Discrete Comput Geom (2008) 39: 174–190
DOI 10.1007/s00454-008-9050-5

Generating All Vertices of a Polyhedron Is Hard

Leonid Khachiyan · Endre Boros · Konrad Borys · Khaled Elbassioni · Vladimir Gurvich

Received: 21 July 2005 / Revised: 8 June 2006

Abstract We show that generating all negative cycles of a weighted graph is a hard enumeration problem, in both the directed and undirected cases. More precisely, given a family of negative (directed) cycles, it is an NP-complete problem to decide whether this family can be extended or there are no other negative (directed) cycles in the graph, implying that (directed) negative cycles cannot be generated in polynomial output time, unless $P = \mathrm{NP}$. As a corollary, we solve in the negative two well-known generating problems from linear programming: (i) Given an infeasible system of linear inequalities, generating all minimal infeasible subsystems is hard. Yet, for generating maximal feasible subsystems the complexity remains open. (ii) Given a feasible system of linear inequalities, generating all vertices of the corresponding polyhedron is hard. Yet, in the case of bounded polyhedra the complexity

Communicated by Günter M. Ziegler.

This research was partially supported by the National Science Foundation (Grant IIS-0118635), and by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science. An extended abstract of this paper appears in the *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms*, Miami, Florida, January 22–24, 2006.

Our friend and colleague, Leo Khachiyan, passed away with tragic suddenness while we were preparing this manuscript.

E. Boros (✉) · K. Borys · V. Gurvich
RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA
e-mail: boros@rutcor.rutgers.edu

K. Borys
e-mail: kborys@rutcor.rutgers.edu

V. Gurvich
e-mail: gurvich@rutcor.rutgers.edu

K. Elbassioni
Department 1, Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany
e-mail: elbassio@mpi-sb.mpg

remains open. Equiva lently, the complexity of generating vertices and extreme rays of polyhedra remains open.

1 Introduction and Main Results

Let $G = (V, E)$ be a directed graph (digraph) and let $w\colon E \to \mathbb{R}$ be a real-valued weight function defined on its arcs. We call such a pair a *weighted digraph* and denote it by (G, w). For every subset of arcs $F \subseteq E$ its weight is defined as the total weight of all its arcs, $w(F) = \sum_{e \in F} w(e)$. We call a simple directed cycle a *circuit*. A circuit is called *negative* if its weight is negative. Finally, we denote by $\mathcal{C}^- = \mathcal{C}^-(G, w)$ the family of negative circuits of (G, w), i.e., $\mathcal{C}^- = \{C \subseteq E \mid C \text{ is a circuit with } w(C) < 0\}$.

First we consider the problem of generating exhaustively all negative circuits of a given weighted directed graph (G, w), in other words the problem of enumerating the family $\mathcal{C}^-(G, w)$. Since the number of negative circuits may be exponential in the size of the input description, i.e., the size of G and w, the efficiency of such enumeration algorithms is measured customarily in both the input and output sizes (see, e.g., [28, 32, 43]). More precisely, such an enumeration problem is said to be solvable in *polynomial total time* if the output can be generated in time polynomial in the input and output sizes. It is easy to see that for *self-reducible* (see, e.g., [29]) problems a family $\mathcal{C}$ is enumerable in polynomial total time if and only if for each subfamily $\mathcal{X} \subseteq \mathcal{C}$, the problem of deciding $\mathcal{X} \neq \mathcal{C}$; if yes, finding $C \in \mathcal{C} \setminus \mathcal{X}$, is solvable in time polynomial in size(G, w) and $|\mathcal{X}|$. On the other hand, when this decision problem is NP-hard, the enumeration problem is called NP-hard, too (see [32]). Thus, NP-hard enumeration problems are unlikely to have total polynomial time enumeration algorithms, unless $P = \text{NP}$.

Our main result claims that enumerating negative circuits of a weighted directed graph is a hard enumeration problem.

Theorem 1 *Given a weighted digraph $G = (V, E)$, $w\colon E \to \mathbb{R}$, and a family $\mathcal{X} \subseteq \mathcal{C}^-$ of its negative circuits, it is an NP-complete problem to decide whether $\mathcal{X} \neq \mathcal{C}^-$, even if w takes only two different values.*

We add that the analogous hardness result can be shown for undirected graphs, as well. In this case we also call a simple cycle a circuit and we denote by $\mathcal{C}^- = \mathcal{C}^-(G, w)$ the family of all negative circuits of an undirected graph $G = (V, E)$.

Theorem 2 *Given a weighted undirected graph $G = (V, E)$, $w\colon E \to \mathbb{R}$, and a family $\mathcal{X} \subseteq \mathcal{C}^-(G, w)$ of its negative circuits, it is an NP-complete problem to decide whether $\mathcal{X} \neq \mathcal{C}^-$, even if w takes only two different values.*

We remark that all circuits of a directed or undirected graph can be enumerated efficiently, e.g., by a simple backtracking algorithm [37].

Note that if w takes the same value for all edges (arcs), then negative circuits either do not exist or all circuits are negative. Thus, the enumeration problems for both directed and undirected graphs can be solved efficiently, as we noted earlier. Furthermore, when w takes only two different values, those can be assumed to be

integers, and hence by edge (arc) splitting the input can be transformed to one in which all edges (arcs) have weight ± 1. Though this transformation may increase the size of the input in a nonpolynomial way, in the case of the specific constructions we provide in the proofs of the above two theorems, it is a polynomial transformation, implying that generating all negative circuits is NP-hard even if all edges (arcs) have weights ± 1.

We derive several consequences of the above results, including the hardness of generating all vertices of a (possibly unbounded) polyhedron, generating all minimal infeasible subsystems of a system of linear inequalities, etc. We prove Theorems 1 and 2 in Sects. 2 and 3, respectively.

1.1 Negative Circuits and Minimal Infeasible Subsystems

We first note that deciding the existence and finding a negative circuit in a weighted directed graph are polynomially solvable tasks. Gallai [25] proved that (G, w) has no negative circuit if and only if by a potential transformation all edge weights can be changed to nonnegative values. Furthermore, a negative circuit can be found in $O(|V|^3)$ time, if the graph has negative circuits [23, 44]. We use Gallai's approach to reformulate the problem and derive some interesting consequences.

To a weighted digraph (G, w), where $G = (V, E)$ and $w\colon E \to \mathbb{R}$, we associate a polyhedron $P(E, w)$ defined by

$$P(E, w) = \left\{ x \in \mathbb{R}^V \;\middle|\; x_v - x_u \le w(u, v) \text{ for all arcs } (u, v) \in E \right\}. \tag{1}$$

Note that every vector $x \in P(E, w)$ is a potential in the sense Gallai [25] defined it, proving that G is negative circuit free. Namely, defining $w'(u, v) = w(u, v) + x_u - x_v$ for all arcs $(u, v) \in E$ we get another weighting of the arcs of G, such that $w'(C) = w(C)$ for all directed circuits $C \subseteq E$, and for which $w'(u, v) \ge 0$ for all arcs $(u, v) \in E$, according to the definition of $P(E, w)$. This latter shows that G is indeed negative cycle free.

