Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron

Komei Fukuda
Inst. for Operations Research
Swiss Federal Institute of Technology
CH-8092 Zurich
Switzerland
fukuda@ifor.math.ethz.ch

Thomas M. Liebling
Dept. of Mathematics
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland
liebling@dma.epfl.ch

Francois Margot *
Dept. of Mathematics
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland
margot@dma.epfl.ch

(August 6, 1994; Revised October 16, 1995)

Abstract

In this paper, we investigate the applicability of backtrack technique to solve the vertex enumeration problem and the face enumeration problem for a convex polyhedron given by a system of linear inequalities. We show that there is a linear-time backtrack algorithm for the face enumeration problem whose space complexity is polynomial in the input size, but the vertex enumeration problem requires a backtrack algorithm to solve a decision problem, called the restricted vertex problem, for each output, which is shown to be NP-complete. Some related NP-complete problems associated with a system of linear inequalities are also discussed, including the optimal vertex problems for polyhedra and arrangements of hyperplanes.

1 Introduction

Listing all vertices of an n-dimensional convex polyhedron given by a system of linear inequalities is a fundamental problem in polyhedral combinatorics and computational geometry. While many interesting ideas for efficient enumeration have been introduced [1, 3, 5, 11, 13, 16], the most important question on the vertex enumeration problem is still open:

> Is there a polynomial-time algorithm for the vertex enumeration problem?

where a *polynomial-time* algorithm for an enumeration problem is defined as one whose time complexity is polynomial in the input size and the output size. One might be naturally tempted to ask an even stronger question:

> Is there a linear-time algorithm for the vertex enumeration problem?

where a *linear-time* algorithm for an enumeration problem is a polynomial-time algorithm whose time complexity is linear in the output size. Obviously one cannot do better than linear-time algorithms.

*Research supported by FNRS grant, Switzerland

Using the language of Valiant [18], the existence of a linear-time algorithm is equivalent to the *P-enumerability* of the vertices of a convex polyhedron.

Under the assumption that an input system is nondegenerate, Dyer [5] found the first linear-time algorithm, which makes use of depth first search and a balanced tree data structure. Recently, Avis and Fukuda [1] proposed a linear-time algorithm for generating all feasible bases of a general input system, using so-called reverse search technique. It is not only that the new algorithm is more efficient than the former algorithm for the nondegenerate inputs, but it has also a substantially smaller space complexity. More precisely, the space complexity of the Avis-Fukuda algorithm is proportional to the input size while Dyer's algorithm requires space proportional to the input size and the output size. Obviously, an algorithm whose space complexity is polynomial in the input size and independent of the output size has a considerable advantage since the number of vertices of a convex polyhedron is in general exponential in the input size. In this context, we shall differentiate the P-enumerability from the *strong P-enumerability* which means the existence of a linear-time algorithm whose space complexity is polynomial in the size of input and not depending on the output size. Thus the latter algorithm proves the strong P-enumerability of vertices of a convex polyhedron under nondegeneracy of an input system.

While the existence of a polynomial-time algorithm for vertex enumeration remains open, Murty and Chung [14] found the first polynomial-time algorithm for enumerating all faces of a general convex polytope using the interesting notion of segments of convex polytopes.

In this paper, we shall look at the two enumeration problems from a new direction. Namely we investigate the applicability of the backtrack search technique to these enumeration problems. It is rather surprising that backtrack search, which is one of the most popular general schemes for enumeration, has not been studied for these problems, as far as we know.

A natural backtrack search algorithm [1] for the vertex (face) enumeration, which we investigate in this paper, is as follows. First enumerate all vertices (faces) satisfying the first inequality with equality. At the general stage k, enumerate all vertices (faces) satisfying the kth inequality with equality and the first $(k-1)$st inequalities with strict inequalities.

Our investigation shows that this straightforward backtrack search algorithm for the face enumeration is in fact a linear-time algorithm with small space complexity, but the same backtrack algorithm for the vertex enumeration problem requires for each output a solution to a problem, which will be shown to be NP-complete in the strong sense. This problem, which we call the *restricted vertex problem*, is to decide for a given system $Ax \leq b$ and a subset S of the linear inequalities whether there exists a vertex of the polyhedron $P = \{x : Ax \leq b\}$ satisfying each member of S with strict inequality.

