TAIWANESE JOURNAL OF MATHEMATICS
Vol. 4, No. 4, pp. 599-619, December 2000

MIXED VOLUME COMPUTATION VIA LINEAR PROGRAMMING

Tangan Gao and T. Y. Li*

Abstract. A renewed algorithm is presented to calculate the mixed volume of the support $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ of a polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$ in $\mathbb{C}^n$. The key ingredient is a specially tailored application of LP feasibility tests, which allows us to calculate the *mixed cells*, their volumes constituting the mixed volume, in a *mixed subdivision* of $\mathcal{A}$ more efficiently.

The problem of finding mixed cells plays a crucial role in polyhedral homotopy methods for finding all isolated zeros of $P(\mathbf{x})$. Our new algorithm advances the speed of mixed volume computation by a considerable margin, illustrated by numerical examples.

1. Introduction

For a system of polynomials $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$ with $\mathbf{x} = (x_1, \ldots, x_n)$, write

$$p_i(\mathbf{x}) = \sum_{\mathbf{a} \in \mathcal{A}_i} c_{i,\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \quad i = 1, \ldots, n,$$

where $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$, $c_{i,\mathbf{a}} \in \mathbb{C}^* = \mathbb{C} \backslash \{0\}$ and $\mathbf{x}^{\mathbf{a}} = x_1^{a_1} \cdots x_n^{a_n}$. Here $\mathcal{A}_i$, a finite subset of $\mathbb{N}^n$, is called the *support* of $p_i(\mathbf{x})$, and $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ is called the *support* of $P(\mathbf{x})$.

By a *cell* of $\mathcal{A}$ we mean an n-tuple $C = (C_1, \ldots, C_n)$ of subsets $C_i \subseteq \mathcal{A}_i$ for $i = 1, \ldots, n$, and the volume of a cell C is the n-dimensional Euclidean volume of

$$\begin{aligned} \operatorname{conv}(C) &:= \operatorname{conv}(C_1) + \cdots + \operatorname{conv}(C_n) \\ &= \operatorname{conv}(\{\mathbf{c}_1 + \cdots + \mathbf{c}_n \,|\, \mathbf{c}_1 \in C_1, \ldots, \mathbf{c}_n \in C_n\}). \end{aligned}$$

0 Received August 2, 2000; revised October 4, 2000.
Communicated by P. Y. Wu.
2000 *Mathematics Subject Classification*: Primary 52A39; Secondary 65H10, 65H20, 90C05.
Key words and phrases: Mixed volume computation, polyhedral homotopies, system of polynomials, linear programming.
*Research was supported in part by NSF under Grant 9804846.

The *type* of C is the vector $(\dim(\mathrm{conv}\,(C_1)),\ldots,\dim(\mathrm{conv}\,(C_n)))$, and a *face* of C is a subcell $F=(F_1,\ldots,F_n)$ of C, where $F_i\subseteq C_i$ and some linear functional $\alpha\in(\mathbb{R}^n)^\vee$ attains its minimum over C_i at F_i for $i=1,\ldots,n$. We call such an α an *inner normal* of F. If F is a face of C, then $\mathrm{conv}\,(F_i)$ is a face of the polytope $\mathrm{conv}\,(C_i)$ for $i=1,\ldots,n$.

Definition 1. *A* fine mixed subdivision of $\mathcal{A}$ is a collection $\{C^{(1)},\ldots,C^{(m)}\}$ of cells of $\mathcal{A}$, $C^{(j)}=(C_1^{(j)},\ldots,C_n^{(j)})$, $j=1,\ldots,m$, such that

(a) each $\mathrm{conv}\,(C_i^{(j)})$ is a simplex of dimension $\#C_i^{(j)}-1$ and for each j,

$$\dim(\mathrm{conv}\,(C^{(j)}))=\dim(\mathrm{conv}\,(C_1^{(j)}))+\cdots+\dim(\mathrm{conv}\,(C_n^{(j)}))=n,$$

(b) $\mathrm{conv}\,(C^{(j)})\cap\mathrm{conv}\,(C^{(k)})$ is a common face of $\mathrm{conv}\,(C^{(j)})$ and $\mathrm{conv}\,(C^{(k)})$ when it is nonempty for $j\neq k$,

(c) $\bigcup_{j=1}^m\mathrm{conv}\,(C^{(j)})=\mathrm{conv}\,(\mathcal{A})$.

The cells of type $(1,\ldots,1)$ in a fine mixed subdivision of $\mathcal{A}$ are called *fine mixed cells* of the subdivision. It is known [7] that the sum of the volumes of those fine mixed cells is equal to the *mixed volume* of the support $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$ defined as follows. Let $\mathcal{Q}_i=\mathrm{conv}\,(\mathcal{A}_i)$ for $i=1,\ldots,n$. For positive numbers $\lambda_1,\ldots,\lambda_n$, the n-dimensional volume of the Minkowski sum

$$\lambda_1\mathcal{Q}_1+\cdots+\lambda_n\mathcal{Q}_n\equiv\{\lambda_1\mathbf{q}_1+\cdots+\lambda_n\mathbf{q}_n\,|\,\mathbf{q}_i\in\mathcal{Q}_i,i=1,\ldots,n\}$$

is a homogeneous polynomial of degree n in the variables $\lambda_1,\ldots,\lambda_n$. The mixed volume of $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$, denoted by $\mathcal{M}(\mathcal{A})$, is defined to be the coefficient of $\lambda_1\times\cdots\times\lambda_n$ in this polynomial.

By the Bernshtein theory [1], the mixed volume $\mathcal{M}(\mathcal{A})$ of the support $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$ of the polynomial system $P(\mathbf{x})=(p_1(\mathbf{x}),\ldots,p_n(\mathbf{x}))$ provides an upper bound for the number of its isolated zeros in $(\mathbb{C}^*)^n$, counting multiplicities. And this bound can be reached if the coefficients of $P(\mathbf{x})$ are generic. This root count in $(\mathbb{C}^*)^n$ has been extended to root count in $\mathbb{C}^n$ in [8, 11, 12, 13, 14]. They are, in general, significantly much sharper than the classical Bézout number and its variants.

In this paper, we present a renewed algorithm for the computation of the mixed volume $\mathcal{M}(\mathcal{A})$ by first computing all the fine mixed cells of a fine mixed subdivision of the support $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$, and the mixed volume $\mathcal{M}(\mathcal{A})$ is then obtained by adding the volumes of all those fine mixed cells. A fine mixed subdivision of $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$ can usually be obtained by the following standard process [7, 9, 10]: The n-tuple $\omega=(\omega_1,\ldots,\omega_n)$ with $\omega_i:\mathcal{A}_i\to\mathbb{R}$ for each $i=1,\ldots,n$ is called a *lifting function* on $\mathcal{A}$, and ω_i is said

to *lift* $\mathcal{A}_i$ to its graph $\hat{\mathcal{A}}_i(\omega) = \{(\mathbf{q}, \omega_i(\mathbf{q})) \,|\, \mathbf{q} \in \mathcal{A}_i\} \subset \mathbb{R}^{n+1}$. This notation is extended in the obvious way: $\hat{\mathbf{q}}(\omega) = (\mathbf{q}, \omega_i(\mathbf{q}))$, $\hat{\mathcal{A}}(\omega) = (\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_n(\omega))$, etc. A *lower face* of $\hat{\mathcal{A}}(\omega) = (\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_n(\omega))$ is a face having an inner normal with positive $(n+1)$th coordinate and a *lower facet* is a lower face $\hat{C}$ with $\dim(\mathrm{conv}\,(\hat{C})) = n$. If $\omega = (\omega_1, \ldots, \omega_n)$ is chosen at random, then the collection

$$S_\omega = \{C = (C_1, \ldots, C_n) \text{ cells of } \mathcal{A} \,|\, \hat{C}(\omega) \text{ is a lower facet of } \hat{\mathcal{A}}(\omega)\}$$

gives a *fine mixed subdivision of* $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ *induced by the lifting function* ω [7].

The fine mixed cells in S_ω, the cells of type $(1, \ldots, 1)$, can be found by identifying their corresponding lower facets of $\hat{\mathcal{A}}(\omega)$ of the same type. Namely, if $\hat{C}(\omega) = (\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}'_1\}, \ldots, \{\hat{\mathbf{a}}_n, \hat{\mathbf{a}}'_n\})$ is a lower facet of $\hat{\mathcal{A}}(\omega)$ of the type $(1, \ldots, 1)$, then $C = (\{\mathbf{a}_1, \mathbf{a}'_1\}, \ldots, \{\mathbf{a}_n, \mathbf{a}'_n\})$ gives a fine mixed cell in S_ω. Existing algorithms [4, 5, 16] as well as our algorithm in the current paper all work with n-tuples of edges of the lower hulls of the input Newton polytopes. While earlier algorithms simply perform a brute-force search to find the mixed cells, our algorithm takes a more economical enumeration by finding each lower facet of $\hat{\mathcal{A}}(\omega)$ adjacent to a vertex using one *Linear Programming pivot operation* after one of them is obtained, and then marching along such vertices. We also incorporate and utilize many other special structures of the problem to speed up the algorithm. Evidenced by the numerical results on a considerable varieties of examples, our algorithm advances the speed of computation by a great margin.

