{48}$$

So P_I is the matching polytope of G. By Theorem 6, $P=P_I$ if and only if G is bipartite. It is not difficult to see that for each graph G, P' is the set of all vectors x satisfying (48) and satisfying

$$\sum_{e\subseteq U}x_e\leqslant\lfloor\tfrac{1}{2}|U|\rfloor \quad (U\subseteq V, |U|\text{odd}).\tag{49}$$

(Of course, there are infinitely many halfspaces H containing P, but the corresponding inequalities $ax\leqslant\lfloor\beta\rfloor$ all are implied by the inequalities in (48) and (49).) So Theorem 11 in fact tells us that $P'=P_I$ for each graph G.

Next consider for any graph $G=(V,E)$ the polytope $P\subseteq\mathbf{R}^V$ determined by the inequalities:

$$\begin{aligned}&\text{(i)}\ \ x_v\geqslant 0 \qquad (v\in V),\\ &\text{(ii)}\ \sum_{v\in e}x_v\leqslant 1 \quad (e\in E).\end{aligned}\tag{50}$$

For this P, P_I is the coclique polytope of G. By Theorem 7, $P=P_I$ if and only if G is bipartite. It is not difficult to check that for any graph G, the polytope P' is the set of vectors x satisfying (50) and satisfying

$$\sum_{v\in V(C)}x_v\leqslant\lfloor\tfrac{1}{2}|V(C)|\rfloor \quad (C\text{ odd circuit}),\tag{51}$$

where $V(C)$ is the vertex set of C, and where an odd circuit is a circuit C with $|V(C)|$ odd.

CHVÁTAL [2] has shown that there exists no fixed t such that $P^{(t)}=P_I$ for each graph G. The problem of finding a largest coclique in a graph is NP-complete, and hence probably not polynomially solvable. Therefore, by Theorem 1, probably there is no 'decent' description of the coclique polytope for all graphs. It is conjectured that for each fixed t, when we restrict ourselves to graphs which have $P^{(t)}=P_I$, the problem of finding a maximum weighted coclique is polynomially solvable (in fact, this problem can be shown to belong to $NP\cap$ co-NP). The conjecture is true for $t=0$ and $t=1$ (using Theorem 1). If we want to show it for $t=2$, by Theorem 1 it suffices to show that the following problem is polynomially solvable: decide if a given vector $x\in\mathbb{R}^V$ belongs to P'', and if not, find a separating hyperplane.

In Section 9 we shall see a class of graphs with $P'=P_I$. As a preparation, we discuss in Section 8 another geometric tool.

8. Binary configurations

We now come to a geometric method of a nature different from those discussed above. Let us call a set $x_1,\ldots,x_k$ of vectors in some space $GF(2)^n$ a *binary configuration.* Usually, the zero-vector will not be among $x_1,\ldots,x_k$, and hence we can consider a binary configuration as a configuration in $PG(d,2)$, the d-dimensional projective space over $GF(2)$.

A well-known binary configuration is the *Fano-configuration* $(=PG(2,2))$ which is the binary configuration represented by the columns of

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{52}$$

and whose 7 points and 7 lines can be represented as:

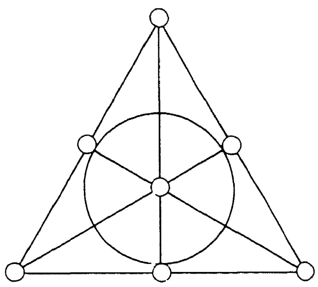


Figure 2

The lines are represented by 6 line segments and one circle. The reader familiar with the Fano-configuration might have tried to draw it on the paper in such a way that *all* 7 lines become straight line segments, so that not all 7 points are on one and the same line in the plane. After some trials one will be convinced that this is not possible, and it is not hard to show this algebraically.

In fact, Fano is in a sense a critical example. A famous and deep theorem of Tutte [26] states that a binary configuration can be embedded in euclidean space so that each subset of points span a space of the same dimension in the binary space as they do in euclidean space, if and only if the binary configuration does not 'contain' the Fano-configuration or its 'dual'.

