Mixed Volume Computation, A Revisit

Tsung-Lin Lee*, Tien-Yien Li†

October 31, 2007

Abstract

The superiority of the *dynamic enumeration* of all mixed cells suggested by T. Mizutani et al. for the mixed volume computation over the existing methods was reported in [15]. In this article, we embed the idea of dynamic enumeration into our original algorithm for the mixed volume computation presented in [7], and a new algorithm is developed by employing both primal and dual simplex algorithms. Illustrated by the numerical results, our new algorithm improves the speed of the code in [15] by a substantial margin.

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

$$p_j(\mathbf{x}) = \sum_{\mathbf{a} \in \mathcal{S}_j} c_{j,\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \qquad j = 1, \ldots, n,$$

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

Let $\mathcal{Q}_j = conv(\mathcal{S}_j)$ for $j = 1, \cdots, n$. For positive numbers $\lambda_1, \cdots, \lambda_n$, the n-dimensional volume of the Minkovski sum

$$\lambda_1 \mathcal{Q}_1 + \cdots + \lambda_n \mathcal{Q}_n \equiv \{\lambda_1 \mathbf{q}_1 + \cdots + \lambda_n \mathbf{q}_n \mid \mathbf{q}_j \in \mathcal{Q}_j \ , \ j = 1, \cdots, n\}$$

is a homogeneous polynomial of degree n in the variables $\lambda_1, \cdots, \lambda_n$. The coefficient of $\lambda_1 \times \cdots \times \lambda_n$ in this polynomial is defined to be the *mixed volume* of $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$, denoted by $\mathcal{M}(\mathcal{S})$. In most occasions, we also call $\mathcal{M}(\mathcal{S})$ the mixed volume of $P(\mathbf{x})$.

By Bernshtein's theory [1], the mixed volume $\mathcal{M}(\mathcal{S})$ of $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$ of the polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ provides an upper bound for the number of isolated zeros

*Department of Mathematics, Michigan State University, East Lansing, MI 48824, email: `leetsung@msu.edu`.

†Department of Mathematics, Michigan State University, East Lansing, MI 48824, email: `li@math.msu.edu`.
Research supported in part by NSF under Grant DMS-0411165.

in $(\mathbb{C}^*)^n$, counting multiplicities. And this bound can be reached if the coefficients of $P(\mathbf{x})$ are *generic*, or the system is *in general position*. This root count in $(\mathbb{C}^*)^n$ has been extended to root count in $\mathbb{C}^n$ [14, 18]. They are, in general, much sharper than the classical Bézout number and its variants for sparse polynomial systems.

Based on this combinatorial root count, the *polyhedral homotopies* are established [9] to approximate all the isolated zeros of $P(\mathbf{x})$ by the homotopy continuation method, yielding a drastic improvement over the classical linear homotopies for sparse polynomial systems. 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 $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$ during the mixed volume computation plays a crucially important role [10, 11, 12]: The mixed volume determines the number of solution paths needed to be traced and the fine mixed cells provide starting points of the solution paths. Calculating the fine mixed cells (and thus the mixed volume) of the support $\mathcal{S}$ consumes large part of the computation and therefore dictates the efficiency of the method as well as the scope of its applications. In 2005, a software package, **MixedVol** [8], produced by T. Gao, T.Y. Li and M. Wu emerged which led the existing codes for the mixed volume computation by a great margin. However, soon after **MixedVol** was published, T. Mizutani, A. Takeda and M. Kojima [15] developed a more advanced algorithm which overshadowed **MixedVol** in speed by a big amount. A major ingredient for the efficiency of their algorithm is the novel idea of *dynamic enumerations* of mixed cells which helps to branch the parent node in the enumeration tree into its child nodes where the size of feasible child nodes is as small as possible.

When locating mixed cells, one must deal with a large scale of linear programming (LP) problems by the simplex method. Different from the primal simplex method used in **MixedVol** (as well as the previous works [6, 7, 13] which led to the development of **MixedVol**), the new algorithm **DEMiCs-0.95** [15] by T. Mizutani et al. adopted the dual simplex method to the LP problems. It was noted in their article: "In terms of the size of problems, the dual problems are superior to the primal ones $\cdots$" and "At least, the application of the dual simplex method is popular in the field of optimization to effectively deal with such a situation." Nonetheless, we believe the primal simplex method for this particular set of LP problems still has advantages of its own. In particular, the involvement of the level sets of the linear functionals, or the hyperplanes, in the primal simplex method as in [6, 7, 8, 13] can help to utilize the important informations that were generated in the process of pivotings more effectively. On the other hand, we found that the new idea of *dynamic enumerations* of mixed cells can be embedded in the algorithm in **MixedVol** with the spirit of the dual simplex method. Thus, in this article, we propose a new algorithm for finding mixed cells (with mixed volume as a by-product) where the primal simplex method and the dual simplex method are both in use. We will maintain the primal simplex method in the main course. But when the dynamic enumerations is used to determine a proper order of the supports for each individual mixed cell as suggested in [15] which involves checking the feasibilities of sets of inequalities, we always look into the boundedness of their dual problems. The details will be elaborated in Section 3.