Thus applying Gallai's result to subgraphs of G we obtain that $P(E', w) = \emptyset$ for some $E' \subseteq E$ if and only if the subgraph $G' = (V, E')$ contains a negative cycle with respect to the weight function w. Therefore, the minimal infeasible subsystems of the system of linear inequalities (1) correspond in a one-to-one way to the negative circuits of (G, w). Hence, Theorem 1 implies the following result.

Corollary 1 *Enumerating all minimal infeasible subsystems of a system of linear inequalities is an NP-hard enumeration problem, even if we restrict the input to linear systems involving at most two variables in each inequality.*

The problems of finding minimal infeasible subsystems of a system of linear inequalities, sometimes called *Irreducible Inconsistent Subsystems* (IIS) or *Helly systems*, and its natural dual of finding maximal feasible subsystems received ample attention in the literature, see, e.g., [5, 35, 38]. The optimization versions of these problems, i.e., finding a maximum cardinality feasible subsystem, and finding a minimum cardinality infeasible subsystem are known to be NP-hard, see, e.g., [14, 27, 35].

1.2 Minimal Infeasible Subsystems and Vertex Enumeration

Recall that the infeasibility of a system of linear inequalities is well characterized by the Farkas Lemma: either the system $Ax \geq b$ has a solution, or there exists a nonnegative vector $y \geq 0$ such that $y^T A = 0$ and $y^T b > 0$, but not both (see [21]). Using this claim, Gleeson and Ryan [26] associated to a system of linear inequalities $Ax \geq b$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, a so-called *alternative polyhedron* defined as $Q = \{y \in \mathbb{R}^m_+ \mid y^T A = 0,\ y^T b = 1\}$, and observed that minimal infeasible subsystems of $Ax \geq b$ are in a one-to-one correspondence with vertices of Q. Indeed, for every vector $y \in Q$ we consider the subsystem of $Ax \geq b$ corresponding to the support set $S(y) = \{i \mid y_i \neq 0\}$. By the Farkas Lemma, we have that these corresponding subsystems are indeed infeasible. Conversely, if S is the index set of an infeasible subsystem of $Ax \geq b$, then again by Farkas's lemma we have a vector $y \in Q$ for which $S(y) \subseteq S$. Thus, minimal infeasible subsystems correspond to vectors $y \in Q$ with minimal support sets, and hence those are indeed vertices of Q.

This observation, coupled with Corollary 1, implies the hardness of enumerating the vertices of polyhedra.

Corollary 2 *Enumerating all vertices of a rational polyhedron, given as the intersection of finitely many closed half-spaces, is an NP-hard enumeration problem.*

Proof We consider an infeasible system of rational linear inequalities $Ax \geq b$, and its alternative polyhedron Q. We can write Q equivalently as $Q = \{y \in \mathbb{R}^m \mid y \geq 0,\ A^T y \geq 0,\ -A^T y \geq 0,\ b^T y \geq 1,\ -b^T y \geq -1\}$, i.e., as the intersection of $m + 2n + 2$ closed half-spaces. Thus, by the above observation, enumerating the vertices of this rational polyhedron would also enumerate all minimal infeasible subsystems of $Ax \geq b$, which is an NP-hard enumeration problem according to Corollary 1. □

Vertex enumeration is a fundamental problem in computational geometry and polyhedral combinatorics (see, e.g., [19] for a list of applications), and has many equivalent formulations. Most notably for bounded polyhedra, vertex enumeration is equivalent with *facet generation*, i.e., enumerating the facets of a polytope given by an explicit list of its vertices (see, e.g., the so-called polytope–polyhedron problem in [31]).

We add that in this paper we consider polyhedra which have vertices. This condition is easy to check in polynomial time and does not restrict generality. We emphasize that whenever the system of equations $A^T y = 0$, $b^T y = 0$ has a nontrivial solution for which $y \geq 0$, then Q in Corollary 2 is an unbounded polyhedron. Thus, our reduction through Theorem 1 yields in general unbounded polyhedra, and hence does not imply the hardness of vertex generation for bounded polyhedra, which remains an open problem. Furthermore, and equivalently, the complexity of enumerating together vertices and extreme rays of polyhedra is also an open problem (any unbounded polyhedron P can be projectively transformed into a bounded polyhedron, by adding one "far face," whose vertices correspond to the extreme rays of P, see, e.g., [35]).

Numerous algorithmic ideas have been introduced in the literature (either for vertex or for facet enumeration, see e.g., [1, 3, 4, 6, 9, 10, 12, 15, 16, 18, 19, 33, 34, 36, 41, 42]). Efficient algorithms (typically linear in the number of vertices) were

proposed for several special cases, including simple polyhedra, i.e., in which every vertex is incident with exactly n facets [3], simplicial polyhedra, which are the dual of simple polyhedra [9], network polytopes [36], polytopes with zero–one vertices [10], and polyhedra in which every facet defining inequality involves at most two nonzero coefficients [1]. Furthermore, for fixed dimension both vertices and rays of a polyhedron can be enumerated efficiently [13]. However, no method proved to be efficient (yet) for the general case. In fact, several publications [2, 11, 24] analyzed the proposed general-purpose methods for vertex/facet enumeration, and showed that all of the known algorithms may require in the worst case superpolynomial time in the output size. Along the same lines, Corollary 2 shows that vertex enumeration is indeed a hard enumeration problem for unbounded polyhedra (unless of course $P = \mathrm{NP}$).

In analyzing the reasons why backtracking methods are not efficient for vertex enumeration, in general, Fukuda et al. [24] noted that such methods require repeatedly solving decision problems, which turn out to be NP-hard. In particular, they showed that for a given rational polyhedron P and an open rational half-space $H = \{x \in \mathbb{R}^n \mid \alpha^T x > \beta\}$, it is NP-hard to decide if P has a vertex in H. We note that the same decision problem for bounded polyhedra is much easier, since it can be decided by maximizing $\alpha^T x$ over P, which is a linear programming problem, known to be polynomially solvable, see Khachiyan [30]. We can show, as a next corollary of Theorem 1, that the enumerative version of this decision problem is hard for bounded polyhedra.

To arrive at this claim, we recall that the vertices of the *circulation polytope*

$$P(G) = \left\{ y \in \mathbb{R}^E \;\middle|\; \begin{array}{ll} \sum_{v:\,(u,v)\in E} y_{uv} - \sum_{w:\,(w,u)\in E} y_{wu} = 0, & \forall u \in V, \\ \sum_{(u,v)\in E} y_{uv} = 1, & \\ 0 \le y_{uv}, & \forall (u,v) \in E \end{array} \right\}$$

of a directed graph $G = (V, E)$ correspond to circuits of G, namely for every vertex y of $P(G)$ its support set $S(y) = \{(u, v) \in E \mid y_{uv} \neq 0\}$ is a circuit in G.

We remark that $P(G)$ frequently occurs in the optimization literature under various names, e.g., as the trans-shipment or flow polyhedron, or simply as the set of feasible circulations, or feasible solutions to a trans-shipment problem, etc. (see, e.g., Chaps. 11–13 in [40]). The vertices and facial structure of $P(G)$ are well studied and understood. In particular, the vertices of $P(G)$ can be generated in linear (output) time by cycle enumeration [37].