We shall make terms more precise. Call a binary configuration $x_1,\ldots,x_k$

in $GF(2)^n$ *embeddable in euclidean space* if there exists a function $\phi:\{x_1, \ldots, x_k\} \to \mathbb{R}^n$ so that for each subset T of $\{x_1, \ldots, x_k\}$, the dimension of $<T>$ in $GF(2)^n$ is equal to the dimension of $<\phi[T]>$ in $\mathbb{R}^n$. The function ϕ is called an *embedding*.

Deleting x_1 from $x_1, \ldots, x_k$ means replacing $x_1, \ldots, x_k$ by $x_2, \ldots, x_k$. *Projecting along* x_1 or *contracting* x_1 means replacing $x_1, \ldots, x_k$ by

$$x_2/<x_1>, \ldots, x_k/<x_1>, \tag{53}$$

where $../<x_1>$ means projecting .. into the quotient space $GF(2)^n/<x_1>$. Two binary configurations $x_1, \ldots, x_k$ and $x'_1, \ldots, x'_k$ are called *geometrically the same* if there is a linear transformation bringing $x_1, \ldots, x_k$ one-to-one to $x'_1, \ldots, x'_k$. Thus the Fano-configuration is geometrically the same as the set of columns of

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{54}$$

A binary configuration Y is called a *minor* of a binary configuration X, if Y can be obtained from X by deletion, projection and permutation of vectors, up to being geometrically the same.

Trivially, embeddability in euclidean space is maintained under deletion. It is also not difficult to see that it is maintained under projection. Indeed, if $\phi:\{x_1, \ldots, x_k\} \to \mathbb{R}^n$ forms an embedding, then also $x_i/<x_1> \mapsto \phi(x_i)/<\phi(x_1)>$ forms an embedding. It follows that embeddability in euclidean space is maintained under taking minors.

The *dual* of the Fano-configuration is the configuration represented by the columns of

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{55}$$

Geometrically, the dual of the Fano-configuration is formed by the 7 points obtained from $PG(3,2)$ by deleting one projective plane and one further (arbitrary) point. Also this configuration is not embeddable in euclidean space.

Now Tutte's theorem is:

THEOREM 14. *A binary configuration $x_1, \ldots, x_k$ is embeddable in euclidean space, if and only if it has no minor equal to the Fano-configuration or its dual.*

In order to interprete and use this difficult theorem, we first make a further study of binary configurations. To each binary configuration $x_1, \ldots, x_k$ we can associate the binary space or *binary code* C of all vectors z in $GF(2)^k$

satisfying $[x_1, \ldots, x_k]z = \mathbf{0}$ (considering $x_1, \ldots, x_k$ as column vectors). Clearly, each linear subspace of $GF(2)^k$ is associated in this way to some binary configuration. Two binary configurations are geometrically the same if and only if the associated binary codes are the same.

The binary configuration $y_1, \ldots, y_k$ is called *dual* to $x_1, \ldots, x_k$ if the associated binary codes are each others orthogonal complements. Note that the well-known Hamming code is associated this way with the Fano-configuration, and the dual Hamming code to the dual of the Fano-configuration.

If C is the binary code associated to the binary configuration $x_1, \ldots, x_k$, and if we delete x_1, the associated binary code becomes $\{z| \begin{bmatrix} 0 \\ z \end{bmatrix} \in C\}$; if we would project along x_1, the associated code becomes $\{z| \begin{bmatrix} 0 \\ z \end{bmatrix} \in C$ or $\begin{bmatrix} 1 \\ z \end{bmatrix} \in C\}$. Thus a binary configuration contains the Fano-configuration or its dual as a minor, if and only if by these operations the associated binary code can be transformed into the Hamming code or its dual.