To summarize, by using the backtrack technique, we show that the set of faces of a general convex polyhedron is strongly P-enumerable, while it is unlikely that the straightforward backtrack algorithm works efficiently for the vertex enumeration problem. For a subclass of the vertex enumeration to be strongly P-enumerable it is sufficient that the associated restricted vertex problem admits a polynomial-time algorithm. In this paper we will also discuss the NP-completeness of related decision problems, including the optimal vertex problem for an arrangement of hyperplanes.

[1] Balinski's backtrack algorithm [3] employs a similar strategy but the crucial difference is that our backtrack algorithm imposes strict inequality instead of removing constraints so that each face/vertex won't be output many times and thus its time complexity is output-sensitive.

2 Basic Terminology

In this section, we give formal definitions of basic terminologies we need for the paper.

Let Σ denote the set $\{0,1\}$ (alphabet) and Σ^* denote the set of ordered sequences (words) of 0's and 1's. A *relation* R is a boolean function from the product space $\Sigma^* \times \Sigma^*$ to Σ. An instance of the *enumeration problem associated with a relation* R is to list all members of the *enumeration* set $Enum(R,x) = \{y : R(x,y) = 1\}$, for a given x.

Let A and b be rational $m \times n$ matrix and m vector, respectively. We note a_i the ith row vector of A and b_i the ith entry of b for $i = 1, \ldots, m$. We denote by $P(A,b)$ the convex polyhedron $\{x : Ax \leq b\}$ in R^n. A subset F of $\{1, 2, \ldots, m\}$ is said to be *face-defining* for the polyhedron $P(A,b)$ if there is a point x in the polyhedron such that $a_i x = b_i$ for all $i \in F$ and $a_j x < b_j$ for all $j \notin F$. For each face-defining subset F, the *face of* $P(A,b)$ *associated with* F is $\{x \in P(A,b) | a_i x = b_i \text{ for all } i \in F\}$. A face-defining subset F is *vertex-defining* if the associated face consists of a single point. Since there is an obvious bijection between the set of faces and the set of face-defining subsets, we identify face-defining subsets with faces whenever there are no ambiguities.

The *vertex relation* R_v (the *face relation* R_f) is defined by $R_v([A,b],F) = 1$ ($R_f([A,b],F) = 1$) if and only if F is vertex-defining (face-defining, respectively) for the convex polyhedron $\{x : Ax \leq b\}$.

The *vertex (face) enumeration problem* is the enumeration problem associated with the vertex (face) relation. The well-known upper bound theorem [12] gives the exact sizes of largest possible outputs as an exponential function of m and n. Clearly, even an ideal algorithm will require in general a running time exponential in the input size.

A natural way to evaluate the performance of enumeration algorithms is thus to express the running time as a function of both the input and output sizes. A *polynomial-time* algorithm for the enumeration problem associated with a relation R is one whose time complexity is polynomial in the input size $|x|$ and the output size $|Enum(R,x)|$ for any input x. A *linear-time* algorithm is a polynomial time algorithm whose time complexity is linear in the output size. Finally, a *compact* algorithm for the enumeration problem associated with a relation R is one whose space complexity is polynomial in the input size $|x|$.

A relation R or the associated enumeration sets are called (*strongly*) *P-enumerable* if there exists a (compact) linear-time algorithm for the enumeration problem.

3 Face and vertex enumeration for a polyhedron

Let A be an $m \times n$ rational matrix, b be a rational m-vector and $P(A,b)$ the polyhedron associated with the pair (A,b).

First we shall describe a backtrack algorithm for listing all face-defining subsets of $\{1, 2, \ldots, m\}$ for $P(A,b)$. The idea is quite simple. We partition the set $\mathcal{F}$ of faces in m parts: $\mathcal{F}^1 \cup \mathcal{F}^2 \cup \cdots \cup \mathcal{F}^m$, where $\mathcal{F}^k = \{F \in \mathcal{F} | k \in F \text{ and } F \cap \{1, 2, \ldots, k-1\} = \emptyset\}$. Each nonempty $\mathcal{F}^k$ will be then partitioned into smaller sets inductively. Thus the core problem to be solved is the following.

RESTRICTED FACE OF A POLYHEDRON (RFP)

Input: A rational $m \times n$ matrix A, a rational m-vector b, two subsets R and S of $\{1, 2, \ldots, m\}$.