The computation of the mixed volume $\mathcal{M}(\mathcal{A})$ of the support $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ for root count of the polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$ is interesting in its own right. Recently, based on this combinatorial root count, the so-called *polyhedral homotopies* were established to approximate all the isolated zeros of $P(\mathbf{x})$ by a nonlinear homotopy continuation method, which offers a dramatic improvement over the classical continuation method using linear homotopies [7, 9, 10]. When the polyhedral homotopy is employed to find all isolated zeros of $P(\mathbf{x})$, the process of locating all the fine mixed cells in a fine mixed subdivision of the support $\mathcal{A}$ during our computation of the mixed volume $\mathcal{M}(\mathcal{A})$ plays a crucially important role. It always dominates the majority of the computation of the algorithm and therefore dictates the efficiency of the method as well as the scope of its application.

2. A Basic Linear Programming Algorithm

We begin by introducing some basic terminologies and the simplex method in Linear Programming that will be used in our algorithms. They can be found in many standard Linear Programming textbooks, e.g., [2].

Consider the model problem

$$\begin{aligned} &\min \langle \mathbf{c}, \mathbf{x} \rangle \\ &\langle \mathbf{a}_i, \mathbf{x} \rangle \le b_i, \quad i = 1, \dots, m, \end{aligned} \tag{1}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\mathbf{x} = (x_1, \dots, x_n)$, $m > n$ and $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^n$. Let R be the feasible region of (1).

Let $\mathbf{x}^0$ be a nondegenerate extreme point of (1) and $J = \{j_1, \dots, j_n\}$ be the set of indices of currently *active constraints at* $\mathbf{x}^0$, that is,

$$\langle \mathbf{a}_i, \mathbf{x}^0 \rangle = b_i, \quad \text{if } i \in J,$$

and

$$\langle \mathbf{a}_i, \mathbf{x}^0 \rangle < b_i, \quad \text{if } i \notin J.$$

Let

$$D^T = [\mathbf{a}_{j_1}, \dots, \mathbf{a}_{j_n}]$$

be the matrix of the gradients of currently active constraints at $\mathbf{x}^0$. Since $\mathbf{x}^0$ is nondegenerate, D must be nonsingular. Let $D^{-1} = [\mathbf{c}_1, \dots, \mathbf{c}_n]$. Then for any $\sigma > 0$ and $1 \le k \le n$, we have

$$\begin{aligned} &\langle \mathbf{a}_i, \mathbf{x}^0 - \sigma \mathbf{c}_k \rangle = \langle \mathbf{a}_i, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{a}_i, \mathbf{c}_k \rangle = \langle \mathbf{a}_i, \mathbf{x}^0 \rangle = b_i, \quad \text{if } i \in J \backslash \{j_k\}, \\ &\langle \mathbf{a}_{j_k}, \mathbf{x}^0 - \sigma \mathbf{c}_k \rangle = \langle \mathbf{a}_{j_k}, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{a}_{j_k}, \mathbf{c}_k \rangle = b_{j_k} - \sigma < b_{j_k}, \end{aligned} \tag{2}$$

and for small $\sigma > 0$,

$$\langle \mathbf{a}_i, \mathbf{x}^0 - \sigma \mathbf{c}_k \rangle = \langle \mathbf{a}_i, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{a}_i, \mathbf{c}_k \rangle < b_i, \quad \text{for } i \notin J.$$

Thus the n edges of the feasible region R emanating from $\mathbf{x}^0$ can be represented in the form

$$\mathbf{x}^0 - \sigma \mathbf{c}_k, \quad \sigma > 0, \quad k = 1, \dots, n.$$

These edges provide possible search directions to minimize the cost function $\langle \mathbf{c}, \mathbf{x} \rangle$. Let $\mathbf{x}^1 = \mathbf{x}^0 - \sigma \mathbf{c}_i$ with $\sigma > 0$. Then the value of the cost function at $\mathbf{x}^1$ is

$$\langle \mathbf{c}, \mathbf{x}^1 \rangle = \langle \mathbf{c}, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{c}, \mathbf{c}_i \rangle,$$

and it decreases when $\langle \mathbf{c}, \mathbf{c}_i \rangle > 0$. It can be easily shown that $\mathbf{x}^0$ is an optimal solution of (1) if $\langle \mathbf{c}, \mathbf{c}_i \rangle \le 0$ for all $i = 1, \dots, n$. If some of the $\langle \mathbf{c}, \mathbf{c}_i \rangle$'s are positive, then the greatest rate of decrease of the cost function is obtained by choosing k such that

$$\langle \mathbf{c}, \mathbf{c}_k \rangle = \max\{\langle \mathbf{c}, \mathbf{c}_i \rangle | 1 \le i \le n\}.$$

Let $\mathbf{s} = \mathbf{c}_k$ be the next search direction. From (2), for all positive σ, the ith constraint is still active at $\mathbf{x}^1 = \mathbf{x}^0 - \sigma\mathbf{s}$ for every $i \in J\backslash\{j_k\}$, and the j_kth constraint becomes inactive but stays feasible. To make $\mathbf{x}^1$ feasible, we must choose $\sigma > 0$ such that

$$\langle \mathbf{a}_i, \mathbf{x}^0 - \sigma\mathbf{s}\rangle = \langle \mathbf{a}_i, \mathbf{x}^0\rangle - \sigma\langle \mathbf{a}_i, \mathbf{s}\rangle \leq b_i, \text{ for } i \notin J. \tag{3}$$

If $\langle \mathbf{a}_i, \mathbf{s}\rangle \geq 0$ for all $i \notin J$, then the inequalities in (3) are valid for all $\sigma > 0$ and thus the problem (1) is unbounded from below with no solutions. Otherwise, from (3), the largest possible σ for $\mathbf{x}^1$ to stay feasible is

$$\sigma_0 = \min\left\{ \frac{\langle \mathbf{a}_i, \mathbf{x}^0\rangle - b_i}{\langle \mathbf{a}_i, \mathbf{s}\rangle} \middle| \text{all } i \notin J \text{ with } \langle \mathbf{a}_i, \mathbf{s}\rangle < 0 \right\}.$$

Let l be the smallest integer such that

$$\sigma_0 = \frac{\langle \mathbf{a}_l, \mathbf{x}^0\rangle - b_l}{\langle \mathbf{a}_l, \mathbf{s}\rangle}.$$

Then $\mathbf{x}^1 = \mathbf{x}^0 - \sigma_0\mathbf{s}$ is a new nondegenerate extreme point of the feasible region R in (1) with reduced value of the cost function. This procedure can be continued until an optimal solution is reached or the problem is determined to be unbounded from below.

For our problem of finding all fine mixed cells of the support $\mathcal{A}$, feasible points are always available but they may not be extreme points. Suppose $\mathbf{x}^0$ is a feasible point of (1), but not a nondegenerate extreme point. Choose a nonsingular matrix $D^T = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ whose columns are either the gradients of the constraints active at $\mathbf{x}_0$ or generically chosen vectors which make D invertible. Let $D^{-1} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$, and $J = \{j_1, \ldots, j_n\}$ where $j_i = k$ if $\mathbf{d}_i = \mathbf{a}_k$ and $j_i = 0$ if $\mathbf{d}_i$ is an arbitrarily chosen column. Consider the search direction $\mathbf{c}_i$ with $j_i = 0$. If $\langle \mathbf{c}, \mathbf{c}_i\rangle > 0$ and $\sigma > 0$, then $\langle \mathbf{c}, \mathbf{x}^0 - \sigma\mathbf{c}_i\rangle = \langle \mathbf{c}, \mathbf{x}^0\rangle - \sigma\langle \mathbf{c}, \mathbf{c}_i\rangle < \langle \mathbf{c}, \mathbf{x}^0\rangle$. Thus $\mathbf{s} = \mathbf{c}_i$ gives a feasible search direction. If $\langle \mathbf{c}, \mathbf{c}_i\rangle < 0$, then $\mathbf{s} = -\mathbf{c}_i$ is a feasible search direction since $\langle \mathbf{c}, \mathbf{x}^0 - \sigma\mathbf{s}\rangle = \langle \mathbf{c}, \mathbf{x}^0\rangle + \sigma\langle \mathbf{c}, \mathbf{c}_i\rangle < \langle \mathbf{c}, \mathbf{x}^0\rangle$. On the other hand, the greatest rate of decrease of the cost function among those directions can be determined by choosing k such that

$$|\langle \mathbf{c}, \mathbf{c}_k\rangle| = \max\{|\langle \mathbf{c}, \mathbf{c}_i\rangle| \mid \text{all } i \text{ with } j_i = 0\}.$$

With $\mathbf{s} = \text{sign}\,(\langle \mathbf{c}, \mathbf{c}_k\rangle)\mathbf{c}_k$ being the next search direction and continuing the process as in the case where $\mathbf{x}^0$ is a nondegenerate extreme point described above, we replace $\mathbf{d}_k$ in D^T by a gradient $\mathbf{a}_l$ of certain constraint. If $\langle \mathbf{c}, \mathbf{c}_i\rangle = 0$ for all i with $j_i = 0$, then the search direction $\mathbf{s}$ may be chosen from the $\mathbf{c}_i$'s with $j_i \neq 0$ and proceed as before.

We summarize the simplex method outlined above in Algorithm 1 below [2]. For locating fine mixed cells in our algorithm, we only need to consider the special case of (1) with $\mathbf{c} = (1, 0, \ldots, 0)$.

Algorithm 1. Solving the model problem (1) with $\mathbf{c} = (1, 0, \ldots, 0)$.