This is all quite standard linear algebra. More specific is the following definition. A subspace C of $GF(2)^n$ is said to be *orientable* if we can associate with each $x \in C$ a vector x' in $\{0, \pm 1\}^n$ and with each $y \in C^\perp$ a vector y'' in $\{0, \pm 1\}^n$ in such a way that:

$$\begin{aligned} &\text{(i)} \quad \forall x \in C: x \text{ and } x' \text{ have the same support;} \\ &\text{(ii)} \quad \forall y \in C: y \text{ and } y'' \text{ have the same support;} \\ &\text{(iii)} \quad \forall x \in C, \forall y \in C^\perp: (x')^T y'' = 0. \end{aligned} \tag{56}$$

The following theorem now is not so difficult to prove:

THEOREM 15. *A binary configuration is embeddable in euclidean space, if and only if the associated binary code is orientable.*

REMARK. Another deep theorem characterizing binary configurations embeddable in euclidean space is due to SEYMOUR [24]. To describe this we need some concepts.

A binary configuration is called *graphic* if it is geometrically the same as a binary configuration $x_1, \ldots, x_k$ where each vector x_i has exactly two 1's. It follows that the associated binary code is the 'cycle space' of a graph. A binary configuration is *cographic* if it is the dual of a graphic configuration. So the associated binary code is the 'cocycle space' of a graph. It is not difficult to see that graphic and cographic configurations are embeddable in euclidean space.

Let be given two binary configurations $x_1, \ldots, x_k$ and $y_1, \ldots, y_t$, not containing the zero-vector, and let $d := \dim <x_1, \ldots, x_k> + \dim <y_1, \ldots, y_t>$ (where $<..>$ denotes projective space generated by .., and dim denotes projective dimension).

First, the two configurations can be embedded into $PG(d, 2)$ so that $<x_1, \ldots, x_k> \cap <y_1, \ldots, y_t> = \emptyset$. If $k \geq 1$, $t \geq 1$, then the binary configuration $x_1, \ldots, x_k, y_1, \ldots, y_t$ is called the *1-sum* of $x_1, \ldots, x_k$ and

$y_1, \ldots, y_t$.

Second, the two configurations can be embedded into $PG(d,2)$ so that $x_1 = y_1$ and $\langle x_1, \ldots, x_k \rangle \cap \langle y_1, \ldots, y_t \rangle = \{x_1\}$. If $k \geqslant 3$, $t \geqslant 3$, then the binary configuration $x_2, \ldots, x_k, y_2, \ldots, y_k$ is called a *2-sum* of $x_1, \ldots, x_k$ and $y_1, \ldots, y_t$.

Third, let x_1, x_2, x_3 form a line and let y_1, y_2, y_3 form a line. Then the two configurations can be embedded into $PG(d,2)$ so that $x_1 = y_1$, $x_2 = y_2$, $x_3 = y_3$ and $\langle x_1, \ldots, x_k \rangle \cap \langle y_1, \ldots, y_t \rangle = \{x_1, x_2, x_3\}$. If $k \geqslant 7$, $t \geqslant 7$, then the binary configuration $x_4, \ldots, x_k, y_4, \ldots, y_t$ is called a *3-sum* of $x_1, \ldots, x_k$ and $y_1, \ldots, y_t$.

Now Seymour's theorem is:

THEOREM 16. *A binary configuration is embeddable in euclidean space if and only if it can be obtained by making 1-, 2- and 3-sums from graphic configurations, cographic configurations, and the binary configuration made by the columns of*

$$\begin{bmatrix} 1&0&0&0&0&1&1&1&0&0 \\ 0&1&0&0&0&0&1&1&1&0 \\ 0&0&1&0&0&0&0&1&1&1 \\ 0&0&0&1&0&1&0&0&1&1 \\ 0&0&0&0&1&1&1&0&0&1 \end{bmatrix}. \tag{57}$$

Note that Tutte's theorem makes 'not embeddable in euclidean space' constructible, while Seymour's theorem makes 'embeddable in euclidean space' constructible.