Our method has been implemented successfully and the preliminary numerical results listed in Section 4 are quite impressive. Our algorithm leads the algorithm **DEMiCs-0.95** by T. Mizutani et al. in speed on all the benchmark systems provided in [15] and the speedups range from 1.3 to 4.8.

2 The main course

Almost all of the existing codes for the mixed volume computation calculate mixed volumes via calculating the mixed cells:

For a generic lifting $\omega = (\omega_1, \cdots, \omega_n)$ on $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$ with $\omega_j : \mathcal{S}_j \to \mathbb{R}$ for $j = 1, \ldots, n$, write

$$\widehat{\mathcal{S}}_j = \{\widehat{\mathbf{a}} = (\mathbf{a}, \omega_j(\mathbf{a})) \,|\, \mathbf{a} \in \mathcal{S}_j\}.$$

A collection of pairs $\{\mathbf{a}_1, \mathbf{a}'_1\} \subset \mathcal{S}_1, \cdots, \{\mathbf{a}_n, \mathbf{a}'_n\} \subset \mathcal{S}_n$ is called a *mixed cell* if there exists $\widehat{\alpha} = (\alpha, 1) \in \mathbb{R}^{n+1}$ with $\alpha \in \mathbb{R}^n$ such that

$$\langle \widehat{\mathbf{a}}_j, \widehat{\alpha} \rangle = \left\langle \widehat{\mathbf{a}}'_j, \widehat{\alpha} \right\rangle < \langle \widehat{\mathbf{a}}, \widehat{\alpha} \rangle \quad \text{for} \quad \mathbf{a} \in \mathcal{S}_j \backslash \left\{\mathbf{a}_j, \mathbf{a}'_j\right\}, \ \ j = 1, \ldots, n.$$

It is known that the mixed volume of $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$ equals the sum of volumes of all such mixed cells. That is,

$$\mathcal{M}(\mathcal{S}) = \sum_{\alpha} \left| \det \left(\mathbf{a}'_1 - \mathbf{a}_1, \cdots, \mathbf{a}'_n - \mathbf{a}_n \right) \right|.$$

On the other hand, those mixed cells play a critically important role in finding isolated zeros of polynomial systems numerically by the polyhedral homotopies [10, 11, 12].

To find all the mixed cells for a given generic lifting $\omega = (\omega_1, \cdots, \omega_n)$ on $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_n)$, we first construct the "Relation Table" $T(i, j)$ for $1 \le i \le j$ which display the relationships between elements of $\widehat{\mathcal{S}}_i$ and $\widehat{\mathcal{S}}_j$ in the following sense:

Given elements $\widehat{\mathbf{a}}_l^{(i)} \in \widehat{\mathcal{S}}_i$ and $\widehat{\mathbf{a}}_m^{(j)} \in \widehat{\mathcal{S}}_j$ where $l \neq m$ when $i = j$, does there exist an $\widehat{\alpha} = (\alpha, 1) \in \mathbb{R}^{n+1}$ such that

$$\begin{aligned} &\left\langle \widehat{\mathbf{a}}_l^{(i)}, \widehat{\alpha} \right\rangle \le \left\langle \widehat{\mathbf{a}}^{(i)}, \widehat{\alpha} \right\rangle \quad \text{for all } \ \widehat{\mathbf{a}}^{(i)} \in \widehat{\mathcal{S}}_i \\ \text{and} \quad &\left\langle \widehat{\mathbf{a}}_m^{(j)}, \widehat{\alpha} \right\rangle \le \left\langle \widehat{\mathbf{a}}^{(j)}, \widehat{\alpha} \right\rangle \quad \text{for all } \ \widehat{\mathbf{a}}^{(j)} \in \widehat{\mathcal{S}}_j \ ? \end{aligned} \tag{1}$$

Denote the entry on Table $T(i, j)$ at the intersection of the row containing $\widehat{\mathbf{a}}_l^{(i)}$ and the column containing $\widehat{\mathbf{a}}_m^{(j)}$ by $\left[\widehat{\mathbf{a}}_l^{(i)}, \widehat{\mathbf{a}}_m^{(j)}\right]$ and set $\left[\widehat{\mathbf{a}}_l^{(i)}, \widehat{\mathbf{a}}_m^{(j)}\right] = 1$ when the answer of Problem (1) is positive and $\left[\widehat{\mathbf{a}}_l^{(i)}, \widehat{\mathbf{a}}_m^{(j)}\right] = 0$ otherwise. An efficient algorithm to construct those tables was presented in [7].

To employ the "dynamic enumeration" suggested by [15], for k distinct integers $\{i_1, \cdots, i_k\} \subset \{1, \ldots, n\}$, we call

$$F_k := \left(\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}, \cdots, \left\{\mathbf{a}_{i_k}, \mathbf{a}'_{i_k}\right\} \right) \quad \text{where } \left\{\mathbf{a}_{i_j}, \mathbf{a}'_{i_j}\right\} \subset \mathcal{S}_{i_j} \ \text{ for } j = 1, \ldots, k \tag{2}$$

a *level-k subface* of $\widehat{\mathcal{S}} = \left(\widehat{\mathcal{S}}_1, \cdots, \widehat{\mathcal{S}}_n\right)$ (or simply "level-k subface" if no confusions exist) if there exists $\widehat{\alpha} = (\alpha, 1) \in \mathbb{R}^{n+1}$ such that for each $j = 1, \ldots, k$

$$\left\langle \widehat{\mathbf{a}}_{i_j}, \widehat{\alpha} \right\rangle = \left\langle \widehat{\mathbf{a}}'_{i_j}, \widehat{\alpha} \right\rangle \le \langle \widehat{\mathbf{a}}, \widehat{\alpha} \rangle \quad \text{for} \quad \mathbf{a} \in \mathcal{S}_{i_j} \backslash \left\{\mathbf{a}_{i_j}, \mathbf{a}'_{i_j}\right\}.$$