Question: Does there exist a face-defining set F for $P(A, b)$ such that $R \subseteq F$ and $F \cap S = \emptyset$? (Or equivalently, does there exist a point x in $P(A, b)$ such that $a_r \cdot x = b_r$ for each $r \in R$ and $a_s \cdot x < b_s$ for each $s \in S$?)

The problem RESTRICTED FACE OF A POLYHEDRON (RFP) is easily seen to be in the class P. Indeed, consider the following linear program:

$$\begin{array}{lll} \max & y & \\ \text{s.t.} & a_r \cdot x = b_r & \text{for each } r \in R, \\ & a_s \cdot x + y \leq b_s & \text{for each } s \in S, \\ & a_t \cdot x \leq b_t & \text{for each } t \in \{1, \ldots, m\} - (R \cup S). \\ & y \leq 1 & \end{array}$$

Then RFP has a "yes" answer if and only if the optimal value of the above LP is strictly positive. Let RFP(R, S) denote the answer to the RFP problem for sets R and S. Note that if RFP(R', S') = "yes" then, by solving the above LP with $R := R'$ and $S := \{j\}$ for each $j \in \{1, \ldots, m\} - R'$, one can find the minimal face-defining set F for $P(A, b)$ with $F \supseteq R'$, namely

$$F = R' \cup \{j \in \{1, \ldots, m\} - R' \mid \text{RFP}(R', \{j\}) = \text{"no"}\}. \tag{3.1}$$

Moreover, as RFP(R', S') = "yes", we have RFP$(R', \{j\})$ = "yes" for each $j \in S'$ and thus $F \cap S' = \emptyset$. To enumerate all faces of a given polyhedron $P(A, b)$, consider the following algorithm:

```
FaceEnum(R, S)
/* lists all face-defining sets F with R ⊆ F and F ∩ S = ∅ */
begin
   if RFP(R, S) = "yes" then
   begin
      find the minimal set F ⊇ R defining a face of P(A, b);
      J := {1, ..., m} − (F ∪ S);
      fix any linear ordering of J as {j_1, j_2, ..., j_t};
      for k := 1 to t do
         FaceEnum(F ∪ {j_k}, S ∪ {j_1, j_2, ..., j_{k−1}});
      output F;
   end;
end.
```

Now we can state and prove the following theorem.

Theorem 3.1 *The algorithm FaceEnum(R, S) outputs correctly all face-defining sets F with $R \subseteq F$ and $F \cap S = \emptyset$ without duplication. Furthermore, there is an implementation of FaceEnum(R, S) which runs with time complexity $O(m\, l(m, n) f)$ and space complexity $O(p(m, n))$ where f is the number of outputs, and $l(m, n)$ and $p(m, n)$ are time and space needed to solve an LP with n variables and m constraints, respectively.*

Proof. To check the correctness of this algorithm FaceEnum(R, S), we assume by induction that it outputs correctly for any $R \supset R$ and $S' \supseteq S$. It is clear that when $R \cup S$ is large enough, there is either a unique face or no face to be listed, and it is easily seen that the algorithm works correctly. Otherwise, the algorithm outputs at least the minimal face-defining set F. The remaining facet-defining sets can be partitioned into disjoint families, $\mathcal{F}^{j_1} \cup \mathcal{F}^{j_2} \cup \cdots \cup \mathcal{F}^{j_t}$, where $\mathcal{F}^{j_k}$ is the set of facet-defining sets F' with $(F \cup \{j_k\}) \subseteq F'$ and $F' \cap (S \cup \{j_1, j_2, \ldots, j_{k-1}\}) = \emptyset$. By induction, these families are listed correctly by the recursive calls of the algorithm, and thus the correctness of the algorithm follows.

For an implementation, one can use an unraveled (nonrecursive) version of the algorithm. The most simple one is to trace the search tree with the depth-first search strategy, which is a backtrack algorithm. Since the depth of recursive calls is at most m, the space requirement for this nonrecursive form is dominated by that of solving linear programs. The time necessary to check whether there exists at least one F with $R \subseteq F$ and $F \cap S = \emptyset$ is $l(m,n)$. If it is verified positively, it takes $O(m\ l(m,n))$ time to find the minimal such F. The subsequent $O(m)$ calls FaceEnum($F \cup \{j_k\}$, $S \cup \{j_1, j_2, \ldots, j_{k-1}\}$) for $k = 1, \ldots, t$ will either find again a new face or detect the emptiness of the family. Since the work of finding a new face is counted as the work of the new face, the total work necessary for each face does not exceed $O(m\ l(m,n))$ and the result follows. ∎

For polynomial time linear programming algorithms, see [17].