Step 0. Choose a feasible point $\mathbf{x}^0$ of (1). Let $\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_s}$ be the linearly independent gradients of the constraints active at $\mathbf{x}^0$, $0 \le s \le n$. When $s = n$, let $D^T = [\mathbf{d}_1, \ldots, \mathbf{d}_n] = [\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_s}]$ and $J = \{j_1, \ldots, j_n\} = \{i_1, \ldots, i_s\}$. When $s < n$, let $D^T = [\mathbf{d}_1, \ldots, \mathbf{d}_n] = [\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_s}, \mathbf{d}_{s+1}, \ldots, \mathbf{d}_n]$, where $\mathbf{d}_{s+1}, \ldots, \mathbf{d}_n$ are generically chosen for which D is nonsingular, and $J = \{j_1, \ldots, j_n\} = \{i_1, \ldots, i_s, j_{s+1}, \ldots, j_n\}$ with $j_{s+1} = \cdots = j_n = 0$.

Step 1. Computation of the search direction $\mathbf{s}$.
Let $D^{-1} = [c_{ij}] = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$. If $j_i \ge 1$ for $i = 1, \ldots, n$, go to Step 1.2. Otherwise, go to Step 1.1.

Step 1.1. Determine the smallest index k such that

$$|\langle \mathbf{c}, \mathbf{c}_k \rangle| = |c_{1k}| = \max\{|\langle \mathbf{c}, \mathbf{c}_i \rangle| = |c_{1i}| \mid \text{all } i \text{ with } j_i = 0\}.$$

If $c_{1k} = 0$, go to Step 1.2. Otherwise, set $\mathbf{s} = \text{sign}\,(c_{1k})\mathbf{c}_k$ and go to Step 2.

Step 1.2. Determine the smallest index k such that

$$\langle \mathbf{c}, \mathbf{c}_k \rangle = c_{1k} = \max\{\langle \mathbf{c}, \mathbf{c}_i \rangle = c_{1i} \,|\, \text{all } i \text{ with } j_i \ge 1\}.$$

If $c_{1k} \le 0$, stop with an optimal solution $\mathbf{x}^0$. Otherwise, set $\mathbf{s} = \mathbf{c}_k$ and go to Step 2.

Step 2. Compute the maximum feasible step size σ.
If $\langle \mathbf{a}_i, \mathbf{s} \rangle \ge 0$ for $i = 1, \ldots, m$, print the message "problem is unbounded from below" and stop. Otherwise, compute the smallest index l and σ such that

$$\sigma = \frac{\langle \mathbf{a}_l, \mathbf{x}^0 \rangle - b_l}{\langle \mathbf{a}_l, \mathbf{s} \rangle} = \min \left\{ \frac{\langle \mathbf{a}_i, \mathbf{x}^0 \rangle - b_i}{\langle \mathbf{a}_i, \mathbf{s} \rangle} \,\middle|\, \text{all } i \notin J \text{ with } \langle \mathbf{a}_i, \mathbf{s} \rangle < 0 \right\},$$

and go to Step 3.

Step 3. Update.
Set $\mathbf{x}^0 := \mathbf{x}^0 - \sigma \mathbf{s}$. Replace the kth column of D^T by $\mathbf{a}_l$ and update the inverse D^{-1}. Replace the kth element of J by l. Go to Step 1.

The process of obtaining the next feasible solution from a given feasible solution with one execution round of Steps 1, 2 and 3 is called a *Linear Programming pivot operation.*

3. Locate All Lower Facets Sharing a Vertex

A *lower face* of a polytope in $\mathbb{R}^{n+1}$ is a face having an inner normal with positive $(n+1)$th coordinate and a *lower facet* is an n-dimensional lower face. Let $\mathcal{B} = \{\mathbf{a}_0, a_1, \ldots, \mathbf{a}_m\}$ be a finite subset of $\mathbb{Z}^n$ and $\omega : \mathcal{B} \to \mathbb{R}$ be a generic lifting function. Let $\hat{\mathcal{B}}(\omega) = \{\hat{\mathbf{a}} = (\mathbf{a}, \omega(\mathbf{a})) \,|\, \mathbf{a} \in \mathcal{B}\}$. One of the main steps of our algorithm is the identification of all the lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ in $\mathbb{R}^{n+1}$.

It is clear that $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ has no lower facets when $\dim(\mathrm{Span}\,(\mathcal{B})) < n$. Therefore, we assume $\dim(\mathrm{Span}\,(\mathcal{B})) = n$. To find all the lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$, our strategy is, for each fixed $\hat{\mathbf{a}}_i$, $i = 0, 1, \ldots, m-1$, we look for all the lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ which contains $\hat{\mathbf{a}}_i$ as a vertex. When a particular point $\hat{\mathbf{a}}_i$ is considered, by parallel transformation of the axes, we may assume $\hat{\mathbf{a}}_i = \mathbf{0} \in \mathbb{R}^{n+1}$. Therefore, we will let $\mathcal{B} = \{\mathbf{0}, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$ and describe our algorithm for finding all lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ which contain $\mathbf{0}$ as a vertex.

3.1. Locate the First Lower Facet

Since the lifting function ω is generic, each lower facet of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ contains exactly $n+1$ points in $\hat{\mathcal{B}}(\omega)$. For each lower facet of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ with inner normal $(\alpha, 1) = (\alpha_1, \ldots, \alpha_n, 1)$, we have

$$0 \le \langle \hat{\mathbf{a}}_i, (\alpha, 1) \rangle, \quad i = 1, 2, \ldots, m,$$

or, equivalently,

$$\langle -\mathbf{a}_i, \alpha \rangle \le \omega(\mathbf{a}_i), \quad i = 1, 2, \ldots, m, \tag{4}$$

and the equality is reached when $\hat{\mathbf{a}}_i$ is a vertex of the facet. On the other hand, if $\alpha \in \mathbb{R}^n$ satisfies (4) with exactly n equalities, say,

$$\langle -\mathbf{a}_{i_j}, \alpha \rangle = \omega(\mathbf{a}_{i_j}), \quad j = 1, \ldots, n,$$

and $\{\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_n}\}$ is linearly independent, then $\mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}_{i_1}, \ldots, \hat{\mathbf{a}}_{i_n}\})$ gives a lower facet of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ with inner normal $(\alpha, 1)$.

The existence of $\alpha \in \mathbb{R}^n$ which satisfies inequalities in (4) is known as the feasibility test in Linear Programming. It can be dealt with by considering the linear optimization problem:

$$\begin{aligned} \min\ & \varepsilon \\ -\varepsilon \ &\le\ 0 \\ \langle -\mathbf{a}_i, \alpha \rangle - \varepsilon \ &\le\ \omega(\mathbf{a}_i), \quad i = 1, 2, \ldots, m \end{aligned} \tag{5}$$

with the variables $(\varepsilon, \alpha) = (\varepsilon, \alpha_1, \ldots, \alpha_n)$. Obviously, if $(\varepsilon, \alpha) = (0, \alpha^*)$ for certain $\alpha^* \in \mathbb{R}^n$ is an optimal solution of this problem, then all the inequalities in (4) are valid, or *feasible*, with $\alpha = \alpha^*$. For an optimal solution $(\varepsilon^*, \alpha^*)$ of the problem with $\varepsilon^* > 0$, there exists $1 \le i_0 \le m$ such that

$$\langle -\mathbf{a}_{i_0}, \alpha \rangle \ge \omega(\mathbf{a}_{i_0}) + \varepsilon^*$$

for all $\alpha \in \mathbb{R}^n$. Consequently, inequalities in (4) become infeasible, or no $\alpha \in \mathbb{R}^n$ can make them all valid. Therefore, we may determine a lower facet of conv $(\hat{\mathcal{B}}(\omega))$ with $\mathbf{0}$ as one of its vertices by looking for the optimal solution $(0, \alpha^*)$ of the linear optimization problem in (5).

To solve problem (5), let

$$\varepsilon_0 = \max\{0, -\omega(\mathbf{a}_1), \ldots, -\omega(\mathbf{a}_m)\}.$$

It is clear that $(\varepsilon_0, 0, \ldots, 0)$ is a feasible point of the constraints in (5). Applying Algorithm 1 in Section 2 with this initial feasible point, we obtain an optimal solution $(\varepsilon^*, \alpha^*) = (\varepsilon^*, \alpha_1^*, \ldots, \alpha_n^*)$ with its corresponding J, D^T and D^{-1}. When $\varepsilon^* = 0$, the constraint $-\varepsilon \le 0$ is active at the optimal solution. We may therefore assume that the gradient $(-1, 0, \ldots, 0)$ of this constraint is the first column of D^T, and write

$$(6) \qquad D^{-1} = [\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n] = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ * & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ * & * & \cdots & * \end{bmatrix} = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ * & & & \\ \vdots & \bar{\mathbf{b}}_1 & \cdots & \bar{\mathbf{c}}_n \\ * & & & \end{bmatrix}.$$

Now, if the rest of the n columns of D^T are all gradients of the constraints in (5), say $(-1, \mathbf{a}_{i_1}), \ldots, (-1, \mathbf{a}_{i_n})$, then as discussed before, conv $(\{\mathbf{0}, \hat{\mathbf{a}}_{\mathbf{i_1}}, \ldots, \hat{\mathbf{a}}_{\mathbf{i_n}}\})$ gives a lower facet of conv $(\hat{\mathcal{B}}(\omega))$. If some of the columns of D^T are not gradients of the constraints in (5), let's choose any one of them, say the kth column $\mathbf{d}_k$. Since the first entry of $\mathbf{c}_k$ in (6) is zero and the direction of the cost function ε is $\mathbf{c} = (1, 0, \ldots, 0)$, moving in either direction of $\mathbf{c}_k = (0, \bar{\mathbf{c}}_k)$ will keep the cost function ε staying at the minimal value 0. Furthermore, since $\dim(\text{Span}\,(\mathcal{B})) = n$, there exists a constraint gradient $(-1, \mathbf{a}_i) \notin \{$the columns of $D^T\}$ such that

$$\langle (-1, -\mathbf{a}_i), \mathbf{c}_k \rangle = \langle (-1, -\mathbf{a}_i), (0, \bar{\mathbf{c}}_k) \rangle = \langle -\mathbf{a}_i, \bar{\mathbf{c}}_k \rangle \ne 0,$$

say, $\langle -\mathbf{a}_i, \bar{\mathbf{c}}_k \rangle < 0$. Then, apparently the pivot operation can be carried out further by moving in the positive direction of $\mathbf{c}_k$ and remaining feasible, and an appropriate gradient $(-1, \mathbf{a}_l)$ can be found to replace $\mathbf{d}_k$ in D^T. This process may be continued until the columns of D^T, starting from the second one, all

become constraint gradients in (5). By then, a lower facet of conv $(\hat{\mathcal{B}}(\omega))$, containing $\mathbf{0}$ as one of its vertices, is located.