$\hat{\mathcal{S}}_i$

$\hat{\mathcal{S}}_i$	$\hat{\mathbf{a}}_2^{(i)}$	$\hat{\mathbf{a}}_3^{(i)}$	$\cdots$	$\hat{\mathbf{a}}_{\lvert s_i\rvert-1}^{(i)}$	$\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)}$
$\hat{\mathbf{a}}_1^{(i)}$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_2^{(i)}]$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_3^{(i)}]$	$\cdots$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_{\lvert s_i\rvert-1}^{(i)}]$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)}]$
$\hat{\mathbf{a}}_2^{(i)}$		$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_3^{(i)}]$	$\cdots$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_{\lvert s_i\rvert-1}^{(i)}]$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)}]$
			$\ddots$	$\vdots$	$\vdots$
$\hat{\mathbf{a}}_{\lvert s_i\rvert-1}^{(i)}$					$[\hat{\mathbf{a}}_{\lvert s_i\rvert-1}^{(i)},\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)}]$

Table T(i,i)

$\hat{\mathcal{S}}_j$

$\hat{\mathcal{S}}_i$	$\hat{\mathbf{a}}_1^{(j)}$	$\hat{\mathbf{a}}_2^{(j)}$	$\hat{\mathbf{a}}_3^{(j)}$	$\cdots$	$\hat{\mathbf{a}}_{\lvert s_j\rvert}^{(j)}$
$\hat{\mathbf{a}}_1^{(i)}$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_1^{(j)}]$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_2^{(j)}]$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_3^{(j)}]$	$\cdots$	$[\hat{\mathbf{a}}_1^{(i)},\hat{\mathbf{a}}_{\lvert s_j\rvert}^{(j)}]$
$\hat{\mathbf{a}}_2^{(i)}$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_1^{(j)}]$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_2^{(j)}]$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_3^{(j)}]$	$\cdots$	$[\hat{\mathbf{a}}_2^{(i)},\hat{\mathbf{a}}_{\lvert s_j\rvert}^{(j)}]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)}$	$[\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)},\hat{\mathbf{a}}_1^{(j)}]$	$[\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)},\hat{\mathbf{a}}_2^{(j)}]$	$[\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)},\hat{\mathbf{a}}_3^{(j)}]$	$\cdots$	$[\hat{\mathbf{a}}_{\lvert s_i\rvert}^{(i)},\hat{\mathbf{a}}_{\lvert s_j\rvert}^{(j)}]$

Table T(i,j)

If a level-k subface $F_k = \left(\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\},\cdots,\left\{\mathbf{a}_{i_k},\mathbf{a}'_{i_k}\right\}\right)$ is supplemented with $\left\{\mathbf{a}_{i_{k+1}},\mathbf{a}'_{i_{k+1}}\right\}\subset\mathcal{S}_{i_{k+1}}$ for certain $i_{k+1}\in\{1,2,\cdots,n\}\setminus\{i_1,\cdots,i_k\}$ so that $F_{k+1}:=F_k\cup\left\{\mathbf{a}_{i_{k+1}},\mathbf{a}'_{i_{k+1}}\right\}$ becomes a level-$(k+1)$ subface, we call F_{k+1} an extension of F_k, and we say F_k is extendable in such situations.

For finding mixed cells, we pick an appropriate $\widehat{\mathcal{S}}_{i_1}$ with $i_1\in\{1,\ldots,n\}$ as our point of departure. From Table $T(i_1,i_1)$, those pairs $\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}\subset\mathcal{S}_{i_1}$ with $\left[\widehat{\mathbf{a}}_{i_1},\widehat{\mathbf{a}}'_{i_1}\right]=1$ are the only possible level-1 subsurfaces. For a fixed pair $\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}$ among them, we search among $\{\mathcal{S}_j : j\in\{1,\cdots,n\}\setminus\{i_1\}\}$ for the support $\mathcal{S}_{i_2}$ having minimal amount of suitable points where only pairs among those points can possibly extend the level-1 subface $\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}$. The main strategy for finding such a support suggested in [15] is the removal of those points, as many as possible, in each support $\mathcal{S}_j$ where $j\neq i_1$, which have no chances to be part of pairs that can extend $\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}$ and select the support with minimal remaining points as $\mathcal{S}_{i_2}$. We will give the details of this procedure in the next section.