Seymour's theorem has the following implication for totally unimodular matrices. Let A be a totally unimodular matrix. Then the binary configuration represented by the columns of the matrix $[I\,A]$ (forgetting the $-$ signs) is embeddable in euclidean space (as can be seen by not forgetting the $-$ signs). Hence Seymour's theorem implies that A can be decomposed into 'network matrices', their transposes, and the following two matrices:

$$\begin{bmatrix} -1&1&-1&0&0 \\ 0&-1&1&-1&0 \\ 0&0&-1&1&-1 \\ -1&0&0&-1&1 \\ 1&-1&0&0&-1 \end{bmatrix} \text{ and } \begin{bmatrix} 1&1&1&1&1 \\ 1&1&1&0&0 \\ 1&0&1&1&0 \\ 1&0&0&1&1 \\ 1&1&0&0&1 \end{bmatrix}. \tag{58}$$

The meaning of 'can be decomposed into' becomes clear after elaborating the meaning of 1-, 2-, and 3-sum. Seymour's theorem also yields a polynomial-time test for total unimodularity.

9. Binary configurations and combinatorial optimization

There are several applications of the geometric results discussed in Section 8, e.g., to 2-commodity flows, the maximum cut problem, the Chinese postman problem, matchings - see Seymour [23, 25]. We here restrict ourselves to one application to the coclique problem.

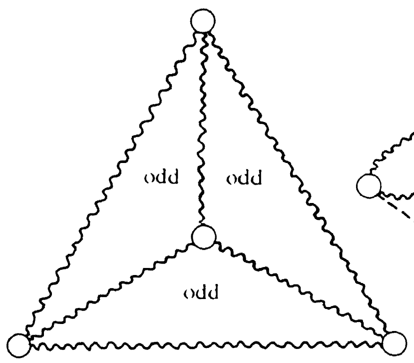
Let $G=(V,E)$ be an undirected graph, and consider the linear subspace C_G of $\{0,1\}\times\{0,1\}^E$ of all vectors $\begin{bmatrix}\epsilon \\ \chi^F\end{bmatrix}$ where F is a collection of edges so that each vertex of G is incident to an even number of edges in F, where χ^F denotes the characteristic vector of F in $\{0,1\}^E$, and where $\epsilon=0$ if $|F|$ is even and $\epsilon=1$ if $|F|$ is odd. Let K_G be the binary configuration (unique up to being geometrically the same) associated with C_G. Note that

$$C_G^{\perp} = \{ \begin{bmatrix}\epsilon \\ \chi^F\end{bmatrix} | \epsilon = 0 \text{ and } F = \delta(W) \text{ for some } W \subseteq V, \text{ or } \epsilon = 1 \text{ and } F = V \setminus \delta(W) \text{ for some } W \subseteq V\}. \tag{59}$$

The following lemma is easy to check:

Lemma. *K_G does not contain the Fano-configuration or its dual as a minor, if and only if G has no subgraph isomorphic to one of the following graphs:*

(60)



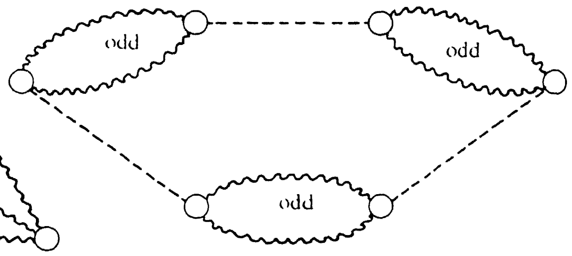


Figure 3 'odd-K_4' Figure 4 'odd-K_3^2'

Here wriggled lines represent paths of positive length and dotted lines represent lines of positive or zero length; *odd* in a face means that the circuit enclosing it has an odd number of edges.

The following theorem is due to Gerards [7].

Theorem 17. *A graph G has no subgraph isomorphic to one of the graphs in* (60), *if and only if we can orient the edges of G in such a way that in each circuit, the number of edges directed one way differs by at most one from the number of edges directed in the other way.*

PROOF. The 'if' part follows easily, since the graphs in (60) do not have the required orientation, as one easily checks.