Associating further to a rational weight function $w\colon E \to \mathbb{R}$ an open rational half-space defined by

$$H = \left\{ y \in \mathbb{R}^E \;\middle|\; \sum_{(u,v)\in E} w(u,v) y_{uv} < 0 \right\},$$

we get that the support sets of vertices of $P(G)$ belonging to H are exactly the negative circuits of the weighted directed graph (G, w). Thus, Theorem 1 readily implies the following claim.

Corollary 3 *Given a rational polyhedron P and an open rational half-space H, it is NP-hard to enumerate all vertices of P which belong to H, even if P is bounded.*

Many applications (see, e.g., [19]) call for the enumeration of all those basic feasible solutions to a linear programming problem (i.e., vertices of the corresponding polyhedron), the corresponding objective function value of which is above a given threshold. Corollary 3 indicates that unfortunately such enumeration problems are difficult in general, unless $P = \mathrm{NP}$.

A further consequence of Theorem 1 is that enumerating all vertices of a bounded polyhedron P which do not belong to a given face of P is also hard, in general.

Corollary 4 *Given a bounded polyhedron P and a proper face F of it, it is NP-hard to enumerate the vertices of P which do not belong to F.*

Proof Let $\bar{H} = \{y \in \mathbb{R}^E \mid \sum_{(u,v)\in E} w(u,v)y_{uv} \leq 0\}$. Note that $P' = P(G) \cap \bar{H}$ is a bounded polyhedron, for which $\bar{H}$ is facet defining. Denoting this facet by F, the vertices of P' outside F correspond in a one-to-one way to the negative circuits of the weighted graph (G, w) to which we associated H and $P(G)$. Thus, the claim follows from Theorem 1. □

By Corollary 2 unless $P = \mathrm{NP}$ there exists no algorithm that outputs in incremental (or total) polynomial time, the vertices and then the extreme directions of a polyhedron, in that order. In contrast we have the following statement.

Proposition 1 *If there exists an algorithm which enumerates all vertices of a bounded polyhedron in incremental polynomial time, then we can enumerate all extreme rays and then all vertices (in this order) of a polyhedron in incremental polynomial time.*

Proof Let $P = \{x \in \mathbb{R}^n\colon a_i^T x \leq b_i, i = 1, \ldots, m\}$ be an unbounded polyhedron and let V and R denote the set of vertices and the set of extreme rays of P, respectively. As before, we can assume that V contains at least one vertex v. Let $a = \sum_{i\colon a_i^T v = b_i} a_i$. Then $P' = \{x\colon a_i^T x \leq b_i, i = 1, \ldots, m, a^T x = -M\}$ is a bounded polyhedron whose vertices correspond to R, where M is an appropriately large constant. Furthermore, $P'' = \{x\colon a_i^T x \leq b_i, i = 1, \ldots, m, a^T x \geq -M\}$ is a bounded polyhedron whose vertices correspond to $V \cup R$.

Assuming the existence of an algorithm **A** that can enumerate all vertices of a bounded polyhedron in incremental polynomial time, it follows that for any given subset W of the vertices of that bounded polyhedron, we can decide if this subset contains all vertices or, if not, can generate a vertex not belonging to W, in time, polynomial in the size of the input description of the polyhedron and the set W of given vertices. This can be accomplished simply by running **A** until it stops, or it outputs $|W| + 1$ vertices, whichever happens earlier.

Thus, by first applying **A** to P' we can generate the set R incrementally efficiently. Furthermore, since R is a subset of the vertices of P'', we can continue by applying **A** to P'' and extend in this way the set R incrementally efficiently to $V \cup R$, as we described earlier. Hence, we can enumerate the set $V \cup R$ in the stated order, first R and then V, incrementally efficiently. □

1.3 Four Geometric Enumeration Problems

We finally recall four strongly related geometric enumeration problems. Let $\mathcal{A} \subseteq \mathbb{R}^n$ be a given subset of vectors in $\mathbb{R}^n$, fix a point $z \in \mathbb{R}^n$ called the *center*, and consider the following four definitions:

- A *simplex* is a minimal subset $X \subseteq \mathcal{A}$ containing the center in its convex hull, i.e., $z \in \text{conv}(X)$.
- An *anti-simplex* is a maximal subset $X \subseteq \mathcal{A}$ not containing the center in its convex hull, i.e., $z \notin \text{conv}(X)$.
- A *body* is a minimal (full-dimensional) subset $X \subseteq \mathcal{A}$ containing the center in the interior of its convex hull, i.e., $z \in \text{int}(\text{conv}(X))$.
- An *anti-body* is a maximal subset $X \subseteq \mathcal{A}$ not containing the center in the interior of its convex hull, i.e., $z \notin \text{int}(\text{conv}(X))$.

Equivalently, a simplex (body) is a minimal collection of the given vectors not contained in an *open* (*closed*) half-space through the center, while an anti-simplex (anti-body) is a maximal collection of vectors contained in an open (closed) half-space through the center. It can be seen easily that $|X| \leq n+1$ for a simplex, and that $n+1 \leq |X| \leq 2n$ for a body.

In what follows we assume that the center is at the origin, i.e., $z = 0$. For a given point set $\mathcal{A} \subseteq \mathbb{R}^n$ we denote, respectively, by $\mathcal{S}$ and $\mathcal{B}$ the hypergraphs on the base set $\mathcal{A}$, consisting of all simplices, and all bodies of $\mathcal{A}$. The corresponding families of maximal independent sets of these two hypergraphs are, respectively, all anti-simplices and anti-bodies of $\mathcal{A}$, denoted respectively by $\mathcal{S}^*$ and $\mathcal{B}^*$, i.e.,

$$\mathcal{S}^* = \{X \subseteq \mathcal{A} \mid X \text{ is maximal such that } X \not\supseteq S,\ \forall S \in \mathcal{S}\},$$

$$\mathcal{B}^* = \{Y \subseteq \mathcal{A} \mid Y \text{ is maximal such that } Y \not\supseteq B,\ \forall B \in \mathcal{B}\}.$$

Simplices, anti-simplices, bodies, and anti-bodies can naturally be related to minimal infeasible or maximal feasible subsystems of certain linear systems of inequalities. Namely, we denote by $A \in \mathbb{R}^{m \times n}$, where $m = |\mathcal{A}|$, the matrix whose row vectors are the vectors of $\mathcal{A}$, and we let $e \in \mathbb{R}^m$ denote the m-dimensional vector of all ones.

It follows from the above definitions that simplices and anti-simplices are in a one-to-one correspondence, respectively, with the minimal infeasible and maximal feasible subsystems of the linear system of inequalities:

$$Ax \geq \mathbf{e}, \quad x \in \mathbb{R}^n. \tag{2}$$

Similarly, it follows that bodies and anti-bodies correspond in a one-to-one way, respectively, to the minimal infeasible and maximal feasible subsystems of the system:

$$Ax \geq 0, \quad x \neq 0. \tag{3}$$

As for the complexity of these enumeration problems, it is known that the generation of anti-bodies is a hard problem:

Proposition 2 [7] *Given a set of vectors $\mathcal{A} \subseteq \mathbb{R}^n$, and a partial list $\mathcal{X} \subseteq \mathcal{B}^*$ of the anti-bodies of $\mathcal{A}$, it is NP-hard to determine if the given list is incomplete, i.e.,*

$\mathcal{X} \neq \mathcal{B}^*$, or not. Equivalently, given an infeasible system (3)*, and a partial list of its maximal feasible subsystems, it is NP-hard to determine if the given partial list is incomplete or not.*

Enumeration of bodies turns out to be at least as hard as the well-known *hypergraph transversal problem* [8] whose exact complexity is still an outstanding open problem [20]. The best currently known algorithm for the hypergraph transversal problem runs in incremental *quasi-polynomial* time [22].