Now we try to apply the same backtrack strategy to the vertex enumeration problem and analyze the complexities. One can easily see that the following problem is the core problem to be solved at the general stage:

RESTRICTED-VERTEX OF A POLYHEDRON (RVP)

Input: A rational $m \times n$ matrix A, a rational m-vector b, two subsets R and S of $\{1, 2, \ldots, m\}$.

Question: Does there exist a vertex-defining set V for $P(A,b)$ such that $R \subseteq V$ and $V \cap S = \emptyset$? (Or equivalently, does there exist a vertex v of $P(A,b)$ such that $a_r \cdot v = b_r$ for each $r \in R$ and $a_s \cdot v < b_s$ for each $s \in S$?)

In order to write an algorithm in recursive form, let RestrictedVertex(R, S) be a function which returns the answer to the RVP problem above with time complexity $T(m,n)$ and space complexity $S(m,n)$. Using this function, we have the following recursive algorithm to enumerate all vertex-defining sets of a polyhedron.

```
VertexEnum(R, S)
/* lists all vertex-defining sets V with R ⊆ V and V ∩ S = ∅
   assuming that there is at least one such set. */
begin
   let J := {1,...,m} − (R ∪ S);
   fix any linear ordering of J as {j_1, j_2, ..., j_t};
   for k = 1 to t do
      rv(k):=RestrictedVertex (R ∪ {j_k}, S ∪ {j_1, j_2, ..., j_{k−1}});
      if rv(k)=yes then
         VertexEnum(R ∪ {j_k}, S ∪ {j_1, j_2, ..., j_{k−1}})
   end;
   if rv(k) = no for all k = 1, 2, ..., t then
      output R;
end.
```

Theorem 3.2 *The algorithm VertexEnum(R, S) outputs correctly all vertex-defining sets V with $R \subseteq V$ and $V \cap S = \emptyset$ without duplication, assuming there is at least one such set. Furthermore, there is an implementation of VertexEnum(R, S) which runs with time complexity $O(m^2\, T(m,n)\, f_0)$ and space complexity $O(S(m,n))$ where f_0 is the number of outputs.*

Proof. The correctness of the algorithm is clear. The backtrack implementation requires only to store the indices k at each level and thus the space complexity is dominated by $S(m,n)$. In order to evaluate the time complexity, observe that the depth of the backtrack search tree (i.e. the depth of recursion) is at most m, and that the leaves of the tree are exactly the vertices to be listed. It is clear that the maximum amount of work necessary before outputting a new vertex (since some vertex has been listed) is bounded by the depth times m calls of RestrictedVertex which is $O(m^2\, T(m,n))$. This completes the proof. ∎

If both $T(m,n)$ and $S(m,n)$ are polynomially bounded in m and n, this would imply that the set of vertices of a polyhedron is strongly P-enumerable. In the next section, however, we will show that the RVP problem is unfortunately NP-complete in the strong sense.

4 Vertex problems for polyhedra and NP-completeness

In this section, we show the NP-completeness of various decision problems associated with a system of linear inequalities, including the restricted vertex problem (RVP) described in the previous section.

First of all we shall prove the NP-completeness of the following two decision problems:

OPTIMAL VERTEX OF A POLYHEDRON (OVP)

Input: A rational $m \times n$ matrix A, rational m-vector b and n-vector c, a rational number K.

Question: Does there exist a vertex v of $P(A,b)$ with $c \cdot v \geq K$?

K-VERTEX OF A POLYHEDRON (KVP)

Input: A rational $m \times n$ matrix A, rational m-vector b and n-vector c, a rational number K.

Question: Does there exist a vertex v of $P(A,b)$ with $c \cdot v = K$?

One might suspect that these problems are polynomially reducible to each other by some direct transformation. To our surprise, neither of the two directions seems to be obvious.

