This is to certify that the

dissertation entitled

SOLVING POLYNOMIAL SYSTEMS
IN C^n BY POLYHEDRAL HOMOTOPIES

presented by

Xing Li

has been accepted towards fulfillment
of the requirements for

Ph. D degree in Mathematics

Major professor

Date August 1,2000

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

SOLVING POLYNOMIAL SYSTEMS IN $\mathbb{C}^n$ BY POLYHEDRAL HOMOTOPIES

By

Xing Li

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

2000

ABSTRACT

SOLVING POLYNOMIAL SYSTEMS IN $\mathbb{C}^n$ BY POLYHEDRAL HOMOTOPIES

By

Xing Li

In the last two decades, the homotopy continuation method has been developed into a reliable and efficient numerical algorithm for solving all isolated zeros of polynomial systems. During the last few years, a major computational breakthrough has emerged in the area. Based on the Bernshtein theory on root count, the *polyhedral homotopy* is established to considerably reduce the number of homotopy paths that need to be traced to find all the isolated roots, making the method much more powerful.

The main goal of this dissertation is to present a strategy which uses homotopy continuation method efficiently to solve polynomial systems via mixed cell calculation.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Tien-Yien Li, my dissertation advisor, for his constant encouragement and kind guidance. His knowledge, insights and enthusiasm were invaluable.

I would also like to thank Professor William C. Brown, Professor Chichia Chiu, Professor Ronald Fintushel, Professor Michael Frazier, Professor Richard Hill, Professor Wei-Eihn Kuan and Professor Jay Kurtz for their time and advice.

TABLE OF CONTENTS

Introduction

Polynomial systems arise quite commonly in many fields of science and engineering, such as formula construction, geometric intersection, inverse kinematics, power flow with PQ-specified bases, computation of equilibrium states, etc., see [10]. Elimination theory-based methods, most notably the Buchberger algorithm [5] for constructing Gröbner bases, are the classical approach to solving multivariate polynomial systems, but their reliance on symbolic manipulation makes those methods somewhat unsuitable for all but small problems.

In 1977, Garcia and Zangwill [14] and Drexler [11] independently presented theorems suggesting that homotopy continuation could be used to find the full set of isolated zeros of a polynomial system numerically. During the last two decades this method has been developed into a reliable and efficient numerical algorithm for approximating all isolated zeros of polynomial systems. See [23] for a survey.

Let $P(\mathbf{x}) = 0$ be a system of n polynomial equations in n unknowns. Denoting $P = (p_1, ..., p_n)$, we want to find all isolated solutions of

$$\begin{aligned} p_1(x_1, ..., x_n) &= 0 \\ &\vdots \\ p_n(x_1, ..., x_n) &= 0, \end{aligned} \tag{1}$$

for $\mathbf{x} = (x_1, ..., x_n)$. The classical homotopy continuation method for solving (1) is to define a system that is easy to solve $Q(\mathbf{x}) = (q_1(\mathbf{x}), ..., q_n(\mathbf{x})) = 0$ and then follow the curves in the real variable t which make up the solution set of

$$0 = H(\mathbf{x}, t) = (1 - t)Q(\mathbf{x}) + tP(\mathbf{x}). \tag{2}$$

More precisely, if the system $Q(\mathbf{x})$, known as the *start* system, is chosen correctly, the following three properties hold:

- Property 1 (*Triviality*). The solutions of $Q(\mathbf{x}) = 0$ are known.
- Property 2 (*Smoothness*). The solution set of $H(\mathbf{x}, t) = 0$ for $0 \le t \le 1$ consists of a finite number of smooth paths, each parametrized by t in $[0, 1)$.
- Property 3 (*Accessibility*). Every isolated solution of $H(\mathbf{x}, 1) = P(\mathbf{x}) = 0$ can be reached by some path originating at $t = 0$. It follows that this path starts at a solution of $H(\mathbf{x}, 0) = Q(\mathbf{x}) = 0$.