To see the 'only if' part, we may assume that G is connected. From the Lemma we know that K_G does not contain the Fano-configuration or its dual as a minor. By Tutte's theorem (Theorem 14), K_G is embeddable in euclidean space. Hence, by Theorem 15, C_G is orientable. Let the oriented vectors be indicated by $'$ and $''$ as in (56). So for each $x \in C_G$, $x' \in \{0,\pm 1\} \times \{0,\pm 1\}^E$ and for each $y \in C_G^{\perp}$, $y'' \in \{0,\pm 1\} \times \{0,\pm 1\}^E$. Without loss of generality

$$\begin{bmatrix} 1 \\ \chi^E \end{bmatrix}'' = \begin{bmatrix} 1 \\ \chi^E \end{bmatrix} \tag{61}$$

since we can multiply a certain coordinate by -1 throughout in all x' and all y'', not violating (56).

Let M be the matrix with columns all vectors

$$\begin{bmatrix} 0 \\ \chi^{\delta(v)} \end{bmatrix}'' \tag{62}$$

for v in V. Let T be a spanning tree in G. Since replacing y'' by $-y''$ does not change (56), we may assume that in any row of M corresponding to an edge in T there is exactly one 1 and one -1.

Now consider any other row, corresponding to edge $e \notin T$. There exist edges $e_1, \ldots, e_k$ in T so that $e_1, \ldots, e_k$, e form a circuit, say C. Since $\begin{bmatrix} \epsilon \\ \chi^C \end{bmatrix}'$ is a $\{0,\pm 1\}$-vector with $M^T \begin{bmatrix} \epsilon \\ \chi^C \end{bmatrix}' = \mathbf{0}$, it follows that the e-th row of M is a linear combination of the rows $e_1, \ldots, e_k$. Since each of the rows $e_1, \ldots, e_k$ has row sum 0, also row e has row sum 0, i.e., it has exactly one 1 and one -1.

So all rows of M (except for the top row) have exactly one 1 and one -1. This gives us an orientation of G: orient any edge e from v to w if M has a $+1$ in position (e,v) and a -1 in position (e,w).

We show that this is an orientation as required. Let C be a circuit in G. Let $\epsilon = 0$ if $|C|$ is even, and $\epsilon = 1$ if $|C|$ is odd. Now, since

$$M^T \begin{bmatrix} \epsilon \\ \chi^C \end{bmatrix}' = \mathbf{0} \tag{63}$$

we know that the coordinates e where $\begin{bmatrix} \epsilon \\ \chi^C \end{bmatrix}'$ is $+1$ and -1, respectively, corresponds to edges e in C oriented one way and the other way, respectively.

Since

$$\left[\begin{bmatrix} \epsilon \\ \chi^C \end{bmatrix}' \right]^T \begin{bmatrix} 1 \\ \chi^E \end{bmatrix} = 0 \tag{64}$$

(cf. (61)), it follows that in C the orientation satisfies the condition described in the theorem. □

Gerards showed that Theorem 17 implies the following result of GERARDS and SCHRIJVER [9].

THEOREM 18. *Let $G=(V,E)$ be a graph, without isolated vertices not containing a subgraph isomorphic to the odd-K_4 in (60). Then the coclique polytope of G is equal to the set of all vectors x in $\mathbb{R}^V$ satisfying*

$$\begin{array}{lll} \text{(i)} & x_v \geq 0 & (v \in V), \\ \text{(ii)} & \sum_{v \in e} x_v \leq 1 & (e \in E), \\ \text{(iii)} & \sum_{v \in V(C)} x_v \leq \lfloor \tfrac{1}{2} |V(C)| \rfloor & (C \text{ circuit with } |V(C)| \text{ odd}). \end{array} \tag{65}$$

PROOF (sketch). Let P be the set of vectors satisfying (65). Let $G=(V,E)$ be a counterexample to the theorem with $|V|$ as small as possible. First one shows that a minimal counterexample to the theorem should be 3-connected (i.e., there are no two vertices whose removal makes the graph disconnected) − otherwise one could make a smaller counterexample. It is not difficult to check that if a graph is 3-connected and does not contain an odd-K_4, then it neither contains an odd-K_3^2. It follows by Theorem 17 that G can be oriented so that:

> in any circuit, the number of edges oriented one way differs by at most one from the number of edges oriented the other way. (66)

Let A denote the set of oriented edges, and let $A^{-1} := \{(v,w) | (w,v) \in A\}$. We now first show the following claim.