Theorem 4.1 *Both OVP and KVP are NP-complete in the strong sense.*

Proof. It is clear that both problems are in the class NP. The proofs of the NP-completeness will be obtained by a polynomial time transformation from the following problem, known to be NP-complete in the strong sense ([9], problem [GT39]):

DIRECTED HAMILTONIAN PATH (DHP)

Input: Directed graph $G = (V, A)$ and two distinct vertices s, $t \in V$.

Question: Does G contain a directed Hamiltonian path from s to t ?

Let $G = (V, A)$ be a directed graph and $s \neq t \in V$. Associate a variable x_{ij} with each arc $(i, j) \in A$. Let $P(G)$ be the polytope given by:

$$\sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} = 0, \qquad \text{for each } i \in V - \{s,t\}, \tag{4.1}$$

$$\sum_{j|(s,j)\in A} x_{sj} - \sum_{j|(j,s)\in A} x_{js} = 1, \tag{4.2}$$

$$\sum_{j|(t,j)\in A} x_{tj} - \sum_{j|(j,t)\in A} x_{jt} = -1, \tag{4.3}$$

$$x_{ij} \geq 0, \qquad \text{for each } (i,j) \in A. \tag{4.4}$$

The matrix of the coefficients of these inequalities is totally unimodular ([15], Proposition 2.6, p. 542) implying that $P(G)$ is integral. It follows that an extreme point x of $P(G)$ is the characteristic vector of a directed path joining s to t in G and, possibly, a set of circuits. If a circuit C exists, then x is a convex combination of the two points obtained by adding or subtracting small $\epsilon > 0$ on all the arcs of the circuit, a contradiction. Hence x is the characteristic vector of a simple directed path joining s to t. One verify easily that all such paths are extreme points of $P(G)$, proving that the extreme points of $P(G)$ are exactly the characteristic vectors of the simple directed paths joining s to t in G. These two facts thus imply that, for $K = |V| - 1$ and $c = \mathbf{1}$ (the vector of all 1's), both the OVP and the KVP problems for $P(G)$ are NP-complete in the strong sense. This completes the proof. ∎

A bounded polyhedron is called a *polytope*. Note that the OVP problem for a polytope can be solved in polynomial time, as it amounts to solve a linear program, but curiously the KVP problem is not polynomially solvable even when restricted to polytopes: let $P^*(G)$ be the polytope obtained by adding to $P(G)$ the inequalities

$$x_{ij} \leq |V|, \qquad \text{for each } (i,j) \in A. \tag{4.5}$$

Then, for $K = |V| - 1$ and $c = \mathbf{1}$, the KVP problem for $P^*(G)$ is NP-complete in the strong sense. Note that the KVP problem has been studied in [4] and shown to be NP-complete (but not in the strong sense) for the assignment polytope.

Using the same basic polynomial transformation from the DHP problem, it is easy to show that the RVP problem is also NP-complete in the strong sense. Let $P^{**}(G)$ the polyhedron obtained by adding the following inequality to $P(G)$:

$$\sum_{(i,j)\in A} x_{ij} \geq |V| - 2. \tag{4.6}$$

Then a vertex v of $P^{**}(G)$ satisfies

$$\sum_{(i,j)\in A} v_{ij} > |V| - 2. \tag{4.7}$$

if and only if v corresponds to the characteristic vector of a solution to the DHP problem. Thus we have:

Theorem 4.2 *RVP is NP-complete in the strong sense.*

Note that RVP remains to be NP-complete in the strong sense for polytopes. To see this, we can simply modify the proof above by adding a bounding inequality: $\sum_{(i,j)\in A} x_{ij} \leq |V|$ and by considering a vertex satisfying this inequality as well as (4.6) with strict inequality.

A polyhedron $P(A,b)$ in R^n is *non degenerate* (or *simple*) if each of its vertex satisfies exactly n of the inequalities defining $P(A,b)$ with equality. The following problem is easily shown to be in NP [5]:

DEGENERATE POLYHEDRON (DP)

Input: A rational $m \times n$ matrix A and a rational m-vector b.

Question: Is $P(A,b)$ a degenerate polyhedron ?