When the three properties hold, the solution paths can be followed from the initial points (known because of Property 1) at $t = 0$ to all solutions of the original problem $P(\mathbf{x}) = 0$ at $t = 1$ using standard numerical techniques [1, 2]. A homotopy $H(\mathbf{x}, t) = 0$ with $H(\mathbf{x}, 0) = Q(\mathbf{x})$ and $H(\mathbf{x}, 1) = P(\mathbf{x})$, which may not be in the form of (2), is considered to be *successful* if it satisfies these three properties.

A typical choice [8, 22, 24, 30, 46, 47] of the system $Q(\mathbf{x}) = (q_1(\mathbf{x}), ..., q_n(\mathbf{x}))$ which satisfies Properties 1-3 is,

$$\begin{aligned} q_1(x_1, ..., x_n) &= a_1 x_1^{d_1} - b_1 \\ &\vdots \\ q_n(x_1, ..., x_n) &= a_n x_n^{d_n} - b_n, \end{aligned} \tag{3}$$

where $d_1, ..., d_n$ are the degrees of $p_1(\mathbf{x}), ..., p_n(\mathbf{x})$ respectively, and a_j, b_j are random complex numbers (and therefore nonzero with probability one). So in one sense, the original problem posed is solved. All solutions of $P(\mathbf{x}) = 0$ are found at the end of $d_1 \cdots d_n$ paths that make up the solution set of $H(\mathbf{x}, t) = 0, 0 \le t \le 1$.

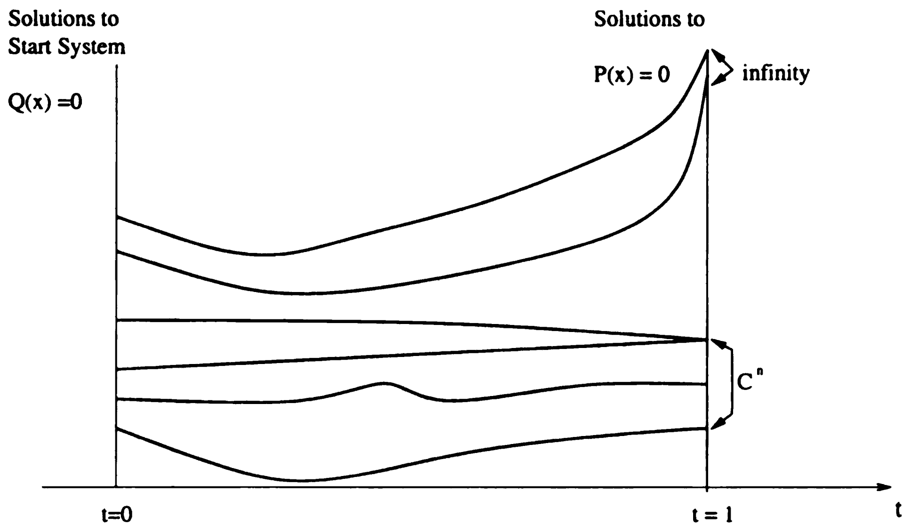

Figure 1: Solution curves of $H(x, t) = 0$

The reason the problem is not satisfactorily solved by the above considerations is the existence of *extraneous* paths. Although the above method produces $d = d_1 \cdots d_n$ paths since $Q(\mathbf{x}) = 0$ in (3) has d isolated nonsingular solutions, the system $P(\mathbf{x}) = 0$ may have fewer than d solutions. We call such a system *deficient.* In this case, some of the paths produced by the above method will be extraneous paths.

More precisely, even though Properties 1-3 imply that each solution of $P(\mathbf{x}) = 0$ will lie at the end of a solution path, it is also consistent with these properties that some of the paths may diverge to infinity as the parameter t approaches 1 (the smoothness property rules this out for $t \to t_0 < 1$). In other words, it is quite possible for $Q(\mathbf{x}) = 0$ to have more solutions than $P(\mathbf{x}) = 0$. In this case, some of the paths leading from roots of $Q(\mathbf{x}) = 0$ are extraneous, and diverge to infinity when $t \to 1$ (See Figure 1).