Summarizing the above discussion, we now modify Algorithm 1 described in the last section to serve our purpose.

Algorithm 2. Find the first lower facet of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$.

First apply Algorithm 1 to the optimization problem (5). When a resulting optimal solution $\mathbf{x}^0 = (\varepsilon^*, \alpha^*)$ is obtained in Step 1.2 of Algorithm 1, if $\varepsilon^* > 0$, then no lower face of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ exists, and stop. If $\varepsilon^* = 0$, then execute

Step 4. If all elements of J are positive, a lower facet of conv $(\hat{\mathcal{B}}(\omega))$ is produced, and stop. Otherwise, some elements of J is zero, say, $j_k = 0$. Go to Step 4.1.

Step 4.1. If $\langle(-1, -\mathbf{a}_i), \mathbf{c}_k\rangle = 0$ for all $i = 1, \ldots, m$, then $\dim(\text{Span}\,(\mathcal{B})) < n$, and no lower facet of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ exists. The face conv $(\{\mathbf{0}\} \cup \bigcup_{\substack{j_i \in J \\ j_i \neq 0}} \hat{\mathbf{a}}_{j_i})$ corresponding to the constraints currently active at $\mathbf{x}^0$ is the largest lower face of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ which contains all the lower edges of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$. Store this face, and stop.
If $\langle(-1, -\mathbf{a}_i), \mathbf{c}_k\rangle < 0$ for some i, let $\mathbf{s} = \mathbf{c}_k$. Otherwise, let $\mathbf{s} = -\mathbf{c}_k$. Compute the smallest index l and σ such that

$$\sigma = \frac{\langle(-1, -\mathbf{a}_l), \mathbf{x}^0\rangle - \omega(\mathbf{a}_l)}{\langle(-1, -\mathbf{a}_l), \mathbf{s}\rangle}$$
$$= \min\left\{ \frac{\langle(-1, -\mathbf{a}_i), \mathbf{x}^0\rangle - \omega(\mathbf{a}_i)}{\langle(-1, -\mathbf{a}_i), \mathbf{s}\rangle} \;\middle|\; \text{all } i \notin J \text{ with } \langle(-1, -\mathbf{a}_i), \mathbf{s}\rangle < 0 \right\},$$

and go to Step 4.2.

Step 4.2. Update.
Set $\mathbf{x}^0 := \mathbf{x}^0 - \sigma\mathbf{s}$. Replace kth column of D^T by $(-1, -\mathbf{a}_l)$ to update the inverse D^{-1}. Replace the kth element of J by l. Go to Step 4.

3.2. Generate All Lower Facets Sharing Vertex 0

Recall the equivalence of the existence of an optimal solution in (5) with 0 optimal value to the feasibility of the inequalities

$$\langle -\mathbf{a}_i, \alpha\rangle \leq \omega(\mathbf{a}_i), \quad i = 1, \ldots, m, \tag{7}$$

in (4) as described in the last section. Let $\mathcal{H}$ be the polyhedron defined by the half spaces in (7). For $\alpha \in \mathcal{H}$, if

$$\langle -\mathbf{a}_k, \alpha \rangle = \omega(\mathbf{a}_k) \quad \text{for } k \in \{1, \ldots, m\},$$

we say the *constraint* $\mathbf{a}_k$ *is active at* α, or α has *active constraint* $\mathbf{a}_k$, and denote by $\mathrm{Act}\,(\alpha)$ the set of active constraints of α. Since the lifting function ω is generically chosen, any vertex α^* of $\mathcal{H}$ is *nondegenerate* in the sense that α^* has exactly n linearly independent active constraints, $\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_n}$, and consequently, $\mathrm{conv}\,\{\mathbf{0}, \hat{\mathbf{a}}_{j_1}, \ldots, \hat{\mathbf{a}}_{j_n}\}$ is a lower facet of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ with inner normal $(\alpha^*, 1)$. On the other hand, when a lower facet $\hat{C}_0^*$ of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ with inner normal $(\alpha^*, 1)$ is produced by Algorithm 2 in the last section, α^* becomes a vertex of $\mathcal{H}$. Thus, to generate all other lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ from $\hat{C}_0^*$ is equivalent to generating all the vertices of $\mathcal{H}$ from the vertex α^*. This task can be carried out by employing Algorithm 1 in Section 2 with certain modifications. Without loss of generality, write $\hat{C}_0^* = \mathrm{conv}\,\{\mathbf{0}, \hat{\mathbf{a}}_\mathbf{1}, \ldots, \hat{\mathbf{a}}_\mathbf{n}\}$. Then the basis matrix corresponding to the optimal solution $(0, \alpha^*)$ in (5) can be written as

$$D^T = \begin{bmatrix} -1 & -1 & \cdots & -1 \\ \mathbf{0} & -\mathbf{a}_1 & \cdots & -\mathbf{a}_n \end{bmatrix}$$

with

$$D^{-1} = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ \mathbf{c}_0 & \mathbf{c}_1 & \cdots & \mathbf{c}_n \end{bmatrix}.$$

To use Algorithm 1 to find all the vertices of $\mathcal{H}$, the cost function $\langle \mathbf{c}, \mathbf{x} \rangle$ in (1) which determines the possible search direction as well as the termination of the algorithm no longer exists. Thus, at the vertex α^* with $\bar{J} = \{1, \ldots, n\}$, $\bar{D}^T = [-\mathbf{a}_1, \ldots, -\mathbf{a}_n]$ and $\bar{D}^{-1} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$, we will search for other vertices of $\mathcal{H}$ along each $\mathbf{c}_i$, $i = 1, \ldots, n$, direction. For each fixed $\mathbf{c}_i$, $i = 1, \ldots, n$, if there exists $j \geq n+1$ such that

$$\langle -\mathbf{a}_j, \mathbf{c}_i \rangle < 0, \tag{8}$$

then with

$$\sigma_0 \equiv \frac{\langle -\mathbf{a}_l, \alpha^* \rangle - \omega(\mathbf{a}_l)}{\langle -\mathbf{a}_l, \mathbf{c}_i \rangle} = \min \left\{ \frac{\langle -\mathbf{a}_j, \alpha^* \rangle - \omega(\mathbf{a}_j)}{\langle -\mathbf{a}_j, \mathbf{c}_i \rangle} \middle| \, j \geq n+1 \text{ and } \langle -\mathbf{a}_j, \mathbf{c}_i \rangle < 0 \right\},$$

$\alpha^{**} = \alpha^* - \sigma_0 \mathbf{c}_i$ becomes a new vertex of $\mathcal{H}$ with $\bar{J} \leftarrow (\bar{J} \backslash \{i\}) \cup \{l\}$ and

$$\bar{D}^T = [-\mathbf{a}_1, \ldots, -\mathbf{a}_{i-1}, -\mathbf{a}_l, -\mathbf{a}_{i+1}, \ldots, -\mathbf{a}_n].$$

Of course, the search terminates along $\mathbf{c}_i$ direction if (8) fails for all $j \geq n+1$.

This search procedure may be applied to every newly discovered vertex of $\mathcal{H}$ and the algorithm terminates when all possible search directions from all existing vertices of $\mathcal{H}$ are exhausted. In the process, one may easily reproduce an already existed vertex. This wasteful computation can be prevented based on the following observations.

Let $\alpha^{(1)}$ and $\alpha^{(2)}$ be two vertices of $\mathcal{H}$ with active constraints $\mathrm{Act}(\alpha^{(1)}) = \{\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_n}\}$ and $\mathrm{Act}(\alpha^{(2)}) = \{\mathbf{a}_{j_2}, \ldots, \mathbf{a}_{j_{n+1}}\}$, respectively, namely, they share $n-1$ active constraints.

Lemma 1. *For any* $\alpha \in \mathcal{H}$, $\mathrm{Act}\,(\alpha) \supset \mathrm{Act}\,(\alpha^{(1)}) \cap \mathrm{Act}\,(\alpha^{(2)}) = \{\mathbf{a}_{j_2}, \ldots, \mathbf{a}_{j_n}\}$ *if and only if* $\alpha \in \mathrm{conv}\,(\alpha^{(1)}, \alpha^{(2)})$.