In [5] the DP problem has been shown to be NP-complete even when restricted to polytopes, but not in the strong sense. Below, we give a strongly polynomial transformation from DHP to DP restricted to polytopes. Let $G = (V,A)$, $s,t \in V$, be an instance of DHP. Associate a variable x_{ij} with each arc $(i,j) \in A$, set $n := |A|$, $m := |V| - 1 + |A|$, $\epsilon := 1/(|V|^3)$ and consider the following polyhedron $Q(G)$:

$$\sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} \;\geq\; \epsilon, \qquad \text{for each } i \in V - \{s,t\}, \tag{4.8}$$

$$\sum_{j|(s,j)\in A} x_{sj} - \sum_{j|(j,s)\in A} x_{js} \;\geq\; 1, \tag{4.9}$$

$$x_{ij} \;\geq\; 0, \qquad \text{for each } (i,j) \in A. \tag{4.10}$$

Note that an extreme point v of $Q(G)$ has to satisfy at least $(|A| - (|V| - 1))$ of the inequalities (4.10) with equality. It follows that v has at most $(|V| - 1)$ positive entries. Moreover, to satisfy (4.8) and (4.9), for each node $i \in V - \{t\}$ there exists at least one node $j \in V$ such that $v_{ij} > 0$. These two observations imply that exactly $(|V| - 1)$ entries of v are positive, yielding that $Q(G)$ is non degenerate.

We claim that if we add the following inequality to $Q(G)$,

$$\sum_{(i,j)\in A} x_{ij} \leq (|V|-1) + (1/2)\ (|V|-1)\ (|V|-2)\ \epsilon, \tag{4.11}$$

the resulting polytope is non degenerate if and only if there is no directed Hamiltonian path joining s to t in G. To verify the claim, it is enough to check that a vertex of $Q(G)$ satisfies (4.11) with equality if and only if it corresponds to a directed Hamiltonian path joining s to t in G.

Recall that a directed graph $G' = (V', A')$ is an *anti-arborescence rooted at node* $t \in V'$ if G' contains $(|V'|-1)$ arcs and a directed path joining i to t for each $i \in V' - \{t\}$.

Let T be the set of arcs (i,j) such that $v_{ij} > 0$. The above discussion shows that, for each $i \in V-\{t\}$, there exists a unique node $j \in V$ such that (i,j) is in T implying that T is the union of an anti-arborescence with root t with possibly a set of circuits. As this anti-arborescence and circuits are necessarily node disjoint, it is indeed impossible that T contains a circuit C, as shown by summing inequalities (4.8)-(4.9) corresponding to the nodes in C. Hence T is an anti-arborescence rooted at t with node set V.

For each node $i \in V - \{t\}$, let p_{it} be the number of arcs in the path joining i to t in T. Then

$$\sum_{(i,j)\in A} v_{ij} = p_{st} + \sum_{i\in V-\{s,t\}} p_{it}\ \epsilon. \tag{4.12}$$

Thus, if T is a directed path joining s to t in G then v satisfies (4.11) with equality as $p_{st} = |V|-1$ and

$$\sum_{i\in V-\{s,t\}} p_{it}\ \epsilon \ = \ (1/2)(|V|-1)(|V|-2)\ \epsilon. \tag{4.13}$$

On the other hand, if T is not a directed path joining s to t in G then v does not satisfy (4.11) with equality as $p_{st} \leq |V|-2$ and

$$\sum_{i\in V-\{s,t\}} p_{it}\ \epsilon \ \leq \ (|V|-2)\ (|V|-2)\ \epsilon < 1. \tag{4.14}$$

As ϵ is polynomial in $|V|$, this yields:

Theorem 4.3 *DP is NP-complete in the strong sense.*

5 Vertex problems for arrangements

In this section, we consider the "optimal vertex problem for arrangements" which is the direct analogue of the optimal vertex problem (OVP) for polyhedra.

Let A be an $m \times n$ rational matrix and b be a rational m-vector. The *arrangement of hyperplanes* associated with the pair (A, b) is the family:

$$\begin{aligned} \mathcal{H}(A,b) &= \{H_1, H_2, \ldots, H_m\}, && (5.1)\\ \text{where} \quad H_i &= \{x \in R^n \,|\, a_i \cdot x = b_i\} \quad \text{for } i = 1, \ldots, m. && (5.2) \end{aligned}$$

A point v in R^n is called a *vertex of the arrangement* $\mathcal{H}$(A, b) if v is the unique solution to the system:

$$a_i \cdot x = b_i \qquad \text{for all } i \text{ with} \qquad a_i \cdot v = b_i. \tag{5.3}$$

In this section, we show the NP-completeness of the following two problems:

OPTIMAL VERTEX OF AN ARRANGEMENT (OVA)

Input: A rational $m \times n$ matrix A, rational m-vector b and n-vector c, a rational number K.