Empirically, we find that most systems arising in applications are deficient. A great majority of the systems have fewer than, and in some cases only a small fraction of, the *expected number* of solutions. For a typical example of this sort, let us look at the following Cassou-Nogues system

$$\begin{aligned}
p_1 &= 15b^4cd^2 + 6b^4c^3 + 21b^4c^2d - 144b^2c - 8b^2c^2e \\
&\quad -28b^2cde - 648b^2d + 36b^2d^2e + 9b^4d^3 - 120, \\
p_2 &= 30b^4c^3d - 32cde^2 - 720b^2cd - 24b^2c^3e - 432b^2c^2 + 576ce - 576de \\
&\quad +16b^2cd^2e + 16d^2e^2 + 16c^2e^2 + 9b^4c^4 + 39b^4c^2d^2 + 18b^4cd^3 \\
&\quad -432b^2d^2 + 24b^2d^3e - 16b^2c^2de - 240c + 5184, \qquad (4) \\
p_3 &= 216b^2cd - 162b^2d^2 - 81b^2c^2 + 1008ce - 1008de + 15b^2c^2de \\
&\quad -15b^2c^3e - 80cde^2 + 40d^2e^2 + 40c^2e^2 + 5184, \\
p_4 &= 4b^2cd - 3b^2d^2 - 4b^2c^2 + 22ce - 22de + 261.
\end{aligned}$$

Since $d_1 = 7, d_2 = 8, d_3 = 6$ and $d_4 = 4$ for this system, the system $Q(\mathbf{x})$ in (3) will produce $d_1 \times d_2 \times d_3 \times d_4 = 7 \times 8 \times 6 \times 4 = 1344$ paths for the homotopy in (2). However, the system (4) has only 16 isolated zeros. Consequently, a major fraction of the paths are extraneous. Sending out 1344 paths in search of 16 solutions is a highly wasteful computation.

The choice of $Q(\mathbf{x})$ in (3) to solve the system $P(\mathbf{x}) = 0$ requires an amount of computational effort proportional to $d_1 \cdots d_n$, known as the *Bézout number*, which bounds the number of isolated zeros, counting multiplicities, of $P(\mathbf{x})$ in $\mathbb{C}^n$ [39]. We wish to derive methods for solving deficient systems for which the computational effort is instead proportional to the actual number of solutions.

In the last few years, a major computational breakthrough has emerged in the area. The new idea takes a great advantage of the Bernshtein theory [4] which provides a much tighter bound, compared to the Bézout bound, for the number of isolated zeros of $P(x)$ in the algebraic tori $(\mathbb{C}^*)^n$, where $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$. The so called *polyhedral homotopy* [18] is then established for the new method and the homotopy paths so produced is much fewer. Accordingly, the required computation effort is considerably reduced. The new algorithm is very promising. In particular, for polynomial systems without special structures, the new algorithm outperformed the existing methods by a big margin.

The purpose of this dissertation is to present a strategy of solving polynomial systems by polyhedral homotopy efficiently via newly developed mixed cell calculation. The polyhdreal homotopy and some necessary terminologies are introduced in Chapter 1. In Chapter 2, we give a basic linear programming algorithm which serves as a main tool for the mixed cell calculation presented in Chapter 3. Our algorithms have been implemented successfullly, the numerical results on substantial variety of examples are presented in Chapter 4.

CHAPTER 1

Polyhedral Homotopy

The Bernshtein theory on root count of polynomial systems is essential for our attempt to reduce the number of homotopy curves need to be traced when the homotopy continuation method is employed to find all isolated zeros of polynomial systems.

In the first section of this chapter, the Bernshtein theory on root count in $(\mathbb{C}^*)^n$, where $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$, as well as its extension to root count in $\mathbb{C}^n$ are presented. In the second section, the polyhedral homotopy, based on the Bershtein theory, for finding all isolated zeros of a polynomial system is introduced. In the last section, we will disscuss how to solve a binomial system to obtain initial solutions of a polyhedral homotopy.