Proof. (1) $\Rightarrow$) Since both $\alpha - \alpha^{(1)}$ and $\alpha^{(2)} - \alpha^{(1)}$ belong to the kernel of the matrix $[\mathbf{a}_{j_2}, \ldots, \mathbf{a}_{j_n}]$ which is one-dimensional, so

$$\alpha - \alpha^{(1)} = t(\alpha^{(2)} - \alpha^{(1)}) \quad \text{for some } t \in \mathbb{R},$$

or,

$$\alpha = \alpha^{(1)} + t(\alpha^{(2)} - \alpha^{(1)}) = (1-t)\alpha^{(1)} + t\alpha^{(2)}.$$

But,

$$\begin{aligned} \langle -\mathbf{a}_{j_1}, \alpha \rangle &= (1-t)\langle -\mathbf{a}_{j_1}, \alpha^{(1)} \rangle + t\langle -\mathbf{a}_{j_1}, \alpha^{(2)} \rangle \\ &= \omega(\mathbf{a}_{j_1}) + t(\langle -\mathbf{a}_{j_1}, \alpha^{(2)} \rangle - \omega(\mathbf{a}_{j_1})) \leq \omega(\mathbf{a}_{j_1}). \end{aligned}$$

So, $t \geq 0$ since $\langle -\mathbf{a}_{j_1}, \alpha^{(2)} \rangle < \omega(\mathbf{a}_{j_1})$. Similarly,

$$\begin{aligned} \langle -\mathbf{a}_{j_{n+1}}, \alpha \rangle &= (1-t)\langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)} \rangle + t\langle -\mathbf{a}_{j_{n+1}}, \alpha^{(2)} \rangle \\ &= (1-t)\langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)} \rangle + t\omega(\mathbf{a}_{j_{n+1}}) \leq \omega(\mathbf{a}_{j_{n+1}}), \end{aligned}$$

or,

$$(1-t)(\langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)} \rangle - \omega(\mathbf{a}_{j_{n+1}})) < 0.$$

Hence, $t \leq 1$ since $\langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)} \rangle < \omega(\mathbf{a}_{j_{n+1}})$, and $\alpha \in \mathrm{conv}\,(\alpha^{(1)}, \alpha^{(2)})$.

(2) $\Leftarrow$) It is obvious. ∎

Lemma 2. $\alpha^{(1)}$ *and* $\alpha^{(2)}$ *are the only two vertices of* $\mathcal{H}$ *which share the active constraints* $\mathrm{Act}\,(\alpha^{(1)}) \cap \mathrm{Act}\,(\alpha^{(2)})$.

Proof. The existence of an extra vertex α of $\mathcal{H}$ active at $\mathrm{Act}\,(\alpha^{(1)}) \cap \mathrm{Act}\,(\alpha^{(2)})$ would imply, by Lemma 1,

$$\alpha = (1-t_0)\alpha^{(1)} + t_0\alpha^{(2)} \quad \text{for some } t_0 \in [0,1]$$

and

$$\alpha^{(1)} = (1 - t_1)\alpha^{(2)} + t_1\alpha \quad \text{for some } t_1 \in [0, 1].$$

This would result in $\alpha^{(1)} = \alpha^{(2)}$. ■

Let $D_1^T = [-\mathbf{a}_{j_1}, \ldots, -\mathbf{a}_{j_n}]$, $D_2^T = [-\mathbf{a}_{j_2}, \ldots, -\mathbf{a}_{j_{n+1}}]$ and $D_1^{-1} = [\mathbf{c}_{j_1}, \ldots, \mathbf{c}_{j_n}]$, $D_2^{-1} = [\mathbf{c}'_{j_2}, \ldots, \mathbf{c}'_{j_{n+1}}]$.

Lemma 3.

(1) $\mathbf{c}_{j_1} = -k\mathbf{c}'_{j_{n+1}} \quad for\ k \geq 0,$
(2) $\langle -\mathbf{a}_{j_{n+1}}, \mathbf{c}_{j_1}\rangle < 0\ and\ \langle -\mathbf{a}_{j_1}, \mathbf{c}'_{j_{n+1}}\rangle < 0.$

Proof. (1) Since both $\mathbf{c}_{j_1}$ and $\mathbf{c}'_{j_{n+1}}$ are in the kernel of the matrix $[-\mathbf{a}_{j_2}, \ldots, -\mathbf{a}_{j_n}]$, we have

$$\alpha^{(1)} - \alpha^{(2)} = l_1\mathbf{c}_{j_1} = l_2\mathbf{c}'_{j_{n+1}}.$$

Now,

$$l_1 = l_1\langle -\mathbf{a}_{j_1}, \mathbf{c}_{j_1}\rangle = \langle -\mathbf{a}_{j_1}, \alpha^{(1)} - \alpha^{(2)}\rangle = \omega(\mathbf{a}_{j_1}) - \langle -\mathbf{a}_{j_1}, \alpha^{(2)}\rangle > 0$$

and

$$l_2 = l_2\langle -\mathbf{a}_{j_{n+1}}, \mathbf{c}'_{j_{n+1}}\rangle = \langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)} - \alpha^{(2)}\rangle = \langle -\mathbf{a}_{j_{n+1}}, \alpha^{(1)}\rangle - \omega(\mathbf{a}_{j_{n+1}}) < 0.$$

So, $\mathbf{c}_{j_1} = -k\mathbf{c}'_{j_{n+1}}$ for $k = -l_2/l_1 > 0$.

(2) $\langle -\mathbf{a}_{j_{n+1}}, \mathbf{c}_{j_1}\rangle = -k\langle -\mathbf{a}_{j_{n+1}}, \mathbf{c}'_{j_{n+1}}\rangle = -k < 0,$
and $\langle -\mathbf{a}_{j_1}, \mathbf{c}_{j_{n+1}}\rangle = -\langle -\mathbf{a}_{j_1}, \mathbf{c}_{j_1}\rangle/k = -(1/k) < 0.$ ∎

It follows from (2) in the above lemma that if one starts from the vertex $\alpha^{(1)}$, searches along $\mathbf{c}_{j_1}$ direction for the replacement of the active constraint $\mathbf{a}_{j_1}$ and arrives at the vertex $\alpha^{(2)}$, then from $\alpha^{(2)}$, $\mathbf{c}'_{j_{n+1}}$ will surely be a successful search direction, namely, constraint $\mathbf{a}_{j_{n+1}}$ can be replaced, because (8) is fulfilled. And, when $\mathbf{a}_{j_{n+1}}$ is replaced, the resulting vertex obtained must be $\alpha^{(1)}$ by Lemma 2.

Therefore, for a given vertex α of $\mathcal{H}$ with $\mathrm{Act}\,(\alpha) = \{\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_n}\}$, every vertex of $\mathcal{H}$ adjacent to α can be generated by one Linear Programming pivot operation. To eliminate waste of computation, when a new vertex is produced by replacing $\mathbf{a}_{j_i}$ for certain $1 \leq i \leq n$, one must store the set of constraints $\mathrm{Act}\,(\alpha)\backslash\{\mathbf{a}_{j_i}\}$ in a counter, say, A_ω. Then, before pursuing the replacement of any active constraint of a vertex, we must verify the existence of the remaining active constraints of the vertex, as a unit, in A_ω. When a positive answer is pronounced, the proceeding of the replacement should be prohibited to prevent the reproduction of existing vertices.

When the algorithm terminates, we will obtained *all* vertices of $\mathcal{H}$ since for any two vertices $\alpha', \alpha'' \in \mathcal{H}$, there exists a chain of vertices $\alpha^{(1)}, \ldots, \alpha^{(k)}$ in $\mathcal{H}$ with $\alpha^{(1)} = \alpha'$, $\alpha^{(k)} = \alpha''$ and each line segment $\mathrm{conv}\,(\alpha^{(i)}, \alpha^{(i+1)})$ for $i = 1, \ldots, k-1$ is an edge of $\mathcal{H}$. (While this assertion is intuitively clear, a rigorous proof is given in the appendix.) Then, by Lemma 1, the number of constraints in $\mathrm{Act}\,(\alpha^{(i)}) \cap \mathrm{Act}\,(\alpha^{(i+1)})$ equals $n-1$ for each $i = 1, \ldots, k-1$

and, as a consequence of Lemma 3, the algorithm may reach $\alpha^{(i+1)}$ from $\alpha^{(i)}$ for each i and therefore *connects* α' and α''.

We summarize the above discussion in the following algorithm.

Algorithm 3. Enumerate all lower facets of conv $(\hat{\mathcal{B}}(\omega))$ sharing the vertex $\mathbf{0}$.

Step 1. Apply algorithm 2 to the linear optimization problem (5). If no lower facet of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ exists, stop. Otherwise, the optimal solution $(\varepsilon^*, \alpha^*)$ corresponding to 0 optimal value gives the first lower facet of conv $(\hat{\mathcal{B}}(\omega))$ having vertex $\mathbf{0}$ with inner normal $(\alpha^*, 1)$. Equivalently, α^* is also the first vertex of the polyhedron $\mathcal{H}$ defined by the half spaces in (7) being found. Let $N_\omega = \{\alpha^*\}$, $F_\omega = \emptyset$ and $A_\omega = \emptyset$. Go to Step 2.

Step 2. Pick a point α, a vertex of $\mathcal{H}$, from N_ω. Let Act $(\alpha) = \{\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_n}\}$ and $D^{-1} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$ be the inverse of the matrix $D^T = [-\mathbf{a}_{j_1}, \ldots, -\mathbf{a}_{j_n}]$.