Question: Does there exist a vertex v of $\mathcal{H}$(A, b) with $c \cdot v \geq K$?

and

K-VERTEX OF AN ARRANGEMENT (KVA)

Input: A rational $m \times n$ matrix A, rational m-vector b and n-vector c, a rational number K.

Question: Does there exist a vertex v of $\mathcal{H}(A, b)$ with $c \cdot v = K$?

Let us give a proof of the NP-completeness of OVA and KVA: Consider an instance of DHP and build a corresponding instance for OVA and KVA as follows: To each arc $(i, j) \in A$, we associate a variable x_{ij}. Let $n := |A|$, $K := |V| - 1$, $c = \mathbf{1}$ and define the arrangement generated by the following set of hyperplanes:

$$H_i := \{\mathbf{x} \in R^n | \sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} = 0\}, \qquad \text{for each } i \in V - \{s, t\}, \tag{5.4}$$

$$H_s := \{\mathbf{x} \in R^n | \sum_{j|(s,j)\in A} x_{sj} - \sum_{j|(j,s)\in A} x_{js} = 1\}, \tag{5.5}$$

$$H_t := \{\mathbf{x} \in R^n | \sum_{j|(t,j)\in A} x_{tj} - \sum_{j|(j,t)\in A} x_{jt} = -1\}, \tag{5.6}$$

$$H_{ij} := \{\mathbf{x} \in R^n | x_{ij} = 0\}, \qquad \text{for each } (i, j) \in A. \tag{5.7}$$

First we observe that if DHP has a "yes" answer, so does the corresponding instance of OVA and KVA, as the characteristic vector of any directed Hamiltonian path lies on the $|V|$ hyperplanes H_i for $i \in V$ as well as on $(|A| - (|V| - 1)) = |A| - |V| + 1$ of the hyperplanes H_{ij} for $i \neq j \in V$. Note that the $|V|$ hyperplanes H_i for $i \in V$ are not linearly independent, but any subset of $(|V| - 1)$ of them are. Hence there are $(|A| - |V| + 1) + (|V| - 1) = |A|$ linearly independent hyperplanes containing the characteristic vector of any directed Hamiltonian path joining s to t in G, implying that the latter is a vertex of the given arrangement.

Now suppose that KVA or OVA has a "yes" answer produced by a vertex v of the constructed instance. One can write the $|A|$ equations defining the hyperplanes of the instance as a system of the form $Ax = b$. It is well known that the matrix $[A, b]$ is totally unimodular (see [15] for example). Thus any vertex of the arrangement has only $+1, -1$, or 0 coordinates, as shown by Cramer's rule for solving a linear system.

Let S be a set of n linearly independent hyperplanes of the given family whose intersection is v. As the $|V|$ hyperplanes in $\{H_i | i \in V\}$ are not linearly independent, the number of these hyperplanes which are in S is at most $(|V|-1)$. Hence the number of non zero coordinates of v is at most $(|V|-1)$. As $c = \mathbf{1}$ and $c \cdot v \geq K = (|V|-1)$, we have that exactly $(|V|-1)$ coordinates of v are $(+1)$, all the others being (0)'s. Thus v is the characteristic vector of a set P of $(|V|-1)$ arcs of A. This also implies that KVA has a "yes" answer if and only if OVA has a "yes" answer

If P is a directed Hamiltonian path in G joining s to t, then we are done. Otherwise, P contains a directed path joining s to t in G and at least one directed cycle C. But consider $v' \in R^n$ defined by

$$v'_{ij} = \begin{cases} 0 & \text{if } (i,j) \in C, \\ v_{ij} & \text{otherwise,} \end{cases} \qquad \text{for each } (i,j) \in A. \tag{5.8}$$

Then $v' \neq v$ also satisfies all equalities in S, a contradiction. Thus we have shown:

Theorem 5.1 *Both OVA and KVA are NP-complete in the strong sense.*

The problems OVA and KVA are important when dealing with basic geometric problems for an arrangement. For example, if the following related problem can be solved with small worst case complexity, this would allow to overcome many technical details in the design of efficient algorithms for enumerating all full dimensional faces of an arrangement [10]:

VERTEX-FREE HYPERPLANES

Input: A rational $m \times n$ matrix A, rational m-vector b defining an arrangement of m hyperplanes $H_1, \ldots, H_m$.