1.1 Bernshtein Theory

Let the given polynomial system be $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x})) \in \mathbb{C}[\mathbf{x}]$, where $\mathbf{x} = (x_1, \cdots, x_n)$. With $\mathbf{x}^{\mathbf{a}} = x_1^{a_1} \cdots x_n^{a_n}$ where $\mathbf{a} = (a_1, \cdots, a_n)$, write

$$\begin{aligned} p_1(\mathbf{x}) &= \sum_{\mathbf{a} \in S_1} c^*_{1,\mathbf{a}} x^{\mathbf{a}}, \\ &\vdots \\ p_n(\mathbf{x}) &= \sum_{\mathbf{a} \in S_n} c^*_{n,\mathbf{a}} x^{\mathbf{a}}, \end{aligned} \tag{1.1}$$

where $S_1, \cdots, S_n$ are fixed subsets of $\mathbb{N}^n$ with cardinals $k_j = \#S_j$, and $c^*_{j,\mathbf{a}} \in \mathbb{C}^*$ for $\mathbf{a} \in S_j, j = 1, \cdots, n$. We call S_j the *support* of $p_j(\mathbf{x})$, denoted by *supp*(p_j), its convex hull $K_j = conv(S_j)$ in $\mathbb{R}^n$ the *Newton polytope* of p_j, and $S = (S_1, \cdots, S_n)$ the *support* of $P(\mathbf{x})$, denoted by *supp*(P).

We now embed the system (1.1) in the system $P(\mathbf{c}, \mathbf{x}) = (p_1(\mathbf{c}, \mathbf{x}), \cdots, p_n(\mathbf{c}, \mathbf{x}))$ where

$$\begin{aligned} p_1(\mathbf{c}, \mathbf{x}) &= \sum_{\mathbf{a} \in S_1} c_{1,\mathbf{a}} x^{\mathbf{a}}, \\ &\vdots \\ p_n(\mathbf{c}, \mathbf{x}) &= \sum_{\mathbf{a} \in S_n} c_{n,\mathbf{a}} x^{\mathbf{a}}, \end{aligned} \tag{1.2}$$

and the coefficients $c_{j,\mathbf{a}}$ with $\mathbf{a} \in S_j$, for $j = 1, \cdots, n$ in the system are taken to be a set of $M \equiv k_1 + \cdots + k_n$ variables. Namely, the system $P(\mathbf{x})$ in (1.1) is considered as a system in (1.2) corresponding to a set of specified values of coefficients $\bar{\mathbf{c}} = (\mathbf{c}^*_{j,\mathbf{a}})$ or $P(\mathbf{x}) = P(\bar{\mathbf{c}}, \mathbf{x})$.

We shall refer to the total number of isolated zeros, counting multiplicities, of a polynomial system as the *root count* of the system.

Lemma 1.1 [17] *For polynomial systems $P(\mathbf{c}, \mathbf{x})$ in (1.2), there exists a polynomial system $G(\mathbf{c}) = (g_1(\mathbf{c}), \cdots, g_n(\mathbf{c}))$ in the variables $\mathbf{c} = (\mathbf{c}_{j,\mathbf{a}})$ for $\mathbf{a} \in S_j$ and $j = 1, \cdots, n$ such that for those coefficients $\bar{\mathbf{c}} = (\mathbf{c}^*_{j,\mathbf{a}})$ for which $G(\bar{\mathbf{c}}) \neq 0$, the root count in $(\mathbb{C}^*)^n$ of the corresponding polynomial systems in (1.2) is a fixed number. And the root count in $(\mathbb{C}^*)^n$ of any other polynomial systems in (1.2) is bounded above by this number.*

Remark 1.2 *Since the zeros of the polynomial system $G(\mathbf{c})$ in the above lemma form an algebraic set with dimension smaller than M, its complement is open and dense with full measure in $\mathbb{C}^M$. Therefore, with probability one, $G(\bar{\mathbf{c}}) \neq 0$ for randomly* *chosen coefficients* $\bar{\mathbf{c}} = (\mathbf{c}^*_{j,\mathbf{a}}) \in \mathbb{C}^M$. *Hence, polynomial systems* $P(\bar{\mathbf{c}}, \mathbf{x})$ *in (1.2) with* $G(\bar{\mathbf{c}}) \neq 0$ *are said to be in* ***general position***.