Suppose the selected $\mathcal{S}_{i_2}$ contains the remaining points $\mathbf{b}_1,\cdots,\mathbf{b}_l$. We follow by finding all the pairs among them that can extend $\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}$ as in [7], and the procedure is outlined below:

Let $M := \left\{\mathbf{a}\mid\mathbf{a}\in\mathcal{S}_{i_1}\setminus\left\{\mathbf{a}_{i_1},\mathbf{a}'_{i_1}\right\}\text{ and }[\widehat{\mathbf{a}},\widehat{\mathbf{a}}_{i_1}]=\left[\widehat{\mathbf{a}},\widehat{\mathbf{a}}'_{i_1}\right]=1\text{ in Table }T\left(i_1,i_1\right)\right\}$. For each $i=1,\ldots,l$, consider the One-Point test on $\mathbf{b}_i$:

$$
\begin{array}{rcl}
\text{Minimize} & & \left\langle\widehat{\mathbf{b}}_i,\widehat{\alpha}\right\rangle-\alpha_0 \\
\langle\widehat{\mathbf{a}}_{i_1},\widehat{\alpha}\rangle=\left\langle\widehat{\mathbf{a}}'_{i_1},\widehat{\alpha}\right\rangle & \leq & \langle\widehat{\mathbf{a}},\widehat{\alpha}\rangle \quad \forall\ \mathbf{a}\in M \\
\alpha_0 & \leq & \left\langle\widehat{\mathbf{b}}_k,\widehat{\alpha}\right\rangle \quad \forall\ k=1,\ldots,l
\end{array}
\tag{3}
$$

in the variables $\widehat{\alpha} = (\alpha, 1) \in \mathbb{R}^{n+1}$ and $\alpha_0 \in \mathbb{R}$. When the optimal value is zero, the point $\mathbf{b}_i$ will be retained for further considerations. Otherwise $\mathbf{b}_i$ would play no role in any pairs that can extend $\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}$, and therefore it can be removed.

An important feature here is that one never needs to solve all these Linear Programming (LP) problems when the simplex method is used, because the informations generated by the simplex pivoting already provide answers to some of the other LP problems. Moreover, as shown in [7], feasible points for the constraints of those LP problems are always available when those relation tables were established.

Let $\mathbf{b}_{j_1}, \cdots, \mathbf{b}_{j_\mu}$ be the remaining points in $\mathcal{S}_{i_2}$ after all the results of One-Point tests are attained. Next the Two-Point test will be applied on the pairs among these points; that is, for $l \neq m$ in $\{1, \cdots, \mu\}$, consider the LP problem

$$\begin{array}{rcl} \text{Minimize} & & \left\langle \widehat{\mathbf{b}}_{j_l} + \widehat{\mathbf{b}}_{j_m}, \widehat{\alpha} \right\rangle - 2\alpha_0 \\ \left\langle \widehat{\mathbf{a}}_{i_1}, \widehat{\alpha} \right\rangle = \left\langle \widehat{\mathbf{a}}'_{i_1}, \widehat{\alpha} \right\rangle & \leq & \left\langle \widehat{\mathbf{a}}, \widehat{\alpha} \right\rangle \quad \forall\, \mathbf{a} \in M \\ \alpha_0 & \leq & \left\langle \widehat{\mathbf{b}}_{j_k}, \widehat{\alpha} \right\rangle \quad \forall\, k = 1, \ldots, \mu. \end{array} \tag{4}$$

Clearly, only zero optimal value of this LP problem grants the permission to the pair $\{\mathbf{b}_{j_l}, \mathbf{b}_{j_m}\}$ for extending $\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}$ to a level-2 subface of $\widehat{\mathcal{S}} = \left(\widehat{\mathcal{S}}_1, \ldots, \widehat{\mathcal{S}}_n\right)$. Again, one never needs to solve these LP problems for *all* pairs, because most of the pairs $\{\mathbf{b}_{j_l}, \mathbf{b}_{j_m}\}$ having zero optimal value for the corresponding LP problem in (4) are revealed when constraints involving both $\widehat{\mathbf{b}}_{j_l}$ and $\widehat{\mathbf{b}}_{j_m}$ are active in certain pivoting stages during One-Point tests in (3) were performed or when Two-Point tests were applied to other pairs.

In summary, applying One-Point tests followed by Two-Point tests on $\{\mathbf{b}_1, \cdots, \mathbf{b}_l\} \subset \mathcal{S}_{i_2}$ will result in a set of pairs in $\mathcal{S}_{i_2}$ in which each pair can extend $\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\} \subset \mathcal{S}_{i_1}$ to a level-2 subface of $\widehat{\mathcal{S}} = \left(\widehat{\mathcal{S}}_1, \cdots, \widehat{\mathcal{S}}_n\right)$. Of course, if this set is empty, i.e., there exists no pairs in $\mathcal{S}_{i_2}$ that can extend $\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}$, then we must stop here and focus our attention on extending other level-1 subfaces in $\mathcal{S}_{i_1}$.

Extending level-2 subface $(\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}, \left\{\mathbf{a}_{i_2}, \mathbf{a}'_{i_2}\right\})$ with certain $\left\{\mathbf{a}_{i_2}, \mathbf{a}'_{i_2}\right\} \subset \mathcal{S}_{i_2}$ may follow the similar procedure as described above and this process can be continued. Finally, mixed cells will be induced by pairs $\left\{\mathbf{a}_{i_n}, \mathbf{a}'_{i_n}\right\}$ in $\mathcal{S}_{i_n}$ when we show $(\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}, \cdots, \left\{\mathbf{a}_{i_{n-1}}, \mathbf{a}'_{i_{n-1}}\right\})$ with $\left\{\mathbf{a}_{i_j}, \mathbf{a}'_{i_j}\right\} \subset \mathcal{S}_{i_j}$ for $j = 1, \ldots, n-1$ is extendable.