Output: A rational $n \times n$ matrix A' and a rational n-vector b' defining an arrangement of n linearly independent hyperplanes $H_{m+1}, \ldots, H_{m+n}$ such that, for $i = m+1, \ldots, m+n$ H_i does not contain a vertex of the arrangement defined by $H_1, \ldots, H_{i-1}$.

Obviously, such a family of hyperplanes always exists and trivial randomized algorithms would find it quite easily. It seems thus very likely that this can be done with a polynomial time deterministic algorithm, although we were unable to prove it.

6 Concluding remarks

We have shown that there is a compact linear-time backtrack algorithm for enumerating all faces of a convex polytope. Unlike the standard incremental algorithms which are described as "optimal" algorithms in computational geometry literature [6], the backtrack algorithm has an output sensitive time complexity and a very small space complexity. Furthermore it is perhaps one of the easiest algorithms to implement when an efficient linear programming code is available.

On the other hand, we have also seen that the vertex enumeration problem is hard to solve by using a simple backtrack strategy. Of course this does not deny the P-enumerability of the vertex set of a convex polyhedron. In fact, we can consider our result to be quite positive one, since it says that if the restricted vertex problem is polynomially solvable for some subclass of polyhedra, the associated vertex set is strongly P-enumerable. One can easily see that the bipartite matching polyhedra constitute such a class. This is quite natural since the strong P-enumerability of the bipartite matching has been proved [7, 8]. In fact, using similar arguments we can prove that the backtrack algorithm works efficiently for many classes of combinatorial polyhedra.

Acknowledgements

The authors are grateful to anonymous referees for their careful reading of the manuscript and many helpful suggestions.

References

[1] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discr. Comput. Geom.* 8 (1992), 295-313.

[2] D. Avis, K. Fukuda, "Reverse search for enumeration", *Discrete Applied Mathematics*, to appear.

[3] M.L. Balinski, "An algorithm for finding all vertices of convex polyhedral sets", *SIAM Journal on Aplied Mathematics* 9 (1961), 72-81.

[4] R. Chandrasekaran, S.N. Kabadi, K.G. Murty, "Some NP-complete problems in linear programming", *Oper. Res. Letters* 1 (1982), 101-104.

[5] M.E. Dyer, "The complexity of vertex enumeration methods," *Math. Oper. Res.*, Vol. 8 (1983), 381-402.

[6] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, New York, 1987.

[7] K. Fukuda and T. Matsui, "Finding all minimum cost perfect matchings in bipartite graphs," *Networks* 22 (1992), 461-468.

[8] K. Fukuda and T. Matsui, "Finding all the perfect matchings in bipartite graphs," *Appl. Math. Lett.* 7 (1994), 15-18.

[9] M.R. Garey, D.S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-completness*, Freeman, 1979.

[10] F. Margot, K. Fukuda, Th. M. Liebling, "Enumeration of full dimensional cells in an arrangement," Working paper RO920918, EPF Lausanne, Switzerland.

[11] T.H. Mattheiss, "An algorithm for determining irrelevant constraints and all vertices in systems of linear inequalities," *Oper. Res.* 21 (1973), 247-260.

[12] P. McMullen and G.C. Shephard, *Convex Polytopes and the Upper Bound Conjecture*, Cambridge University Press, 1971.

[13] T.S. Motzkin, H. Raiffa, G.L. Thompson and R.M. Thrall, "The double description method," in "Contribution to the Theory of Games, Vol. II" (H.W. Kuhn and A.W. Tucker, eds.), Annals of Math. Studies 8 Princeton University Press, 1953, 51-73.

[14] K.G. Murty and S.-J. Chung, "Segments in enumerating faces," preprint (1993).

[15] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1988.

[16] R. Seidel, "Constructing higher-dimensional convex hulls at logarithmic cost per face," *Proc. 18th ACM STOC* (1986), 404-413.

[17] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, 1986.

[18] L.G. Valiant, " The complexity of enumeration and reliability problems," *SIAM J. Comput.* 8 (1979), 410-421.