Theorem 1.3 ([4], Theorem A) *The root count in* $(\mathbb{C}^*)^n$ *of a polynomial system* $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$ *in general position equals to the* ***mixed volume*** *of its support.*

The terminology in this theorem needs explanation. For non-negative variables $\lambda_1, \cdots, \lambda_n$ and the Newton polytopes K_j of p_j, for $j = 1, \cdots, n$, let $\lambda_1 K_1 + \cdots + \lambda_n K_n$ denote the *Minkowski sum* of $\lambda_1 K_1, \cdots, \lambda_n K_n$, that is,

$$\lambda_1 K_1 + \cdots + \lambda_n K_n = \{\lambda_1 r_1 + \cdots + \lambda_n r_n | r_j \in K_j, j = 1, \cdots, n\}.$$

It can be shown that the n-dimensional volume of this polytope $Vol_n(\lambda_1 K_1 + \cdots + \lambda_n K_n)$ is a homogeneous polynomial of degree n in $\lambda_1, \cdots, \lambda_n$. The coefficient of the term $\lambda_1 \times \cdots \times \lambda_n$ in this homogeneous polynomial is called the *mixed volume* of the polytopes $K_1, \cdots, K_n$, denoted by $\mathcal{M}(K_1, \cdots, K_n)$, or the mixed volume of the support of the system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$, denoted by $\mathcal{M}(S_1, \cdots, S_n)$ where $S_j = supp(p_j)$ for $j = 1, \cdots, n$. Sometimes, when no ambiguities exist, it is called the mixed volume of $P(\mathbf{x})$.

In [6], this root count was nicknamed the BKK bound after its inventors, Bernshtein [4], Kushnirenko [21] and Khovanskii [20]. In general, it provides a much tighter bound compared to variant Bézout bounds [32, 39]. An apparent limitation of the theorem is that it only counts the isolated zeros of polynomial systems in $(\mathbb{C}^*)^n$ rather than all the isolated zeros in the affine space $\mathbb{C}^n$. For the purpose of finding all the isolated zeros of a polynomial system in $\mathbb{C}^n$, a generalized version of the theorem which counts the roots in $\mathbb{C}^n$ is strongly desirable. This problem was first attempted in [36] where the notion of the *shadowed* sets was introduced and a bound for the root count in $\mathbb{C}^n$ was obtained. Later, a significantly much tighter bound was discovered in the following theorem.

Theorem 1.4 [27] *The root count in* $\mathbb{C}^n$ *of a polynomial system* $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ *with supports* $S_j = supp(p_j), j = 1, \cdots, n$ *is bounded above by the mixed volume* $\mathcal{M}(S_1 \bigcup\{0\}, \ldots, S_n \bigcup\{0\})$.

In other words, the theorem says that the root count in $\mathbb{C}^n$ of a polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ is bounded above by the root count in $(\mathbb{C}^*)^n$ of the polynomial system $\bar{P}(\mathbf{x})$ in general position obtained by augmenting the constant term to those $p_j's$ in $P(\mathbf{x})$ in which the constant term is absent. As a corollary, when $0 \in S_j$ for all $j = 1, \cdots, n$, namely, all $p_j(\mathbf{x})$ in $P(\mathbf{x})$ have constant terms, then the mixed volume of $P(\mathbf{x})$ also serves as a bound for the root count of $P(\mathbf{x})$ in $\mathbb{C}^n$, rather than in $(\mathbb{C}^*)^n$ as Theorem 1.3 asserts.

This theorem was further extended in several different ways [19, 37].