3 Removing non-essential points in the supports

For integer k with $1 \leq k < n-1$, let $F_k = \left(\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}, \cdots, \left\{\mathbf{a}_{i_k}, \mathbf{a}'_{i_k}\right\}\right)$ be a level-k subface of $\widehat{\mathcal{S}} = \left(\widehat{\mathcal{S}}_1, \cdots, \widehat{\mathcal{S}}_n\right)$ where $\{\mathbf{a}_l, \mathbf{a}'_l\} \subset \mathcal{S}_l$ for $l \in Q := \{i_1, \cdots, i_k\} \subset \{1, \cdots, n\}$. For a fixed $j \in \{1, \cdots, n\} \backslash Q$, let $V := \{\mathbf{b}_1, \ldots, \mathbf{b}_r\} \subset \mathcal{S}_j$ where for each $\mu = 1, \ldots, r$,

$$\left[\widehat{\mathbf{b}}_\mu, \widehat{\mathbf{a}}_l\right] = \left[\widehat{\mathbf{b}}_\mu, \widehat{\mathbf{a}}'_l\right] = 1 \quad \forall\, l \in Q$$

in the relation table $T(l, j)$. We wish to remove, as many as possible, those points in V which would be absent in any pairs in V that can extend the level-k subface F_k.

For a particular point $\mathbf{b}_v \in V$, consider the set of constraints:

$$\begin{array}{rcll} \langle \widehat{\mathbf{a}}_l, \widehat{\alpha} \rangle & = & \left\langle \widehat{\mathbf{a}}'_l, \widehat{\alpha} \right\rangle & \forall \;\; l \in Q \\ & \leq & \langle \widehat{\mathbf{a}}, \; \widehat{\alpha} \rangle & \forall \; \mathbf{a} \in \mathcal{S}_l \backslash \left\{ \mathbf{a}_l, \mathbf{a}'_l \right\} \\ \left\langle \widehat{\mathbf{b}}_v, \widehat{\alpha} \right\rangle & \leq & \left\langle \widehat{\mathbf{b}}, \widehat{\alpha} \right\rangle & \forall \;\; \mathbf{b} \in V \backslash \left\{ \mathbf{b}_v \right\}. \end{array} \tag{5}$$

Apparently, when the set of equalities and inequalities in (5) is infeasible, then there is no $\mathbf{b}_\mu$ in V such that $\{\mathbf{b}_v, \mathbf{b}_\mu\}$ can extend F_k to become a level-$(k+1)$ subface, and therefore $\mathbf{b}_v$ can be safely removed from V.

More explicitly, equalities and inequalities in (5) yield,

$$\begin{array}{rcll} \left\langle \mathbf{a}_l - \mathbf{a}'_l \;, \alpha \right\rangle & = & \omega_l(\mathbf{a}'_l) - \omega_l(\mathbf{a}_l) & \forall \; l \in Q \\ \langle \mathbf{a}_l - \mathbf{a} \;\;, \alpha \rangle & \leq & \omega_l(\mathbf{a}) - \omega_l(\mathbf{a}_l) & \forall \; \mathbf{a} \in \mathcal{S}_l \backslash \left\{ \mathbf{a}_l, \mathbf{a}'_l \right\} \\ \langle \mathbf{b}_v - \mathbf{b} \;, \alpha \rangle & \leq & \omega_j(\mathbf{b}) - \omega_j(\mathbf{b}_v) & \forall \; \mathbf{b} \in V \backslash \left\{ \mathbf{b}_v \right\}. \end{array} \tag{I}$$

As in [15], the feasibility check of (I) can be formulated via an LP problem,

$$\text{(P)} \qquad \begin{array}{rl} \text{Max} & \langle \mathbf{r}, \alpha \rangle \\ \text{subject to} & \text{(I)} \end{array}$$

where $\mathbf{r} \in \mathbb{R}^n$ is some fixed vector, along with its dual problem

$$\text{(D)} \qquad \begin{array}{rcll} \text{Min} \;\; \Phi(\mathbf{x}) & & & \\ \text{subject to} \; \Psi(\mathbf{x}) & = & \mathbf{r} & \\ x_{\mathbf{a}} & \geq & 0 & \left(\; \mathbf{a} \in \mathcal{S}_l \backslash \left\{ \mathbf{a}_l, \mathbf{a}'_l \right\}, \forall \; l \in Q \; \right) \\ x_{\mathbf{b}} & \geq & 0 & \left(\; \mathbf{b} \in V \backslash \left\{ \mathbf{b}_v \right\} \; \right) \\ -\infty < x_{\mathbf{a}'_l} & < & \infty & \left(\; l \in Q \; \right) \end{array}$$

in the variables

$$\mathbf{x} = \left(x_{\mathbf{a}} : \mathbf{a} \in \mathcal{S}_l \backslash \left\{ \mathbf{a}_l \right\}, \; l \in Q; \;\; x_{\mathbf{b}} : \mathbf{b} \in V \backslash \left\{ \mathbf{b}_v \right\} \right)$$