CLAIM. *A vector x belongs to P if and only if there exist vectors $y,z \in \mathbb{R}^V$ such that*

$$\begin{array}{lll} \text{(i)} & 0 \leq x_v \leq y_v + z_v & (v \in V), \\ \text{(ii)} & y_v + z_w \leq 1 & ((v,w) \in A), \\ \text{(iii)} & y_v + z_w \leq 0 & ((v,w) \in A^{-1}). \end{array} \tag{67}$$

PROOF OF THE CLAIM. 'if': If there exist y,z satisfying (67), condition (i) in (65) is trivial. Condition (ii) holds as for any $\{v,w\} \in E$

$$x_v + x_w \leq (y_v + z_v) + (y_w + z_w) = (y_v + z_w) + (y_w + z_v) \leq 1, \tag{68}$$

since either (v,w) or (w,v) belongs to A.

To check condition (iii), let C be an odd circuit in G. Let $v_0, v_1, \ldots, v_k = v_0$ be a cyclic order of the vertices in C so that $|A \cap \{(v_{i-1}, v_i) | i = 1, \ldots, k\}| = \lfloor \tfrac{1}{2} k \rfloor = \lfloor \tfrac{1}{2} |V(C)| \rfloor$. Then

$$\sum_{v \in V(C)} x_v = \sum_{i=1}^{k} x_{v_i} \leqslant \sum_{i=1}^{k} (y_{v_i} + z_{v_i}) = \sum_{i=1}^{k} (y_{v_{i-1}} + z_{v_i}) \tag{69}$$
$$\leqslant |A \cap \{(v_{i-1}, v_i) | i = 1, \ldots, k\}| = \lfloor \tfrac{1}{2} |V(C)| \rfloor.$$

'only if': Define a 'length' function $l: A \cup A^{-1} \to \mathbb{R}$ by:

$$\begin{aligned} l(v,w) &:= 1 - x_v \quad \text{if } (v,w) \in A, \\ l(v,w) &:= -x_v \quad \text{if } (v,w) \in A^{-1}. \end{aligned} \tag{70}$$

Note that each directed cycle C in the directed graph $(V, A \cup A^{-1})$ has nonnegative length $\Sigma_{a \in C} l(a)$, since

$$\sum_{a \in C} l(a) = \sum_{a \in C \cap A} l(a) + \sum_{a \in C \cap A^{-1}} l(a) = -\sum_{v \in V(C)} x_v + |C \cap A| \geqslant 0, \tag{71}$$

since $|C \cap A| \geqslant \lfloor \frac{1}{2} |V(C)| \rfloor$ by (66) and $\Sigma_{v \in V(C)} x_v \leqslant \lfloor \frac{1}{2} |V(C)| \rfloor$ as $x \in P$ (where $V(C) :=$ set of vertices in C).