1.2 Polyhedral Homotopy

In light of Theorem 1.4 given in the last section, to find all isolated zeros of a given polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ in $\mathbb{C}^n$ with support $S = (S_1, \cdots, S_n)$, we first augment the monomial $\mathbf{x}^0$ (=1) to those p_j's which do not have constant terms. Followed by choosing coefficients of all the monomials in the system generically, a new system $Q(\mathbf{x}) = (q_1(\mathbf{x}), \cdots, q_n(\mathbf{x}))$ with support $S' = (S'_1, \cdots, S'_n)$ is obtained, where, of course, $S'_j = S_j \cup \{0\}$ for $j = 1, \cdots, n$. We will solve this system in the first place, and the details will be discussed in this section. Afterwards, in Chapter 4, we will present our algorithm to solve $P(\mathbf{x}) = 0$.

To begin, we write

$$Q(\mathbf{x}) = \begin{cases} q_1(\mathbf{x}) &= \sum_{\mathbf{a}\in S'_1} \bar{c}_{1,\mathbf{a}}\mathbf{x}^{\mathbf{a}}, \\ \vdots & \\ q_n(\mathbf{x}) &= \sum_{\mathbf{a}\in S'_n} \bar{c}_{n,\mathbf{a}}\mathbf{x}^{\mathbf{a}}. \end{cases} \tag{1.3}$$

Since all those coefficients $\bar{c}_{j,\mathbf{a}}$, for $\mathbf{a} \in S'_j$, $j = 1, \cdots, n$, are chosen generically, this system may be considered as a system in general position. Namely, there exists a polynomial system

$$G(\mathbf{c}) = (g_1(\mathbf{c}), \cdots, g_m(\mathbf{c})) \tag{1.4}$$

in the variables $\mathbf{c} = (c_{j,\mathbf{a}})$, for $\mathbf{a} \in S'_j$, $j = 1, \cdots, n$, such that polynomial systems with $G(\mathbf{c}) \neq 0$ reach the maximum root count in $(\mathbb{C}^*)^n$ for the support $S' = (S'_1, \cdots, S'_n)$ and we have $G(\bar{\mathbf{c}}) \neq 0$ for the set of coefficients $\bar{\mathbf{c}} = (\bar{c}_{j,\mathbf{a}})$ in (1.3).

Let t denote a new complex variable and consider the polynomial system $\hat{Q}(\mathbf{x}, t) = (\hat{q}_1(\mathbf{x}, t), \cdots, \hat{q}_n(\mathbf{x}, t))$ in the $n + 1$ variables $(\mathbf{x}, t)$ given by

$$\hat{Q}(\mathbf{x}, t) = \begin{cases} \hat{q}_1(\mathbf{x}, t) &= \sum_{\mathbf{a}\in S'_1} \bar{c}_{1,\mathbf{a}}\mathbf{x}^{\mathbf{a}} t^{w_1(\mathbf{a})}, \\ \vdots & \\ \hat{q}_n(\mathbf{x}, t) &= \sum_{\mathbf{a}\in S'_n} \bar{c}_{n,\mathbf{a}}\mathbf{x}^{\mathbf{a}} t^{w_n(\mathbf{a})}, \end{cases} \tag{1.5}$$

where each $w_j : S'_j \to \mathbb{R}$ for $j = 1, \cdots, n$ is chosen generically and known as a *lifting* on S'_j. For a fixed t_0, we rewrite the system in (1.5) as

$$\hat{Q}(\mathbf{x}, t_0) = \begin{cases} \hat{q}_1(\mathbf{x}, t_0) &= \sum_{\mathbf{a}\in S'_1} (\bar{c}_{1,\mathbf{a}} t_0^{w_1(\mathbf{a})})\mathbf{x}^{\mathbf{a}}, \\ \vdots & \\ \hat{q}_n(\mathbf{x}, t_0) &= \sum_{\mathbf{a}\in S'_n} (\bar{c}_{n,\mathbf{a}} t_0^{w_n(\mathbf{a})})\mathbf{x}^{\mathbf{a}}. \end{cases}$$