Proposition 3 [7] *The problem of incrementally enumerating bodies, for a given set of $m+n$ points $\mathcal{A} \subseteq \mathbb{R}^n$, includes as a special case the problem of enumerating all minimal transversals for a given hypergraph $\mathcal{H}$ with n hyperedges on m vertices. Equivalently, generating minimal infeasible subsystems of* (3) *is at least as hard as hypergraph transversal generation.*

The problem of generating simplices turns out to be equivalent, in general, to the problem of enumerating the vertices of bounded polyhedra, or enumerating the vertices and extreme rays of possibly unbounded polyhedra. To see this, we consider a vector set $\mathcal{A} = \{a_1, \ldots, a_n, b\} \subseteq \mathbb{R}^d$ and associate to it a polyhedron $P = \{x \in \mathbb{R}^n \mid Ax = -b, x \geq 0\}$, where $A = [a_1, \ldots, a_n]$ is the matrix with columns $a_1, \ldots, a_n$.

Recall that for a vector $y \in \mathbb{R}^n$ we called the set $S(y) = \{i \mid y_i \neq 0\}$ its support set.

Proposition 4 *If $y \in P$ is a vertex of P, then the set $\{a_i \mid i \in S(y)\} \cup \{b\}$ is a simplex of $\mathcal{A}$, while if $y \in P$ is an extreme ray of P, then the set $\{a_i \mid i \in S(y)\}$ is a simplex of $\mathcal{A}$. Furthermore, every simplex of $\mathcal{A}$ corresponds in this way either to a vertex or to an extreme ray of P.*

Proof It is well known that the vertices of P are the solutions which have minimal support sets, and the extreme rays are those solutions of the homogenized system (replace b by 0) which have minimal support sets (see, e.g., Chap. 8 in [39]). Clearly, the minimality of support sets in both cases implies the first two claims, by the definition of a simplex of $\mathcal{A}$.

For the last claim, let $S \subseteq \mathcal{A}$ be a simplex, i.e., a minimal subset for which $0 \in \operatorname{conv}(S)$. If $b \in S$, then we have for some $\lambda_a \geq 0$, $a \in S \backslash \{b\}$, and $\lambda_b \geq 0$, with $\lambda_b + \sum_{a \in S\backslash\{b\}} \lambda_a = 1$, that

$$-\lambda_b b = \sum_{a \in S\backslash\{b\}} \lambda_a a.$$

Since S is minimal, we must have all these coefficients positive, and thus

$$-b = \sum_{a \in S\backslash\{b\}} \frac{\lambda_a}{\lambda_b} a.$$

Thus, the vector $x \in \mathbb{R}^n$, defined by

$$x_i = \begin{cases} \lambda_{a_i}/\lambda_b & \text{if } a_i \in S\backslash\{b\}, \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, n$, is a vertex of P, again by the minimality of S. While if $b \notin S$, then we have

$$0 = \sum_{a \in S} \lambda_a a$$

for some positive coefficients $\lambda_a > 0$, $a \in S$, for which $\sum_{a \in S} \lambda_a = 1$, and thus the vector $x \in \mathbb{R}^n$, defined by

$$x_i = \begin{cases} \lambda_{a_i} & \text{if } a_i \in S, \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, n$, is an extreme ray of P, once more by the minimality of S. □

In particular, if $P = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$ is a bounded polyhedron, i.e., if $Ax = 0$ has no nontrivial nonnegative solutions, then the vertices of P correspond in a one-to-one way to the simplices of the set $\mathcal{A}$ formed by the column vectors of A and b.

For the special case of vectors $\mathcal{A} \subseteq \mathbb{R}^n$ in general position, we have $\mathcal{B} = \mathcal{S}$, and consequently the problem of enumerating bodies of $\mathcal{A}$ turns into the problem of enumerating vertices of the bounded polyhedron $\{x \in \mathbb{R}^n \mid Ax = 0, \mathbf{e}^T x = 1, x \geq 0\}$, each vertex of which is nondegenerate and has exactly $n + 1$ positive components. For such kinds of *simple* bounded polyhedra there exist algorithms that generate all vertices with polynomial delay (see e.g., [15] and [3]).

We finally mention that, although the status of the problem of enumerating all maximal feasible subsystems of (2) is not known in general, the situation changes if we fix a consistent subfamily of inequalities, and ask for enumerating all its extensions to a maximal feasible subsystem. In fact, such a problem turns out to be NP-hard, even if we fix only nonnegativity constraints.

Proposition 5 [7] *Let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix, let $b \in \mathbb{R}^m$ be an m-dimensional vector, and assume that the system*

$$Ax \geq b, \quad x \in \mathbb{R}^n, \tag{4}$$

has no solution $x \geq 0$. Let $\mathcal{F}$ be the family of all maximal subsystems of (4) which can be satisfied by a nonnegative solution x. Then, given a partial list $\mathcal{X} \subseteq \mathcal{F}$, it is an NP-complete problem to determine if the list is incomplete, i.e., if $\mathcal{X} \neq \mathcal{F}$, even if b is a unit vector, and entries in A are either, -1, 1, or 0.

We conclude with the observation that the problem of finding, for an infeasible system

$$A'x \geq b', \qquad A''x \geq b'', \tag{5}$$

all maximal feasible subsystems extending the feasible subsystem $A''x \geq b''$, naturally includes both problems of generating anti-simplices and simplices. Clearly, the former problem can be written in the form (5) by considering (2) and all maximal extensions of an empty subsystem. For the latter problem, note that the vertices of a bounded polyhedron $\{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$, where $b \neq 0$, are in one-to-one correspondence with the maximal feasible extensions of the subsystem $Ax = b$, $x \geq 0$

in the infeasible system $Ax = b$, $x \geq 0$, $x \leq 0$. Although the general problem of generating maximal feasible extensions is NP-hard as stated above, the special cases of generating simplices and anti-simplices remain open.

2 Proof of Theorem 1

In this section we prove Theorem 1 by a reduction from satisfiability, a well-known NP-complete problem (see [17]).

We consider n propositional Boolean variables X_j, $j = 1, \ldots, n$, we denote by $\overline{X} = 1 - X$ the negation of X, we call variables and their negations *literals*, and elementary disjunctions of literals *clauses*. We next consider an arbitrary conjunctive normal form (CNF) $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, i.e., where C_i, $i = 1, \ldots, m$, are clauses. A truth assignment to the variables is called *satisfying* for the CNF ϕ, if ϕ evaluates to true, i.e., if at least one literal evaluates to true in each of the clauses of ϕ.