Step 2.1. For each fixed $\mathbf{c}_k$, $k = 1, \ldots, n$, if Act $(\alpha)\backslash\{\mathbf{a}_{j_k}\} \notin A_\omega$ and in the meantime,

$$\langle -\mathbf{a}_j, \mathbf{c}_k \rangle < 0$$

for certain $\mathbf{a}_j \notin$ Act (α), then with

$$\begin{aligned} \sigma_0 &\equiv \frac{\langle -\mathbf{a}_l, \alpha \rangle - \omega(\mathbf{a}_l)}{\langle -\mathbf{a}_l, \mathbf{c}_k \rangle} \\ &= \min \left\{ \frac{\langle -\mathbf{a}_j, \alpha \rangle - \omega(\mathbf{a}_j)}{\langle -\mathbf{a}_j, \mathbf{c}_k \rangle} \middle| \mathbf{a}_j \notin \text{Act}\,(\alpha) \text{ and } \langle -\mathbf{a}_j, \mathbf{c}_k \rangle < 0 \right\}, \end{aligned}$$

$\alpha' = \alpha - \sigma_0 \mathbf{c}_k$ becomes a new vertex of $\mathcal{H}$. Save α' in N_ω and Act $(\alpha)\backslash\{\mathbf{a}_{i_k}\}$, as a unit, in A_ω.

Step 2.2. Save α and Act (α), as a unit, in F_ω and delete α from N_ω. If N_ω is empty, stop. F_ω contains all the vertices of $\mathcal{H}$, or all the lower facets of $\hat{\mathcal{B}}(\omega)$ have vertex $\mathbf{0}$. Otherwise, go to Step 2.

4. Find All Fine Mixed Cells

For the support $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ of a polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$, write $\mathcal{A}_i = \{\mathbf{a}_{i1}, \ldots, \mathbf{a}_{im_i}\}$, $m_i \geq 2$, for $i = 1, \ldots, n$. Recall that with a random lifting function $\omega = (\omega_1, \ldots, \omega_n)$ which lifts $\mathcal{A}$ to $\hat{\mathcal{A}}(\omega) = (\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_n(\omega))$, the collection

$$S_\omega = \{C = (C_1, \ldots, C_n) \text{ cells of } \mathcal{A} \,|\, \hat{C}(\omega) \text{ is a lower facet of } \hat{\mathcal{A}}(\omega)\}$$

yields a fine mixed subdivision of $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$. Our goal in this section is to find all the fine mixed cells, the cells of type $(1, \cdots, 1)$, in S_ω.

For a cell $C = (C_1, \ldots, C_n)$ of type $(1, \ldots, 1)$ in S_ω, write

$$C = (C_1, \ldots, C_n) = (\{\mathbf{a}_1, \mathbf{a}_1'\}, \ldots, \{\mathbf{a}_n, \mathbf{a}_n'\}),$$

where $\{\mathbf{a}_i, \mathbf{a}_i'\} \subset \mathcal{A}_i$, $i = 1, \ldots, n$. Then,

$$\hat{C}(\omega) = (\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_1'\}, \ldots, \{\hat{\mathbf{a}}_n, \hat{\mathbf{a}}_n'\})$$

is a lower facet of $\hat{\mathcal{A}}(\omega) = (\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_n(\omega))$. It follows that, for fixed $\hat{\mathbf{a}}_1 \in \hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathbf{a}}_n \in \hat{\mathcal{A}}_n(\omega)$, the cell

$$\hat{C}_0(\omega) = (\{\mathbf{0}, \hat{\mathbf{a}}_1' - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}_n' - \hat{\mathbf{a}}_n\})$$

forms a lower facet of $\hat{\mathcal{B}}(\omega) = (\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_n(\omega))$, where

$$\hat{\mathcal{B}}_i(\omega) = \{\hat{\mathbf{a}}_{ij} - \hat{\mathbf{a}}_i \,|\, j = 1, \ldots, m_i\}, \quad i = 1, \ldots, n. \tag{9}$$

This means

$$\mathrm{conv}\,(\hat{C}_0(\omega)) = \mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}_1' - \hat{\mathbf{a}}_1\}) + \cdots + \mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}_n' - \hat{\mathbf{a}}_n\})$$

is a lower facet of the polytope

$$\mathrm{conv}\,(\hat{\mathcal{B}}(\omega)) = \mathrm{conv}\,(\hat{\mathcal{B}}_1(\omega)) + \cdots + \mathrm{conv}\,(\hat{\mathcal{B}}_n(\omega)).$$

Notice that $\mathbf{0}$ is a vertex of both $\mathrm{conv}\,(\hat{C}_0(\omega))$ and $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$.

So, to find all the fine mixed cells in S_ω, we may proceed as follows. For any fixed $(\mathbf{a}_1, \ldots, \mathbf{a}_n) \in (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ and its corresponding $\hat{\mathcal{B}}(\omega) = (\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_n(\omega))$ as in (9), we search for all the lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ with their vertices consisting of $\{\mathbf{0}, \bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_n\}$, where $\bar{\mathbf{b}}_i \in \hat{\mathcal{B}}_i(\omega)$ for each $i = 1, \ldots, n$. Then, with $\hat{\mathbf{a}}_i' = \hat{\mathbf{a}}_i + \bar{\mathbf{b}}_i$, $i = 1, \ldots, n$, the cell

$$C = (\{\mathbf{a}_1, \mathbf{a}_1'\}, \ldots, \{\mathbf{a}_n, \mathbf{a}_n'\})$$

gives a fine mixed cell in S_ω. Evidently, one must carefully organize the procedure for this search in order to avoid potentially huge amount of wasteful computations. In the following, we will present our approach to locate all the lower facets of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ in a proper order so that all the fine mixed cells in S_ω can be found most efficiently.

For a fixed combination $(\mathbf{a}_1, \ldots, \mathbf{a}_n) \in (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ together with its corresponding $\hat{\mathcal{B}}(\omega) = (\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_n(\omega))$ as in (9), consider, for $k < n$, the lower faces of $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_k(\omega))$ of the type

$$\hat{C}_{0,k} = (\{\mathbf{0}, \hat{\mathbf{a}}_1' - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}_k' - \hat{\mathbf{a}}_k\}),$$

where $\hat{\mathbf{a}}'_i \in \hat{\mathcal{A}}_i(\omega)$ for each $i = 1, \ldots, k$. We say $\hat{C}_{0,k}$ is *extendable* if there exists $\hat{\mathbf{a}}'_{k+1} \in \hat{\mathcal{A}}_{k+1}$ for which the cell

$$\hat{C}_{0,k+1} := (\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}'_{k+1} - \hat{\mathbf{a}}_{k+1}\})$$

is a lower face of $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_{k+1}(\omega))$. And the corresponding lower face $\hat{C}_k = (\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}'_1\}, \ldots, \{\hat{\mathbf{a}}_k, \hat{\mathbf{a}}'_k\})$ of $(\hat{\mathcal{A}}_1, \ldots, \hat{\mathcal{A}}_k)$ is said to be extended to a lower face $\hat{C}_{k+1} = (\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}'_1\}, \ldots, \{\hat{\mathbf{a}}_{k+1}, \hat{\mathbf{a}}'_{k+1}\})$ of $(\hat{\mathcal{A}}_1, \ldots, \hat{\mathcal{A}}_{k+1})$. Obviously, when $k = n-1$, an extendable cell $\hat{C}_{0,k}$ yields fine mixed cells in S_ω with its extended cells (possibly several). Thus, our search procedure will mainly concentrate on the possibility of extending all the one-dimensional lower faces of $\hat{\mathcal{B}}_1(\omega)$ of the type $\mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\})$, step by step, to $k = n$, so that the cell

$$\hat{C}_{0,n} = (\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}'_n - \hat{\mathbf{a}}_n\})$$

for certain $(\hat{\mathbf{a}}'_1, \ldots, \hat{\mathbf{a}}'_n) \in (\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_n(\omega))$ becomes a lower facet of $\mathrm{conv}\,(\hat{\mathcal{B}}(\omega))$ and a fine mixed cell

$$C_n = (\{\mathbf{a}_1, \mathbf{a}'_1\}, \ldots, \{\mathbf{a}_n, \mathbf{a}'_n\})$$

of S_ω is thus obtained.

When $\mathcal{A}_1$ has more than n points, namely, $m_1 > n$, then one-dimensional lower faces of $\hat{\mathcal{B}}_1(\omega)$ of the type $\mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\})$ are available when lower facets of $\hat{\mathcal{B}}_1(\omega)$ with vertex $\mathbf{0}$ are found by applying Algorithm 3 to $\hat{\mathcal{B}}_1(\omega)$. If $m_1 \le n$, we will start our extension procedure from each edge $\mathrm{conv}\,(\{\mathbf{0}, \hat{\mathbf{a}}_{1j} - \hat{\mathbf{a}}_1\})$ for $j = 1, \ldots, m_1$ with $\mathbf{a}_{1j} \neq \mathbf{a}_1$.