where

$$\Phi(\mathbf{x}) = \sum_{l \in Q} \sum_{\mathbf{a} \in \mathcal{S}_l \backslash \{\mathbf{a}_l\}} \left(\omega_l(\mathbf{a}) - \omega_l(\mathbf{a}_l) \right) x_{\mathbf{a}} + \sum_{\mathbf{b} \in V \backslash \{\mathbf{b}_v\}} \left(\omega_j(\mathbf{b}) - \omega_j(\mathbf{b}_v) \right) x_{\mathbf{b}}$$

and

$$\Psi(\mathbf{x}) = \sum_{l \in Q} \sum_{\mathbf{a} \in \mathcal{S}_l \backslash \{\mathbf{a}_l\}} \left(\mathbf{a}_l - \mathbf{a} \right) x_{\mathbf{a}} + \sum_{\mathbf{b} \in V \backslash \{\mathbf{b}_v\}} \left(\mathbf{b}_v - \mathbf{b} \right) x_{\mathbf{b}}.$$

Any vector $\mathbf{r} \in \mathbb{R}^n$ may be chosen for the cost vector in (P), therefore one may choose $\mathbf{r}$ so that (D) becomes feasible. From the duality theorem, (I) is infeasible if and only if (D) is unbounded.

To determine the boundedness of the LP problem in (D), we first recall that when $F_k = \left(\left\{ \mathbf{a}_{i_1}, \mathbf{a}'_{i_1} \right\}, \cdots, \left\{ \mathbf{a}_{i_k}, \mathbf{a}'_{i_k} \right\} \right)$ with $\{\mathbf{a}_l, \mathbf{a}'_l\} \subset \mathcal{S}_l$ for all $l \in Q$ is declaired to be a level-k subface of $\widehat{\mathcal{S}} = \left(\widehat{\mathcal{S}}_1, \cdots, \widehat{\mathcal{S}}_n \right)$, the following set of constraints must be satisfied for certain $\alpha \in \mathbb{R}^n$,

$$\begin{array}{rcll} \left\langle \mathbf{a}_l - \mathbf{a}'_l, \alpha \right\rangle & = & \omega_l(\mathbf{a}'_l) - \omega_l(\mathbf{a}_l), & \forall \; l \in Q \\ \langle \mathbf{a}_l - \mathbf{a}, \alpha \rangle & \leq & \omega_l(\mathbf{a}) - \omega_l(\mathbf{a}_l), & \forall \; \mathbf{a} \in \mathcal{S}_l \backslash \left\{ \mathbf{a}_l, \mathbf{a}'_l \right\}. \end{array} \tag{6}$$

Resulting from implementing the One-Point test or the Two-Point test by solving LP problems in (3) or (4) by the simplex algorithm, there are exactly n equalities in (6), and the inverse for the matrix determined by this $n \times n$ linear equations is always available in the context. Let those equalities be

$$\begin{aligned} \left\langle \mathbf{a}_l - \mathbf{a}'_l, \alpha \right\rangle &= \omega_l(\mathbf{a}'_l) - \omega_l(\mathbf{a}_l), \qquad \forall\, l \in Q = \{i_1, \cdots, i_k\} \\ \left\langle \mathbf{a}_{j_{k+h}} - \widetilde{\mathbf{a}}_{k+h}, \alpha \right\rangle &= \omega_{j_{k+h}}(\widetilde{\mathbf{a}}_{k+h}) - \omega_{j_{k+h}}(\mathbf{a}_{j_{k+h}}) \end{aligned}$$

where for $h = 1, \ldots, n-k,\ \ j_{k+h} \in Q$ and $\widetilde{\mathbf{a}}_{k+h} \in \mathcal{S}_{j_{k+h}} \setminus \left\{\mathbf{a}_{j_{k+h}}, \mathbf{a}'_{j_{k+h}}\right\}$.

In matrix form, we have

$$D^\top \alpha = E \tag{7}$$

where

$$D = \left[\mathbf{a}_{i_1} - \mathbf{a}'_{i_1}, \cdots, \mathbf{a}_{i_k} - \mathbf{a}'_{i_k}, \mathbf{a}_{j_{k+1}} - \widetilde{\mathbf{a}}_{k+1}, \cdots, \mathbf{a}_{j_n} - \widetilde{\mathbf{a}}_n\right]$$

$$E = \begin{bmatrix} \omega_{i_1}(\mathbf{a}'_{i_1}) - \omega_{i_1}(\mathbf{a}_{i_1}) \\ \vdots \\ \omega_{i_k}(\mathbf{a}'_{i_k}) - \omega_{i_k}(\mathbf{a}_{i_k}) \\ \omega_{j_{k+1}}(\widetilde{\mathbf{a}}_{k+1}) - \omega_{j_{k+1}}(\mathbf{a}_{j_{k+1}}) \\ \vdots \\ \omega_{j_n}(\widetilde{\mathbf{a}}_n) - \omega_{j_n}(\mathbf{a}_{j_n}) \end{bmatrix},$$

and, as mentioned above, $\left(D^\top\right)^{-1}$ is available. Actually, the columns of matrix D provide a basis when we deal with the LP problem in (D). Under this basis, a criteria for the unboundedness of the LP problem is, if the cost

$$r_{\mathbf{b}} = \omega_j(\mathbf{b}) - \omega_j(\mathbf{b}_v) - E^\top D^{-1}(\mathbf{b}_v - \mathbf{b}) < 0$$