In what follows we associate to ϕ a weighted directed graph (G, w) and a set $\mathcal{X}$ of negative circuits of G such that (G, w) has a negative circuit not belonging to $\mathcal{X}$ if and only if ϕ has a satisfying assignment. Because (G, w) and $\mathcal{X}$ are constructed from ϕ in $O(mn)$ time, and the weight function w uses only two different values (1 and -1), Theorem 1 follows readily from this construction. This is because the decision problem "*Is there a negative circuit in* (G, w) *which does not belong to* $\mathcal{X}$*?*" is in NP. To complete the proof of Theorem 1, we provide in the following a construction with these properties, such that every satisfying assignment to ϕ corresponds to a negative circuit of (G, w) not belonging to $\mathcal{X}$ and, vice versa, every negative circuit of (G, w) which does not belong to $\mathcal{X}$ corresponds to a satisfying assignment of ϕ (though the correspondence is not necessarily one-to-one).

To describe our construction, we denote for $j = 1, \ldots, n$, respectively by o_j and $\bar{o}_j$, the number of occurrences of literal X_j and its negation $\overline{X}_j$; we denote by x_j^k the kth occurrence of X_j, $k = 1, \ldots, o_j$, and by $\bar{x}_j^k$ the kth occurrence of $\overline{X}_j$, $k = 1, \ldots, \bar{o}_j$, and let L denote the set of all literal occurrences, i.e.,

$$|L| = \sum_{i=1}^{m} |C_i| = \sum_{j=1}^{n} (o_j + \bar{o}_j).$$

Since monotone variables, i.e., ones for which $o_j = 0$ or $\bar{o}_j = 0$, can be easily eliminated from a satisfiability problem, we can assume without any loss of generality that $o_j > 0$ and $\bar{o}_j > 0$ hold for all variables $j = 1, \ldots, n$.

For instance, if $n = 3$ and

$$\phi = \left(X_1 \vee X_2 \vee \overline{X}_3\right) \wedge \left(X_1 \vee \overline{X}_2 \vee X_3\right) \wedge \left(\overline{X}_1 \vee X_2 \vee \overline{X}_3\right), \tag{6}$$

then we have $o_1 = 2$, $\bar{o}_1 = 1$, $o_2 = 2$, $\bar{o}_2 = 1$, $o_3 = 1$, $\bar{o}_3 = 2$, and

$$L = \left\{x_1^1, x_2^1, \bar{x}_3^1, x_1^2, \bar{x}_2^1, x_3^1, \bar{x}_1^1, x_2^2, \bar{x}_3^2\right\}.$$

We define the vertex set of the graph $G = (V, E)$ associated to ϕ as

$$V = U \cup Q \cup \bigcup_{j=1}^{n} (Y_j \cup Z_j),$$

where U, Q, and Y_j and Z_j for $j = 1, \ldots, n$ are pairwise disjoint, defined as

$$U = \{u_k \mid k = 0, 1, \ldots, m+n\},$$
$$Q = \big\{a(\ell), b(\ell) \bigm| \ell \in L\big\},$$
$$Y_j = \{y_{jk} \mid k = 1, \ldots, o_j - 1\} \quad \text{for } j = 1, \ldots, n, \quad \text{and}$$
$$Z_j = \{z_{jk} \mid k = 1, \ldots, \bar{o}_j - 1\} \quad \text{for } j = 1, \ldots, n.$$

The graph itself has a ring structure, the skeleton of which is the set U. For every variable X_j of ϕ we have two parallel directed paths from u_{j-1} to u_j. The first path corresponding to X_j contains vertices Y_j (and some other vertices), while the second path, corresponding to $\overline{X}_j$, passes through vertices of Z_j ($j = 1, \ldots, n$). For convenience, we also introduce the notation

$$y_{j0} = z_{j0} = u_{j-1} \quad \text{and} \quad y_{j,o_j} = z_{j,\bar{o}_j} = u_j \tag{7}$$

for $j = 1, \ldots, n$. To every clause C_i of ϕ we associate $|C_i|$ parallel directed paths from u_{n+i-1} to u_{n+i}, one for each of the literals in C_i ($i = 1, \ldots, m$). Finally vertices $a(\ell)$ and $b(\ell)$ correspond exclusively to literal occurrence $\ell \in L$.

We consider next the weighted graph $H(a, b, p, q, r, s)$ (see Fig. 1) on six nodes a, b, p, q, r, and s, having six arcs, the weights of which are as follows:

$$\begin{aligned} &w(a, b) = w(b, a) = -2 \quad \text{and} \\ &w(p, a) = w(b, q) = w(r, b) = w(a, s) = 1. \end{aligned} \tag{8}$$

To every literal occurrence $\ell \in L$ we associate a disjoint copy of $H(a, b, p, q, r, s)$, and denote by $a(\ell)$, $b(\ell)$, etc., its nodes, and by E_ℓ its arc set. Note that each of these small subgraphs can be decomposed into two directed paths, each consisting of three arcs, $E_\ell = E_\ell^v \cup E_\ell^c$, where

$$E_\ell^v = \big\{\big(p(\ell), a(\ell)\big), \big(a(\ell), b(\ell)\big), \big(b(\ell), q(\ell)\big)\big\}, \quad \text{and}$$
$$E_\ell^c = \big\{\big(r(\ell), b(\ell)\big), \big(b(\ell), a(\ell)\big), \big(a(\ell), s(\ell)\big)\big\}.$$

Finally we set

$$E = E_0 \cup \bigcup_{\ell \in L} E_\ell,$$

where $E_0 = \{(u_{m+n}, u_0)\}$ with weight $w(u_{m+n}, u_0) = -1$.

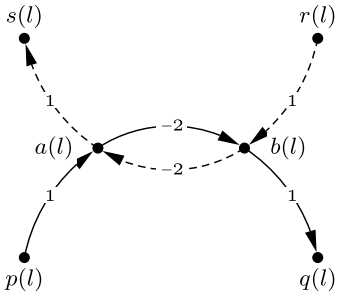

Fig. 1 The directed graph $H(a, b, p, q, r, s)$ associated with literal occurrences

In each of the subgraphs corresponding to the literal occurrences $\ell \in L$, we have the nodes $a(\ell)$ and $b(\ell)$ already introduced in $Q \subseteq V$, while the nodes $p(\ell)$, $q(\ell)$, $r(\ell)$, and $s(\ell)$ for $\ell \in L$ are corresponding to some other vertices of G, according to the following definitions:

$$p(\ell) = y_{j,k-1} \quad \text{and} \quad q(\ell) = y_{jk} \quad \text{if } \ell = x_j^k,$$
$$p(\ell) = z_{j,k-1} \quad \text{and} \quad q(\ell) = z_{jk} \quad \text{if } \ell = \bar{x}_j^k, \quad \text{and}$$
$$r(\ell) = u_{n+i-1} \quad \text{and} \quad s(\ell) = u_{n+i} \quad \text{if } \ell \in C_i.$$

In other words, for every literal occurrence ℓ of clause C_i the set E_ℓ^c forms a three-arc directed path from u_{n+i-1} to u_{n+i}. Furthermore, by (7) and by the above definitions, the sets E_ℓ^v for $\ell = x_j^1, x_j^2, \ldots, x_j^{o_j}$ form a directed path from u_{j-1} to u_j through the vertices of Y_j, consisting of $3o_j$ arcs, for every variable X_j. Similarly, the sets E_ℓ^v for $\ell = \bar{x}_j^1, \bar{x}_j^2, \ldots, \bar{x}_j^{\bar{o}_j}$ form another directed path from u_{j-1} to u_j through the vertices of Z_j, consisting of $3\bar{o}_j$ arcs.