Now, for $k \ge 1$, for the extendability of the cell

$$\hat{C}_{0,k} = (\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}'_k - \hat{\mathbf{a}}_k\}),$$

consider the linear optimization problem

$$\begin{aligned}
&\min\ \varepsilon \\
-\varepsilon &\le 0, \\
\langle \mathbf{a}_i - \mathbf{a}'_i, \alpha \rangle &= \omega_i(\mathbf{a}'_i) - \omega_i(\mathbf{a}_i), \quad i = 1, \ldots, k, \\
\langle \mathbf{a}_i - \mathbf{a}_{i,j}, \alpha \rangle - \varepsilon &\le \omega_i(\mathbf{a}_{i,j}) - \omega_i(\mathbf{a}_i), \\
&\qquad j = 1, \ldots, m_i, \mathbf{a}_{i,j} \neq \mathbf{a}_i, i = 1, \ldots, k, \\
\langle \mathbf{a}_{k+1} - \mathbf{a}_{k+1,j}, \alpha \rangle - \varepsilon &\le \omega_{k+1}(\mathbf{a}_{k+1,j}) - \omega_{k+1}(\mathbf{a}_{k+1}), \\
&\qquad j = 1, \ldots, m_{k+1}, \mathbf{a}_{k+1,j} \neq \mathbf{a}_{k+1}.
\end{aligned} \tag{10}$$

It is clear that an optimal solution of (10) with $\varepsilon = 0$ corresponds to a lower facet of $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_{k+1}(\omega))$ which includes $\hat{C}_{0,k}$ as its face. And among those lower facets corresponding to 0 optimal solutions only those which possess an active constraint $\mathbf{a}_{k+1} - \mathbf{a}_{k+1,j}$ for certain $j = 1, \ldots, m_{k+1}$ contain cells

$$\hat{C}_{0,k+1} = (\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}_{k+1,j} - \hat{\mathbf{a}}_{k+1}\})$$

as extended cells of $\hat{C}_{0,k}$ in $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_{k+1}(\omega))$.

To solve (10), we may eliminate k variables of α by using the first k equality constraints and use Algorithm 3 in the last section on the resulting linear optimization problem instead. If none of the 0 optimal solutions found by the algorithm contains active constraint $\mathbf{a}_{k+1} - \mathbf{a}_{k+1,j}$ for certain $j = 1, \ldots, m_{k+1}$ with $\mathbf{a}_{k+1} \neq \mathbf{a}_{k+1,j}$, then the cell $\hat{C}_{0,k}$ will be considered *nonextendable*. And, when $\hat{C}_{0,k}$ is extendable, we continue our extension attempt from each of the extended cells $\hat{C}_{0,k+1}$ until a fine mixed cell in S_ω is produced.

Algorithm 4. Given $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$, $\mathcal{A}_i = \{\mathbf{a}_{i,1}, \ldots, \mathbf{a}_{i,m_i}\}$, $m_i \geq 2$, $i = 1, \ldots, n$, find all fine mixed cells in S_ω.

Step 0. Use a random lifting ω to lift $\mathcal{A}$ to $\hat{\mathcal{A}}(\omega)$.

Step 1. Find all the lower edges $\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}'_1\}$ of $\text{conv}\,(\hat{\mathcal{A}}_1(\omega))$. Store all these edges $\{\mathbf{a}_1, \mathbf{a}'_1\}$ in C^1. Let $k = 1$. Go to Step 2.

Step 2. Pick a cell C_k from C^k, say,

$$C_k = (\{\mathbf{a}_1, \mathbf{a}'_1\}, \ldots, \{\mathbf{a}_k, \mathbf{a}'_k\}).$$

Define

$$\hat{\mathcal{B}}_i(\omega) = \{\hat{\mathbf{a}}_{i,j} - \hat{\mathbf{a}}_i \,|\, j = 1, \ldots, m_i\}, \quad i = 1, \ldots, k.$$

Go to Step 2.1.

Step 2.1. For each point $\mathbf{a}_{k+1} \in \mathcal{A}_{k+1} \backslash \{\mathbf{a}_{k+1,m_{k+1}}\}$, define

$$\hat{\mathcal{B}}_{k+1}(\omega) = \{\hat{\mathbf{a}}_{k+1,j} - \hat{\mathbf{a}}_{k+1} \,|\, j = 1, \ldots, m_{k+1}\}.$$

Extend $\hat{C}_k$ to all possible lower faces $\hat{C}_{k+1} = (\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}'_1\}, \ldots, \{\hat{\mathbf{a}}_{k+1}, \hat{\mathbf{a}}'_{k+1}\})$ of $(\hat{\mathcal{A}}_1(\omega), \ldots, \hat{\mathcal{A}}_{k+1}(\omega))$ by applying Algorithm 3 to the optimization problem in (10) to find all possible lower facets of $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_{k+1}(\omega))$ containing the cell $\hat{C}_{0,k} = (\{\mathbf{0}, \hat{\mathbf{a}}'_1 - \hat{\mathbf{a}}_1\}, \ldots, \{\mathbf{0}, \hat{\mathbf{a}}'_k - \hat{\mathbf{a}}_k\})$. When the dimension of the polytope $\text{conv}\,(\hat{\mathcal{B}}_1(\omega)) + \cdots + \text{conv}\,(\hat{\mathcal{B}}_{k+1}(\omega))$ is less than n, find the largest lower face of $(\hat{\mathcal{B}}_1(\omega), \ldots, \hat{\mathcal{B}}_{k+1}(\omega))$ containing the cell $\hat{C}_{0,k}$. Store these new lower faces C_{k+1} found in C^{k+1}. Go to Step 2.2.

Step 2.2. Delete the cell C_k from C^k. If $k+1=n$, go to Step 2.3. If $C^{k+1}\neq\emptyset$, then let $k:=k+1$ and go to Step 2. Otherwise, go to Step 2.3.

Step 2.3. Find the largest index i with $1\leq i\leq k$ such that C^i is not empty. If no such i exists, then all fine mixed cells in the fine mixed subdivision S_ω of $\mathcal{A}$ induced by the lifting ω have been found and stored in C^n, stop. Otherwise, let $k=i$, and go to Step 2.

Remark 1. There are several strategies we may use in Step 2.1 to speed up the computation. Here, we list several most important ones.

1. When the algorithm moves from Step 2.2 directly to Step 2, the points in $\hat{\mathcal{A}}_k(\omega)$ which never appear in C^k should not be used to form constraints in (10) in Step 2.1 since these points never enter the lower faces when we intend to extend $\hat{C}_k$.
2. Since elimination of variables is used in every extension attempt, the elimination result from previous steps should be carried over to the current step.
3. When extending a cell $\hat{C}_k$ in C^k to all possible lower faces $\hat{C}_{k+1}$ of $(\hat{\mathcal{A}}_1(\omega),\ldots,\hat{\mathcal{A}}_{k+1}(\omega))$ which contains $\hat{\mathbf{a}}_{k+1}\in\hat{\mathcal{A}}_{k+1}(\omega)$ in Step 2.1, and when a lower facet of $(\hat{\mathcal{B}}_1(\omega),\ldots,\hat{\mathcal{B}}_{k+1}(\omega))$ containing an edge $\{\mathbf{0},\hat{\mathbf{a}}'_{k+1}-\hat{\mathbf{a}}_{k+1}\}$ with $\hat{\mathbf{a}}'_{k+1}\in\hat{\mathcal{A}}_{k+1}(\omega)$ is identified, then J and D^{-1} associated with the optimal solution should be stored if the extension attempt of $\hat{C}_k$ to lower faces $\hat{C}_{k+1}$ of $(\hat{\mathcal{A}}_1(\omega),\ldots,\hat{\mathcal{A}}_{k+1}(\omega))$ which contain $\hat{\mathbf{a}}'_{k+1}$ has not been performed. So, when we intend to extend $\hat{C}_k$ to lower faces $\hat{C}_{k+1}$ which contain $\hat{\mathbf{a}}'_{k+1}$, the first lower facet of $(\hat{\mathcal{B}}_1(\omega),\ldots,\hat{\mathcal{B}}_{k+1}(\omega))$ with
$$\hat{\mathcal{B}}_{k+1}(\omega)=\{\hat{\mathbf{a}}_{k+1,j}-\hat{\mathbf{a}}'_{k+1}\,|\,j=1,\ldots,m_{k+1}\}$$
is available from J and D^{-1} in storage. Thus, the calculation of Step 1 of Algorithm 3 becomes unnecessary.

5. The Computation of the Mixed Volumes and Numerical Results

As mentioned in Section 1, it is known [7, 9] that the mixed volume of the support $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_n)$ can be computed by the summation of the volumes of all fine mixed cells in the fine mixed subdivision S_ω induced by a generic lifting $\omega=(\omega_1,\ldots,\omega_n)$ on $\mathcal{A}$. Let $C=(\{\mathbf{a}_1,\mathbf{a}'_1\},\ldots,\{\mathbf{a}_n,\mathbf{a}'_n\})$ be such a fine mixed cells in S_ω. Then, since
$$\dim(C)=\dim(\mathrm{conv}\,\{\mathbf{a}_1,\mathbf{a}'_1\}+\cdots+\mathrm{conv}\,\{\mathbf{a}_n,\mathbf{a}'_n\})=n,$$

it follows that the set of vectors

$$\{\mathbf{a}'_1 - \mathbf{a}_1, \ldots, \mathbf{a}'_n - \mathbf{a}_n\}$$

is linearly independent. Hence,

$$\begin{aligned} \mathrm{Vol}\,(C) &= \mathrm{Vol}\,(\mathrm{conv}\,\{\mathbf{a}_1, \mathbf{a}'_1\} + \cdots + \mathrm{conv}\,\{\mathbf{a}_n, \mathbf{a}'_n\}) \\ &= \left| \det \begin{pmatrix} \mathbf{a}'_1 - \mathbf{a}_1 \\ \vdots \\ \mathbf{a}'_n - \mathbf{a}_n \end{pmatrix} \right|. \end{aligned} \tag{11}$$

Therefore, when all the fine mixed cells in S_ω are obtained by Algorithm 4 in the last section, the sum of the volumes of those cells computed in (11) gives the mixed volume of $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$.