and the $\xi-$th entry of $D^{-1}(\mathbf{b}_v - \mathbf{b})$ is non-positive for all $\xi \geq k+1$, then this LP problem is unbounded. However, by (7),

$$\begin{aligned} r_{\mathbf{b}} &= \omega_j(\mathbf{b}) - \omega_j(\mathbf{b}_v) - \left(\alpha^\top D\right) D^{-1}(\mathbf{b}_v - \mathbf{b}) \\ &= \omega_j(\mathbf{b}) + \langle \mathbf{b}, \alpha\rangle - (\omega_j(\mathbf{b}_v) + \langle \mathbf{b}_v, \alpha\rangle) \\ &= \left\langle \widehat{\mathbf{b}}, \widehat{\alpha} \right\rangle - \left\langle \widehat{\mathbf{b}}_v, \widehat{\alpha} \right\rangle. \end{aligned}$$

So, only the sign of entries of $D^{-1}(\mathbf{b}_v - \mathbf{b})$ for those $\mathbf{b} \in V$ for which $\left\langle \widehat{\mathbf{b}}, \widehat{\alpha} \right\rangle < \left\langle \widehat{\mathbf{b}}_v, \widehat{\alpha} \right\rangle$ needed to be checked to determine the possible unboundedness of the LP problem in (D).

Summarizing the above, yields the following algorithm.

Algorithm : Removing non-essential points in the supports

Input: A level-k subface $F_k := \left(\left\{\mathbf{a}_{i_1}, \mathbf{a}'_{i_1}\right\}, \cdots, \left\{\mathbf{a}_{i_k}, \mathbf{a}'_{i_k}\right\}\right)$ along with α and D in (7).

Output: i_{k+1} for which the support $\mathcal{S}_{i_{k+1}}$ has fewest points where only pairs among which can possibly extend F_k.

for all $j \in \{1, \ldots, n\} \backslash Q$ where $Q := \{i_1, \cdots, i_k\}$ **do**

$M_j \longleftarrow \emptyset$

for all $\mathbf{b}_\mu \in \mathcal{S}_j$ **do**

if for all $l \in Q$, $\left[\widehat{\mathbf{b}}_\mu, \widehat{\mathbf{a}}_l\right] = \left[\widehat{\mathbf{b}}_\mu, \widehat{\mathbf{a}}'_l\right] = 1$ in the Relation Table $T(j, l)$, **then**

$M_j \longleftarrow M_j \cup \{\mathbf{b}_\mu\}$

end if

end for

if M_j doesn't contain at least 2 elements, **then** output "F_k is not extendible".

for all $\mathbf{b}_v \in M_j$ **do**

for all $\mathbf{b}_l \in M_j \backslash \{\mathbf{b}_v\}$ **do**

if $\left\langle \widehat{\mathbf{b}}_l, \widehat{\alpha} \right\rangle < \left\langle \widehat{\mathbf{b}}_v, \widehat{\alpha} \right\rangle$, **then**

for all $\xi \in \{k+1, \cdots, n\}$ **do**

if the $\xi-$th entry of $D^{-1}(\mathbf{b}_v - \mathbf{b})$ is positive, **then**

go to **(A)**.

end if

end for

$M_j \longleftarrow M_j \backslash \{\mathbf{b}_v\}$

end if

end for

(A)

end for

$N_j \longleftarrow$ the number of elements in M_j

if $N_j < 2$, **then** output "F_k is not extendible".

end for

return i_{k+1} where $N_{i_{k+1}} = \min\{N_j : j \in \{1, \cdots, n\} \backslash Q\}$

4 Numerical Results

Our algorithm has been successfully implemented in FORTRAN-90 and its Matlab interface version is available at http://www.msu.edu/∼leetsung/Software.htm.

To compare our new code **MixedVol-2.0** with existing codes **MixedVol** and **DEMiCs-0.95** for the mixed volume computation, we will mainly concentrate on the benchmark systems

system	size(n)	mixed volume	**MixedVol-2.0**	**MixedVol**	speed-up ratio
Cyclic-n	12	500,352	2.41m	10.8m	4.48
	13	2,704,156	20.9m	1.93hr	5.54
	14	8,795,976	2.72hr	17.1hr	6.29
	15	35,243,520	23.9hr	–	–
Noon-n	19	1,162,261,429	28.0m	78.3hr	168.
	20	3,486,784,361	1.32hr	–	–
Eco-n	18	65,536	32.7m	2.12hr	3.89
	19	131,072	2.19hr	14.5hr	6.62
	20	262,144	8.53hr	62.3hr	7.30
	21	524,288	28.1hr	–	–
Chandra-n	20	524,288	18.6m	78.2hr	252.
	21	1,048,576	46.8m	–	–
Katsura-n	13	8,192	7.10m	3.48hr	29.4
	14	16,384	31.5m	19.7hr	37.5
	15	32,768	2.58hr	108.hr	41.9
	16	65,536	15.8hr	–	–
Gaukwa-n	6	371,293	16.1s	1.11m	4.14
	7	11,390,625	6.79m	46.1m	6.79
	8	410,338,673	3.20hr	33.5hr	10.5

Table 1: 1.6GHz Itanium2 processor, 2G RAM

listed in [15], such as, Cyclic-n [2, 5], Noon-n [17], Economic-n [16], Chandra-n [4], Katsura-n [3], and Gaukwa-n [19]. All the computations were carried out on a 1.6GHz Itanium2 CPU with 2G RAM.