In summary, $G = (V, E)$ consists of $|V| = 3|L| + m - n + 1$ vertices and $|E| = 6|L| + 1$ arcs, and the weight function w takes only values in $\{-2, -1, 1\}$. Note that we can split arcs of weight -2 to obtain a graph whose arcs all have weight ± 1.

Returning to the example CNF ϕ given in (6), the corresponding graph $G = (V, E)$ is shown in Fig. 2. To make the drawing of such a graph visually more clear, for every literal occurrence ℓ nodes $a(\ell)$ and $b(\ell)$ of G are represented by two separate points of the picture each, labeled as $a(\ell)$ and $a'(\ell)$, and as $b(\ell)$ and $b'(\ell)$, respectively. Similarly, node u_n is represented by two points in the figure, labeled u_n and u'_n. Arcs

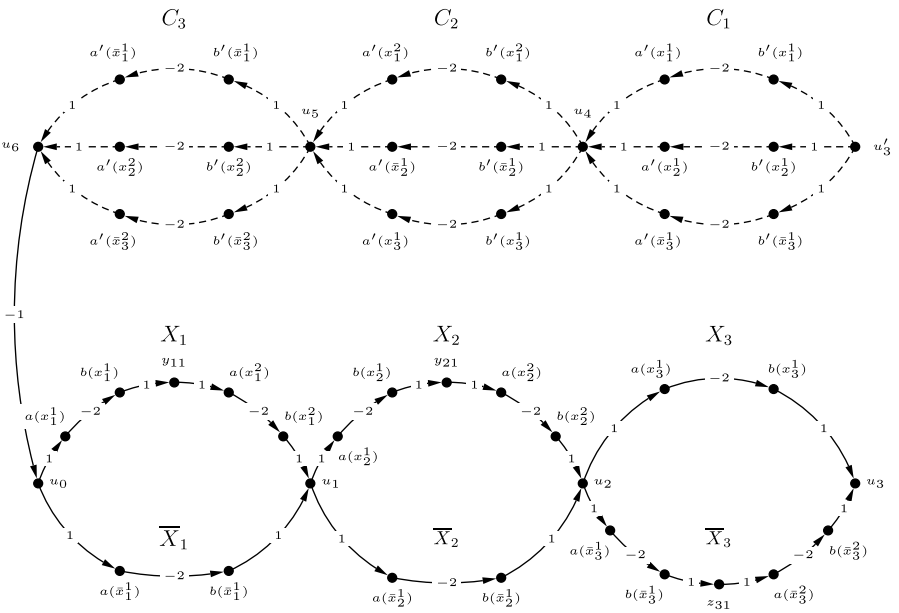

Fig. 2 G is obtained by identifying vertices $a(l)$, $a'(l)$, and $b(l)$, $b'(l)$, for each literal occurrence l, and u_3, u'_3 in the graph above. The *lower part* of the graph corresponds to the literals and the *upper part* corresponds to the clauses

in the sets E^c_ℓ for $\ell \in L$ are drawn as dashed lines, while those belonging to E^v_ℓ for $\ell \in L$ are drawn as solid lines.

Observe first that the arcs $(a(\ell), b(\ell))$ and $(b(\ell), a(\ell))$ form a circuit of total weight -4 for every literal occurrence $\ell \in L$. We denote by $\mathcal{X}$ the set of these circuits, i.e., $|\mathcal{X}| = |L|$, and we denote by $\mathcal{F}$ the set of all directed negative circuits of G.

We claim that from every satisfying assignment X of ϕ we can construct a directed negative circuit $D^X \in \mathcal{F}\backslash\mathcal{X}$ and, conversely, from every directed negative circuit $D \in \mathcal{F}\backslash\mathcal{X}$ we can construct a satisfying assignment X^D of ϕ. As we noted at the beginning of this section, this claim implies Theorem 1.

To see this claim, we first consider a satisfying assignment $X = (X_1, \ldots, X_n) \in \{0,1\}^n$ of ϕ. Since X satisfies ϕ, we have a literal occurrence ℓ_i in every clause C_i, $i = 1, \ldots, m$, such that ℓ_i evaluates to true at X (i.e., $\ell_i(X) = 1$). We also denote by W the set of all those literal occurrences which evaluate to false at X, i.e., $W = \{\ell \in L \mid \ell(X) = 0\}$. Clearly, $\ell_i \notin W$ for $i = 1, \ldots, m$ by the above definitions. Then the set of arcs

$$D^X = \left(\bigcup_{i=1}^{m} E^c_{\ell_i}\right) \cup \left(\bigcup_{\ell \in W} E^v_\ell\right) \cup \{(u_{m+n}, u_0)\}$$

forms a circuit in G not belonging to $\mathcal{X}$. Since we have $w(E^c_\ell) = w(E^v_\ell) = 0$ for all literal occurrences $\ell \in L$, it follows by the above definitions that $w(D^X) = w(u_{m+n}, u_0) = -1$, i.e., $D^X \in \mathcal{F}\backslash\mathcal{X}$ as claimed.

We again return to the CNF ϕ given in (6). We consider the satisfying assignment $X = (1, 0, 0)$ of ϕ. We choose literal occurrences $\bar{x}^1_3 \in C_1$, $\bar{x}^2_1 \in C_2$, and $\bar{x}^2_3 \in C_3$ that evaluate to true at X. Figure 3 depicts the negative circuit $D^X = E^c_{\bar{x}^1_3} \cup E^c_{\bar{x}^2_1} \cup E^c_{\bar{x}^2_3} \cup E^v_{\bar{x}^1_1} \cup E^v_{\bar{x}^1_2} \cup E^v_{\bar{x}^2_2} \cup E^v_{\bar{x}^1_3} \cup (u_6, u_0)$.

Before proving the reverse direction of our main claim, we first observe some simple properties of our construction. To simplify notation, recall that $E_\ell = E^c_\ell \cup E^v_\ell$ for $\ell \in L$, and that the six-vertex subgraphs induced by the arc set E_ℓ have the same structure and weights, as in Fig. 1, for all $\ell \in L$. The following property of these subgraphs are instrumental in our proof.

Lemma 1 *Given a circuit $D \subseteq E$ of G, not belonging to $\mathcal{X}$, and given a literal occurrence $\ell \in L$, we have*

$$w(D \cap E_\ell) \in \{0, 2, 4\}.$$

Moreover, $w(D \cap E_\ell) = 0$ only if the set $D \cap E_\ell$ is one of the following three subsets of E_ℓ: E^c_ℓ, E^v_ℓ, or $\emptyset$.