Since each directed cycle in $(V, A \cup A^{-1})$ has nonnegative length, there exists a vector $z \in \mathbb{R}^V$ so that $z_w - z_v \leqslant l(v,w)$ for each $(v,w) \in A \cup A^{-1}$ (we could take $z_v :=$ the minimum length of any directed path in $(V, A \cup A^{-1})$ ending in v - then trivially $z_w \leqslant z_v + l(v,w)$ for each $(v,w) \in A \cup A^{-1}$). Hence

$$\begin{aligned} z_w - z_v &\leqslant 1 - x_v \quad \text{if } (v,w) \in A, \\ z_w - z_v &\leqslant -x_v \quad \text{if } (v,w) \in A^{-1}. \end{aligned} \tag{72}$$

Defining $y_v := x_v - z_v$ we obtain x, y, z satisfying (67). *End of proof of the Claim.*

Now let Q be the set of all vectors $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^V \times \mathbb{R}^V \times \mathbb{R}^V$ satisfying (67). Then Q is a polyhedron, and it is equal to the convex hull of the integral vectors in Q, i.e., $Q = Q_I$. This follows from the total unimodularity of the constraint matrix in (67), which is of type

$$\begin{bmatrix} I & I & I \\ 0 & M & N \end{bmatrix}, \tag{73}$$

where I is a $V \times V$-identity matrix, and where M and N are $\{0,1\}$-matrices so that every row of M and every row of N contains exactly one 1. By Theorem 5, matrix (73) is totally unimodular, and hence $Q = Q_I$.

Since by the Claim, P is a projection of Q, all vertices of P are integral. Hence, each vertex of P is the characteristic vector of some coclique of G, implying that P is the coclique polytope of G. □

Note that the inequalities (iii) in (65) are the cutting planes added to (i) and (ii). So the theorem states that, if G contains no odd-K_4 as a subgraph, then the coclique polytope is equal to $\{x \in \mathbb{R}_+^V | x_v + x_w \leqslant 1 \ (\{v,w\} \in E)\}'$.

With the help of Theorem 1 one can derive from Theorem 18 the polynomial-time solvability of the maximum-weighted coclique problem for graphs without odd-K_4. Indeed, one must show that (i), (ii) and (iii) in (65) can be checked in polynomial time. Conditions (i) and (ii) are easily checked one by one. Condition (iii) however consists of exponentially many inequalities. To check them in polynomial time, define a 'length' function $l:E \to \mathbb{R}_+$ by $l(e) := 1 - x_v - x_w$ if $e = \{v,w\}$. Then checking (iii) is equivalent to testing if each odd circuit has length at least 1. This last is not difficult to do in polynomial time, by adaptation of a shortest path algorithm.

GERARDS [8] also derived the following min-max relation.

THEOREM 19. *Let $G=(V,E)$ be a graph without isolated vertices, not containing an odd-K_4 as a subgraph. Then*

$$\max\{|C| \mid C \text{ coclique }\} = \min\{|F| + \sum_{i=1}^{t} \lfloor \tfrac{1}{2}|V(C_i)| \rfloor \mid F \subseteq E;\ C_1,\ldots,C_t \quad (74)$$

$$\text{odd circuits so that } V = \bigcup_{e \in F} e \cup \bigcup_{i=1}^{t} V(C_i)\}.$$

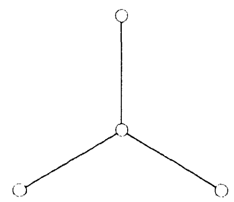
So the minimum ranges over all sets of edges and odd circuits which together cover all vertices of G. Note the similarity to Theorem 12.

The theorem means that if we write $\max\{|C| \mid C \text{ coclique}\}$ as the problem of maximizing $\mathbf{1}^T x$ over vectors x satisfying (65), we obtain a linear program with integral optimum primal *and* dual solutions.

FINAL REMARK. In this paper we saw the polynomial-time solvability of two combinatorial optimization problems:

(i) finding a maximum-weighted matching in a graph; (75)
(ii) finding a maximum-weighted coclique in a graph without odd-K_4.

In fact, a maximum-weighted matching in a graph H can be considered as a maximum-weighted coclique in the line-graph $L(G)$ of G. MINTY [17] and SBIHI [19] showed that more generally, the maximum-weighted coclique problem for *claw-free* graphs is polynomially solvable, i.e., for graphs not containing



as an *induced* subgraph. By Theorem 1 it implies that the separation problem for coclique polytopes of claw-free graphs is polynomially solvable. However, no explicit description by inequalities for these polytopes has been found. It has been shown by CHVÁTAL [3] that there exists no fixed t so that for claw-free graphs the coclique polytope is equal to $\{x \in \mathbb{R}_+^V | x_v + x_w \leqslant 1$ $(\{v,w\} \in E)\}^{(t)}$, in the notation of Section 7.