Our algorithm has been successfully implemented in Fortran 77 and tested on a large variety of polynomial systems, and the numerical results we obtained clearly indicate our method has made a great progress in promoting the speed of computation of the mixed volume. Here, we elect to present our results on the widely considered benchmark system, the cyclic-n root problem [3]: $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where

$$\begin{aligned} &p_1(\mathbf{x}) = x_1 + x_2 + \cdots + x_n, \\ &p_2(\mathbf{x}) = x_1x_2 + x_2x_3 + \cdots + x_{n-1}x_n, \\ &\cdots\cdots \\ &p_n(\mathbf{x}) = x_1x_2\cdots x_n - 1. \end{aligned}$$

System	Mixed Volume	CPU time
Cyclic-8	2560	20s
Cyclic-9	11016	4m 10s
Cyclic-10	35940	46m 25s
Cyclic-11	184756	11h 14m 55s

The computations were carried out on a SPARCsever-1000 (50 MHz, 256M RAM). As a reference, we also list some results of CPU time computed on the same machine by the publicly available codes MVLP [4, 5] and PHC [16]:

System	MVLP	PHC
Cyclic-8	4m 35s	6m 57s
Cyclic-9	48m 44s	1h 37m 43s
Cyclic-10	6h 5m 34s	21h 23m 26s

Appendix

Proposition 1. *For any two vertices α' and α'' of the polyhedron $\mathcal{H}$ defined by the half spaces in $\mathbb{R}^n$:*

$$\langle \mathbf{a}_i, \alpha \rangle \le b_i, \quad i = 1, \ldots, m,$$

there exists a chain of vertices $\alpha^{(1)}, \ldots, \alpha^{(k)}$ of $\mathcal{H}$ with $\alpha^{(1)} = \alpha'$ and $\alpha^{(k)} = \alpha''$ and each line segment $\operatorname{conv}(\alpha^{(i)}, \alpha^{(i+1)})$, *$i = 1, \ldots, k-1$, is an edge of $\mathcal{H}$.*

Proof. Let $\alpha^{(1)} = \alpha'$. Without loss of generality, we assume both $\alpha^{(1)}$ and α'' are nondegenerate and

$$\operatorname{Act}(\alpha^{(1)}) \cap \operatorname{Act}(\alpha'') = \{\mathbf{a}_1, \ldots, \mathbf{a}_j\}, \quad 0 \le j < n.$$

We claim that

Lemma 4. *There exist vertices $\alpha^{(2)}, \ldots, \alpha^{(s)}$ of $\mathcal{H}$ such that for each $i = 1, \ldots, s-1$,* $\operatorname{conv}(\alpha^{(i)}, \alpha^{(i+1)})$ *is an edge of $\mathcal{H}$ and $\alpha^{(s)}$ and α'' share $j+1$ common active constraints.*

By applying this lemma repeatedly, the assertion of the proposition is obvious.

Proof of the Lemma. Let $\operatorname{Act}(\alpha^{(1)}) = \{\mathbf{a}_1, \ldots, \mathbf{a}_j, \mathbf{a}'_{j+1}, \ldots, \mathbf{a}'_n\}$ and $\operatorname{Act}(\alpha'') = \{\mathbf{a}_1, \ldots, \mathbf{a}_j, \mathbf{a}''_{j+1}, \ldots, \mathbf{a}''_n\}$ with $D^T = [\mathbf{a}_1, \ldots, \mathbf{a}_j, \mathbf{a}'_{j+1}, \ldots, \mathbf{a}'_n]$ and $D^{-1} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$. Let $\mathcal{P}$ be the polyhedron defined by

$$\langle \mathbf{a}_i, \alpha \rangle = b_i \quad \text{for } 1 \le i \le j$$

and

$$\langle \mathbf{a}''_i, \alpha \rangle \le b''_i \quad \text{for } j+1 \le i \le n.$$

Let $\mathcal{H}_1 = \mathcal{P} \cap \mathcal{H}$. Clearly, $\alpha^{(1)}$ is a relative interior point of $\mathcal{P}$. Moreover, a vertex of $\mathcal{H}_1$ is automatically a vertex of $\mathcal{H}$. Now, since both $\alpha^{(1)}$ and α'' are vertices of $\mathcal{H}_1$, there must exist $j+1 \leq l \leq n$ such that

$$\langle \mathbf{a}_i, \mathbf{c}_l \rangle < 0 \quad \text{for certain} \quad j < i \leq m;$$

otherwise, $\alpha^{(1)}$ would be the only vertex of $\mathcal{H}_1$. This means that the active constraint $\mathbf{a}'_l$ of $\alpha^{(1)}$ may be replaced by a constraint $\mathbf{a}_{i_0}$ with $i_0 > j$ to reach a new vertex $\bar{\alpha}$ of $\mathcal{H}_1$ with $\text{conv}\,(\alpha^{(1)}, \bar{\alpha})$ being an edge of $\mathcal{H}$ as we described in Section 2. The vertex $\bar{\alpha}$ may share $j+1$ common active constraints with α'', namely, $\mathbf{a}_{i_0} = \mathbf{a}''_t$ for certain $j+1 \leq t \leq n$. In that case, we let $\alpha^{(2)} = \bar{\alpha}$ and the conclusion of the lemma is reached with $s = 2$. Otherwise, let

$$\mathcal{B} = \{\bar{\alpha} \mid \begin{array}{l} \bar{\alpha} \text{ vertex of } \mathcal{H}_1 \text{ and } \bar{\alpha} \text{ and } \alpha^{(1)} \text{ have } n-1 \text{ common} \\ \text{active constraints, i.e., } \text{conv}\,(\bar{\alpha}, \alpha^{(1)}) \text{ is an edge of } \mathcal{H}\}. \end{array}$$

This set represents all the vertices of $\mathcal{H}_1$ that can be reached by applying one step of *pivoting* on $\alpha^{(1)}$ as described in Section 2. Now, let $\alpha^{(2)}$ be the vertex of $\mathcal{H}_1$ in $\mathcal{B}$ for which

$$|\alpha^{(2)} - \alpha''| = \min_{\bar{\alpha} \in \mathcal{B}} |\bar{\alpha} - \alpha''|.$$

Obviously, $\alpha^{(2)}$ is a relative interior point of $\mathcal{P}$. We may repeat the same argument on $\alpha^{(2)}$ and so on. This will eventually lead to a sequence of vertices of $\mathcal{H}_1$,

$$\alpha^{(2)}, \ldots, \alpha^{(s)},$$

where $\text{conv}\,(\alpha^{(i)}, \alpha^{(i+1)})$ are edges of $\mathcal{H}$ for all $1 \leq i \leq s-1$ and $\alpha^{(s)}$ reaches the relative boundary of $\mathcal{P}$. Namely, $\alpha^{(s)}$ and α'' share $j+1$ common active constraints.

■

References

1. D. N. Bernshtein, The number of roots of a system of equations, *Funct. Anal. Appl.* **9** (1975), 183-185.
2. M. J. Best and K. Ritter, *Linear Programming*: *Active Set Analysis and Computer Programs*, Prentice-Hall, New Jersey, 1985.
3. G. Björck and R. Fröberg, A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots, *J. Symbolic Comput.* **12** (1991), 329-336.

4. I. Z. Emiris and J. F. Canny, A practical method for the sparse resultant, in: *Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation*, ACM, 1993, pp. 183-192.

5. I. Z. Emiris and J. F. Canny, Efficient incremental algorithms for the sparse resultant and the mixed volume, *J. Symbolic Comput.* **20** (1995), 117-149.

6. T. Gao, T. Y. Li and X. Wang, Finding all isolated zeros of polynomial systems in $\mathbb{C}^n$ via stable mixed volume, *J. Symbolic Comput.* **28** (1999), 187-211.

7. B. Huber and B. Sturmfels, A polyhedral method for solving sparse polynomial systems, *Math. Comp.* **64** (1995), 1541-1555.

8. B. Huber and B. Sturmfels, Bernshtein's theorem in affine space, *Discrete Comput. Geom.* **17** (1997), 137-141.

9. T. Y. Li, Numerical solution of multivariate polynomial systems by homotopy continuation methods, in: *Acta Numerica*, Cambridge Univ. Press, Cambridge, 1997, pp. 399-436.

10. T. Y. Li, Solving polynomial systems by polyhedral homotopies, *Taiwanese J. Math.* **3** (1999), 251-279.

11. T. Y. Li and X. Wang, The BKK root count in $\mathbb{C}^n$, *Math. Comp.* **65** (1997), 1477-1484.

12. J. M. Rojas, A convex geometric approach to counting the roots of a polynomial system, *Theoret. Comput. Sci.* **133** (1994), 105-140.

13. J. M. Rojas, Toric intersection theory for affine root counting, *J. Pure Appl. Algebra* **136** (1999), 67-100.

14. J. M. Rojas and X. Wang, Counting affine roots of polynomial systems via pointed Newton polytopes, *J. Complexity* **12** (1996), 116-133.

15. J. Verschelde, *Homotopy Continuation Methods for Solving Polynomial Systems*, Ph.D. thesis, Department of Computer Science, Katholieke Universiteit Leuven (Leuven, Belgium), 1996.

16. J. Verschelde, HCPACK: A general-purpose solver for polynomial systems by homotopy continuation, *ACM Trans. Math. Software* **25** (1999), 251-276.

Tangan Gao
Department of Mathematics, California State University
Long Beach, CA 90840, U.S.A.

T. Y. Li
Department of Mathematics, Michigan State University
East Lansing, MI 48824, U.S.A.