Proof Since D is a circuit not belonging to $\mathcal{X}$, D cannot contain both arcs $(a(\ell), b(\ell))$ and $(b(\ell), a(\ell))$. Thus, denoting $A_\ell = \{(p(\ell), a(\ell)), (a(\ell), s(\ell))\}$ and $B_\ell = \{(r(\ell), b(\ell)), (b(\ell), q(\ell))\}$ we have that $D \cap E_\ell$ is one of the following six sets: $\emptyset$, A_ℓ, B_ℓ, $A_\ell \cup B_\ell$, E^c_ℓ, and E^v_ℓ. Since we have $w(\emptyset) = w(E^c_\ell) = w(E^v_\ell) = 0$, $w(A_\ell) = w(B_\ell) = 2$, and hence $w(A_\ell \cup B_\ell) = 4$, the statement follows. □

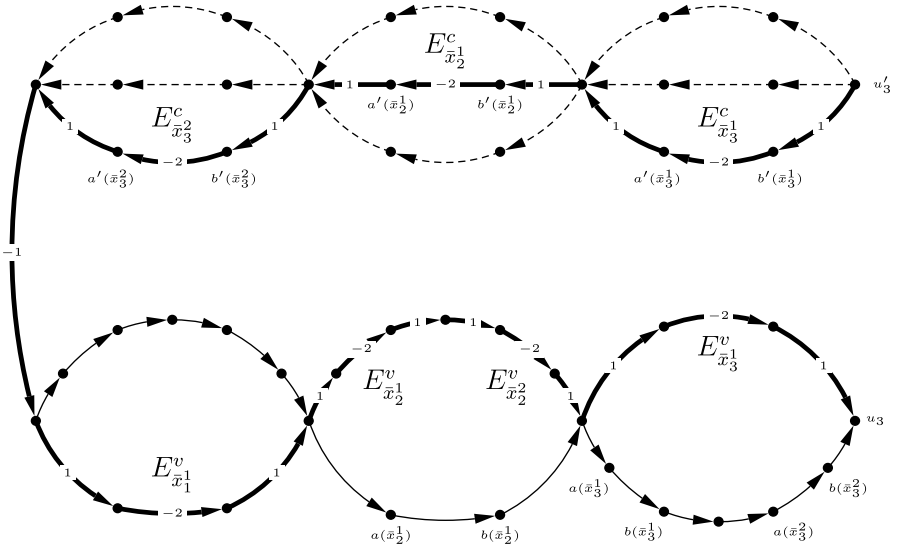

Fig. 3 *Thick lines* are edges of the negative circuit D^X corresponding to the satisfying assignment $X = (1, 0, 0)$. The vertices u_3, u'_3 are identified. Since D^X contains arcs $(b'(\bar{x}^1_3)a'(\bar{x}^1_3))$, $(b'(\bar{x}^1_2)a'(\bar{x}^1_2))$, and $(b'(\bar{x}^2_3)a'(\bar{x}^2_3))$ but it does not contain arcs $(a(\bar{x}^1_3)b(\bar{x}^1_3))$, $(a(\bar{x}^1_2)b(\bar{x}^1_2))$, and $(a(\bar{x}^2_3)b(\bar{x}^2_3))$, no circuit of $\mathcal{X}$ is contained in D^X

Returning to the reverse direction of our main claim, we consider a negative circuit $D \in \mathcal{F}\backslash\mathcal{X}$ of G. Since

$$w(D) = \sum_{\ell \in L} w(D \cap E_\ell) + w\big(D \cap \{(u_{m+n}, u_0)\}\big)$$

we must have by Lemma 1 that $(u_{m+n}, u_0) \in D$ and

$$w(D \cap E_\ell) = 0 \quad \text{for all } \ell \in L. \tag{9}$$

We show first that D passes through all vertices in U, includes exactly one of the two parallel paths between u_{j-1} and u_j for $j = 1, \ldots, n$, and exactly one of the parallel paths between u_{n+i-1} and u_{n+i} for all $i = 1, \ldots, m$.

As we observed above, we have u_0 as a vertex of D. Thus D must contain an arc leaving u_0, say it contains $(u_0, a_{x^1_1})$. Then, by (9) and by Lemma 1, we must have $E^v_{x^1_1} \subseteq D$, i.e., D must pass through vertex y_{11}. Since only $(y_{11}, a(x^2_1))$ is leaving y_{11}, by repeating the above argument we can conclude that we must also have $E^v_{x^2_1} \subseteq D$, etc., finally arriving at $E^v_{x^{o_1}_1} \subseteq D$, i.e., that D includes u_1 as a vertex. Repeating the same argument, we can prove by induction that for all indices $j = 1, \ldots, n$, if $E^v_{x^1_j} \subseteq D$, then we must have $E^v_{x^k_j} \subseteq D$ for all $k = 1, \ldots, o_j$, and that if $E^v_{\bar{x}^1_j} \subseteq D$, then we must also have $E^v_{\bar{x}^k_j} \subseteq D$ for all $k = 1, \ldots, \bar{o}_j$. We then define a truth assignment

X^D by

$$X_j^D = \begin{cases} 1 & \text{if } E^v_{\bar{x}_j^1} \subseteq D, \\ 0 & \text{if } E^v_{x_j^1} \subseteq D. \end{cases}$$

Furthermore, repeating a similar argument for vertices $u_n, u_{n+1}, \ldots, u_{n+m-1}, u_{n+m}$ we can also conclude that D must contain the set $E^c_{\ell_i}$ for exactly one of the literals $\ell_i \in C_i$, for each clause C_i of ϕ. Since D is a circuit in which no vertex $a(\ell)$ or $b(\ell)$ is repeated, we must have that $\ell_i(X^D) = 1$ for all $i = 1, \ldots, m$, i.e., that X^D is indeed a satisfying assignment of ϕ.

These observations prove the reverse direction of our main claim, and hence conclude the proof of Theorem 1. □

3 Proof of Theorem 2

We can repeat essentially the same proof as for the directed case, with the exception that we associate with every literal occurrence $\ell \in L$ a different subgraph denoted by E_ℓ: We now associate with $\ell \in L$ six nodes, $a = a(\ell), b = b(\ell), c = c(\ell)$, $d = d(\ell), e = e(\ell)$, and $f = f(\ell)$, and the following ten edges:

$$E_\ell = \big\{(a,b),(b,c),(c,d),(d,e),(e,f),(a,f),(a,p),(b,q),(d,r),(e,s)\big\},$$

where nodes $p = p(\ell), q = q(\ell), r = r(\ell)$, and $s = s(\ell)$ are identified with the other nodes of G, in the same way as in the previous proof. To simplify notation, we omit the reference to ℓ whenever it is clear from the context which literal occurrence we are talking about. The weights of the edges of E_ℓ are defined as

$$w(a,p) = w(b,q) = w(d,r) = w(e,s) = \frac{5}{2}, \quad \text{and}$$
$$w(a,b) = w(b,c) = w(c,d) = w(d,e) = w(e,f) = w(a,f) = -1.$$

Note that in each of these subgraphs there is a negative circuit (see Fig. 4), formed by the six edges $D_\ell = \{(a,b),(b,c),(c,d),(d,e),(e,f),(a,f)\}$. We denote by $\mathcal{X} = \{D_\ell \mid \ell \in L\}$ the collection of these negative circuits, and let $\mathcal{F}$ denote the family of all negative circuits in G.

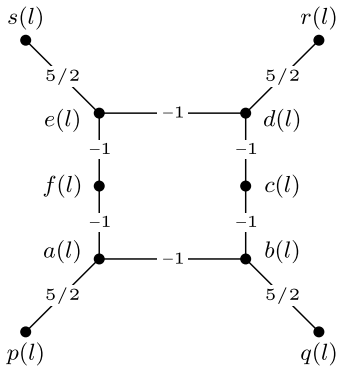

Fig. 4 The undirected graph associated with literal occurrences

By an analogous proof as in the previous section, we can show that there exists a negative circuit belonging to $\mathcal{F}\backslash\mathcal{X}$ if and only if ϕ has a satisfying assignment. The key observation in this case, the analogue of Lemma 1, is the following claim, which can easily be verified, e.g., by looking at Fig. 4.