REFERENCES

1. C. BERGE (1958). Sur le couplage maximum d'un graphe. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 247,* 258-259.
2. V. CHVÁTAL (1973). Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics 4,* 305-337.
3. V. CHVÁTAL (1984). *Cutting-Plane Proofs and the Stability Number of a Graph,* Report No. 84326, Institut für Operations Research, Universität Bonn.
4. S.A. COOK (1971). The complexity of theorem-proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing,* The Association of Computing Machinery, New York, 151-158.
5. J. EDMONDS (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards (B) 69,* 125-130.
6. A. FRANK, É. TARDOS (1985). An application of simultaneous approximation in combinatorial optimization. *26th Annual Symposium on Foundations of Computer Science,* IEEE, New York, 459-463.
7. A.M.H. GERARDS. *Homomorphisms of Graphs into Odd Cycles,* preprint.
8. A.M.H. GERARDS, to appear.
9. A.M.H. GERARDS, A. SCHRIJVER. Matrices with the Edmonds-Johnson property, to appear in *Combinatorica.*
10. R.E. GOMORY (1958). Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society 64,* 275-278.
11. M. GRÖTSCHEL, L. LOVÁSZ, A. SCHRIJVER (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1,* 169-197.
12. M. GRÖTSCHEL, L. LOVÁSZ, A. SCHRIJVER (1986). *The Ellipsoid Method and Combinatorial Optimization,* Springer-Verlag, to appear.
13. A.J. HOFFMAN, J.B. KRUSKAL (1956). Integral boundary points of convex polyhedra. H.W. KUHN, A.W. TUCKER (eds.). *Linear Inequalities and Related Systems,* Princeton University Press, Princeton, N.J., 223-246.
14. R.M. KARP (1972). Reducibility among combinatorial problems. R.E. MILLER, J.W. THATCHER (eds.). *Complexity of Computer Computations,* Plenum Press, New York, 85-103.
15. L.G. KHACHIYAN (1979). A polynomial algorithm in linear programming (in Russian). *Doklady Akademii Nauk SSSR 244,* 1093-1096 (English translation: *Soviet Mathematics Doklady 20,* 191-194).
16. A.K. LENSTRA, H.W. LENSTRA, JR., L. LOVÁSZ (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen 261,* 515-534.

17. G.J. MINTY (1980). On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory (B) 28*, 284-304.
18. M.W. PADBERG, M.R. RAO (1982). Odd minimum cut-sets and *b*-matchings. *Mathematics of Operations Research 7*, 67-80.
19. N. SBIHI (1980). Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics 29*, 53-76.
20. A. SCHRIJVER (1980). On cutting planes. *Annals of Discrete Mathematics 9*, 291-296.
21. A. SCHRIJVER (1983). Short proofs on the matching polyhedron. *Journal of Combinatorial Theory (B) 34*, 104-108.
22. A. SCHRIJVER (1986). *Theory of Linear and Integer Programming*, Wiley, Chichester.
23. P.D. SEYMOUR (1977). The matroids with the max-flow min-cut property. *Journal of Combinatorial Theory (B) 23*, 189-222.
24. P.D. SEYMOUR (1980). Decomposition of regular matroids. *Journal of Combinatorial Theory (B) 28*, 305-359.
25. P.D. SEYMOUR (1981). Matroids and multicommodity flows. *European Journal of Combinatorics 2*, 257-290.
26. W.T. TUTTE (1947). The factorization of linear graphs. *Journal of the London Mathematical Society 22*, 107-111.
27. W.T. TUTTE (1958). A homotopy theorem for matroids I, II. *Transactions of the American Mathematical Society 88*, 144-174.