Table 1 compares our new code **MixedVol-2.0** with **MixedVol**. As expected, it illustrates a very high level of speed-ups.

The superiority of **DEMiCs-0.95** in speed had been demonstrated in [15]. In Table 2, we compare our code with **DEMiCs-0.95**. As indicated, our algorithm is uniformly faster than **DEMiCs-0.95** in cpu time on all the systems and the speedups range from 1.37 to 4.83.

system	size(n)	mixed volume	**MixedVol-2.0**	**DEMiCs-0.95**	speed-up ratio
Cyclic-n	12	500,352	2.41m	3.31m	1.37
	13	2,704,156	20.9m	29.5m	1.41
	14	8,795,976	2.72hr	4.06hr	1.49
	15	35,243,520	23.9hr	37.8hr	1.58
Noon-n	19	1,162,261,429	28.0m	70.6m	2.52
	20	3,486,784,361	1.32hr	2.69hr	2.04
	21	10,460,353,161	3.31hr	9.46hr	2.86
	22	31,381,059,565	7.12hr	25.8hr	3.62
	23	94,143,178,781	21.8hr	74.4hr	3.41
Eco-n	18	65,536	32.7m	52.3m	1.60
	19	131,072	2.19hr	3.31hr	1.51
	20	262,144	8.53hr	12.0hr	1.41
	21	524,288	28.1hr	40.2hr	1.43
Chandra-n	20	524,288	18.6m	76.3m	4.10
	21	1,048,576	46.8m	3.37hr	4.32
	22	2,097,152	2.36hr	8.63hr	3.66
	23	4,194,304	5.75hr	27.8hr	4.83
	24	8,388,608	18.5hr	75.2hr	4.06
Katsura-n	13	8,192	7.10m	11.0m	1.55
	14	16,384	31.5m	60.2m	1.91
	15	32,768	2.58hr	5.14hr	1.99
	16	65,536	15.8hr	23.2hr	1.47
Gaukwa-n	6	371,293	16.1s	33.2s	2.06
	7	11,390,625	6.79m	20.9m	3.08
	8	410,338,673	3.20hr	13.1hr	4.09

Table 2: 1.6GHz Itanium2 processor, 2G RAM

References

[1] D. N. Bernshtein (1975), "The number of roots of a system of equations", *Functional Analysis and Appl.* **9**(3), 183-185. Translated from *Funktsional. Anal. i Prilozhen.*, **9**(3), 1 - 4.

[2] G. Björk and R. Fröberg (1991), "A faster way to count the solutions of inhomogeneous systems of algebraic equations", *J. Symbolic Comput.* **12**(3), 329 - 336.

[3] W. Boege, R. Gebauer and H. Kredel (1986), "Some examples for solving systems of algebraic equations by calculating Groebner bases", *J. Symbolic Comput.* **2**, 83 - 98.

[4] S. Chandrasekhar (1960), "Radiative Transfer", Dover, NY.

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

[6] T. Gao and T. Y. Li (2000), "Mixed volume computation via linear programming", *Taiwan J. of Math.*, **4**, 599-619.

[7] T. Gao and T. Y. Li (2003), "Mixed volume computation for Semi-Mixed Systems", *Discrete Comput. Geom.*, **29**(2), 257-277.

[8] T. Gao, T. Y. Li and M. Wu (2005), "MixedVol: A Software Package for Mixed Volume Computation", *ACM Transactions on Math. Software*, **31**(4), 555 - 560.

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

[10] T. Y. Li (1997), "Numerical solution of multivariate polynomial systems by homotopy continuation methods", *ACTA Numerica*, 399-436.

[11] T. Y. Li (1999), "Solving polynomial systems by polyhedral homotopies", *Taiwan J. of Math.*, **3**, 251 - 279.

[12] T. Y. Li (2003), "Solving polynomial systems by the homotopy continuation method", *Handbook of numerical analysis*, Vol. XI, Edited by P. G. Ciarlet, North-Holland, Amsterdam.

[13] T. Y. Li and X. Li (2001), "Finding mixed cells in the mixed volume computation", *Foundation of Computational Mathematics*, **1**, 161-181.

[14] T. Y. Li and X. Wang (1997), "The BKK root count in $\mathbb{C}^n$", *Math. Comp*, **65**(216), 1477 - 1484.

[15] T. Mizutani, A. Takeda and M. Kojima (2007), "Dynamic Enumeration of All Mixed Cells", *Discrete Comput. Geom.*, **37**, 351 - 367.

[16] A. Morgan (1987), "Solving polynomial systems using continuation for engineering and scientific problems", Prentice-Hall, Englewood Cliffs, New Jersey.

[17] V. W. Noonburg (1989), "A neural network modeled by an adaptive Lotka-Volterra system", *SIAM J. Appl. Math.* **49**, 1779 - 1792.

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

[19] A. H. Stroud and D. Secrest (1966), "Gaussian Quadrature Formulas", Prentice-Hall, Englewood Cliffs, New Jersey.