Lemma 2 *For a circuit D of G not belonging to $\mathcal{X}$ and literal occurrence $\ell \in L$ we have*

$$w(D \cap E_\ell) \in \{0, 1, 2, 3, 4\}$$

and it is equal to 0 *only if $D \cap E_\ell$ is one of the following three sets*: $\emptyset$,

$$E_\ell^v = \big\{(b,c),(c,d),(d,e),(e,f),(a,f),(a,p),(b,q)\big\}, \quad \text{or}$$

$$E_\ell^c = \big\{(a,b),(b,c),(c,d),(e,f),(a,f),(d,r),(e,s)\big\}.$$

Remark The construction in Theorem 1 can be slightly modified to show that the NP-hardness result of Corollary 2 applies to polyhedra with 0/1-vertices (see arXiv:0801.3790v1 for more details).

Acknowledgements We thank the anonymous referees for their helpful remarks.

References

1. Abdullahi, S.D.: Vertex enumeration and counting for certain classes of polyhedra. Ph.D. thesis, Computing (Computer Algorithms), Leeds University (2003)
2. Avis, D., Bremner, B., Seidel, R.: How good are convex hull algorithms. Comput. Geom. Theory Appl. **7**, 265–302 (1997)
3. Avis, D., Fukuda, K.: A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. Discrete Comput. Geom. **8**(3), 295–313 (1992)
4. Avis, D., Fukudam, K.: Reverse search for enumeration. Discrete Appl. Math. **65**(1–3), 21–46 (1996)
5. Amaldi, E., Pfetsch, M.E., Trotter, L.E.: On the maximum feasible subsystem problem, IISs and IIS-hypergraphs. Math. Program. **95**, 533–554 (2003)
6. Balinski, M.L.: An algorithm for finding all vertices of convex polyhedral sets. SIAM J. Appl. Math. **9**, 72–81 (1961)
7. Boros, E., Elbassioni, K., Gurvich, V., Khachiyan, L.: Enumerating minimal dicuts and strongly connected subgraphs and related geometric problems. In: Bienstock, D., Nemhauser, G. (eds.) Integer Programming and Combinatorial Optimization, 10th International IPCO Conference. Lecture Notes in Computer Science, vol. 3064, pp. 152–162. Springer, Berlin (2004). (An extended version is to appear in Algorithmica)
8. Berge, C.: Hypergraphs. Elsevier-North Holland, Amsterdam (1989)
9. Bremner, D., Fukuda, K., Marzetta, A.: Primal–dual methods for vertex and facet enumeration. Discrete Comput. Geom. **20**, 333–357 (1998)
10. Bussieck, M.R., Lübbecke, M.E.: The vertex set of a 0/1 polytope is strongly $\mathcal{P}$-enumerable. Comput. Geom. Theory Appl. **11**(2), 103–109 (1998)
11. Bremner, D.: Incremental convex hull algorothms are not output sensitive. Discrete Comput. Geom. **21**, 57–68 (1999)
12. Charnes, A., Cooper, W.W., Henderson, A.: An Introduction to Linear Programming. Wiley, New York (1953)
13. Chazelle, B.: An optimal convex hull algorithm in any fixed dimension. Discrete Comput. Geom. **10**, 377–409 (1993)
14. Chakravarti, N.: Some results concerning post-infeasibility analysis. Eur. J. Oper. Res. **73**, 139–143 (1994)
15. Chvátal, V.: Linear Programming. Freeman, San Francisco (1983)

16. Chand, D.R., Kapur, S.S.: An algorithm for convex polytopes. J. Assoc. Comput. Mach. **17**(1), 78–86 (1970)
17. Cook, S.A.: The complexity of theorem proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing, pp. 151–158 (1971)
18. Dyer, M.E.: The complexity of vertex enumeration methods. Math. Oper. Res. **8**, 381–402 (1983)
19. Dyer, M.E., Proll, L.G.: An algorithm for determining all extreme points of a convex polytope. Math. Program. **12**, 81–96 (1977)
20. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. SIAM J. Comput. **24**, 1278–1304 (1995)
21. Farkas, J.: Theorie der einfachen ungleichungen. J. Rein. Angew. Math. **124**, 1–27 (1901)
22. Fredman, M., Khachiyan, L.: On the complexity of dualization of monotone disjunctive normal forms. J. Algorithms **21**, 618–628 (1996)
23. Floyd, R.W.: Algorithm 97: Shortest path. Commun. Assoc. Comput. Mach. **5**, 345 (1962)
24. Fukuda, K., Liebling, Th.M., Margot, F.: Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. CGTA **8**, 1–12 (1997)
25. Gallai, T.: Maximum-minimum Sätze über Graphen. Acta Math. Acad. Sci. Hung. **9**, 395–434 (1958)
26. Gleeson, J., Ryan, J.: Identifying minimally infeasible subsystems of inequalities. ORSA J. Comput. **2**(1), 61–63 (1990)
27. Johnson, D.S., Preparata, F.P.: The densest hemisphere problem. Theor. Comput. Sci. **6**, 93–107 (1978)
28. Johnson, D.S., Papadimitriou, Ch.H.: On generating all maximal independent sets. Inf. Process. Lett. **27**, 119–123 (1988)
29. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. **44**, 169–188 (1986)
30. Khachiyan, L.: A polynomial algorithm in linear programming. Sov. Math. Dokl. **20**, 191–194 (1979)
31. Lovász, L.: Combinatorial optimization: some problems and trends. DIMACS Technical Report 92-53, Rutgers University (1992)
32. Lawler, E., Lenstra, J.K., Rinnooy Kan, A.H.G.: Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. SIAM J. Comput. **9**, 558–565 (1980)
33. Mattheiss, T.H.: An algorithm for determining irrelevant constraints and all vertices in systems of linear inequalities. Oper. Res. **21**, 247–260 (1973)
34. Motzkin, T.S., Raiffa, H., Thompson, G.L., Thrall, R.M.: The double description method. In: H.W. Kuhn and A.W. Tucker (eds.) Contributions to the Theory of Games, vol. II, pp. 51–73 (1953)
35. Pfetsch, M.E.: The maximum feasible subsystem problem and vertex-facet incidences of polyhedra. Dissertation, TU Berlin (2002)
36. Provan, J.S.: Efficient enumeration of the vertices of polyhedra associated with network lp's. Math. Program. **63**(1), 47–64 (1994)
37. Read, R.C., Tarjan, R.E.: Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. Networks **5**, 237–252 (1975)
38. Ryan, J.: IIS-hypergraphs. SIAM J. Discrete Math. **9**(4), 643–653 (1996)
39. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1986)
40. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. A. Springer, Berlin (2003)
41. Seidel, R.: Output-size sensitive algorithms for constructive problems in computational geometry. Computer Science. Cornell University, Ithaka (1986)
42. Swart, G.: Finding the convex hull facet by facet. J. Algorithms **6**, 17–48 (1985)
43. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comput. **8**, 410–421 (1979)
44. Warshall, S.: A theorem on boolean matrices. J. Assoc. Comput. Mach. **9**, 11–12 (1962)