This system is in general position if for $G(\mathbf{c})$ in (1.4),

$$T(t_0) \equiv G(\bar{c}_{j,\mathbf{a}} t_0^{w_j(\mathbf{a})}) \neq 0, \text{ for } \mathbf{a} \in S'_j \text{ and } j = 1, \cdots, n.$$

The system $T(t) = 0$ can have at most finitely many solutions, since $T(t)$ is not identically 0 because $T(1) = G(\bar{c}_{j,\mathbf{a}}) \neq 0$. Let

$$t_1 = r_1 e^{i\theta_1}, \cdots, t_k = r_k e^{i\theta_k}$$

be the solutions of $T(t) = 0$. Then, for any $\theta \neq \theta_j$ for $j = 1, \cdots, k$, the systems $\bar{Q}(\mathbf{x}, t) = (\bar{q}_1(\mathbf{x}, t), \cdots, \bar{q}_n(\mathbf{x}, t))$ given by

$$\bar{Q}(\mathbf{x}, t) = \begin{cases} \bar{q}_1(\mathbf{x}, t) & = \sum_{\mathbf{a} \in S'_1} (\bar{c}_{1,\mathbf{a}} e^{i w_1(\mathbf{a})\theta}) \mathbf{x}^{\mathbf{a}} t^{w_1(\mathbf{a})}, \\ \vdots & \\ \bar{q}_n(\mathbf{x}, t) & = \sum_{\mathbf{a} \in S'_n} (\bar{c}_{n,\mathbf{a}} e^{i w_n(\mathbf{a})\theta}) \mathbf{x}^{\mathbf{a}} t^{w_n(\mathbf{a})}, \end{cases}$$

are in general position for all $t > 0$ because

$$\bar{c}_{j,\mathbf{a}} e^{i w_j(\mathbf{a})\theta} t^{w_j(\mathbf{a})} = \bar{c}_{j,\mathbf{a}} (t e^{i\theta})^{w_j(\mathbf{a})}$$

and,

$$G(\bar{c}_{j,\mathbf{a}} (t e^{i\theta})^{w_j(\mathbf{a})}) = T(t e^{i\theta}) \neq 0.$$

Therefore, without loss of generality, (choose an angle θ at random and change the coefficients $\bar{c}_{j,\mathbf{a}}$ to $\bar{c}_{j,\mathbf{a}} e^{i w_j(a)\theta}$ if necessary) we may suppose the systems $\hat{Q}(\mathbf{x}, t)$ in (1.5) are in general position for all $t > 0$. Together with Lemma 1.1 given in the last section, it follows that for all $t > 0$ the systems $\hat{Q}(\mathbf{x}, t)$ in (1.5) have the same number of isolated zeros in $(\mathbb{C}^*)^n$. This number, say k, should equal to the mixed volume of the support of $Q(\mathbf{x})$ in (1.3) by Theorem 1.3. We shall skip this fact temporarily and will reach this assertion at the end of this section.

Now, consider $\hat{Q}(\mathbf{x}, t) = 0$ as a homotopy, known as the *polyhedral homotopy*, defined on $(\mathbb{C}^*)^n \times [0, 1]$. We have $\hat{Q}(\mathbf{x}, 1) = Q(\mathbf{x})$, and the zero set of this homotopy is made up of k homotopy paths, say, $\mathbf{x}^1(t), \cdots, \mathbf{x}^k(t)$, since for each $0 < t \le 1$, $\hat{Q}(\mathbf{x}, t)$ has exactly k isolated zeros from the argument given above. Since each $\hat{q}_j(\mathbf{x}, t)$ has nonzero constant term for all $j = 1, \cdots, n$, by a standard application of generalized Sard's Theorem [7], all those homotopy paths are smooth with no bifurcations. Therefore, both Property 2 (Smoothness) and Property 3 (Accessibility) introduced earlier hold for this homotopy. However, at $t = 0$, $\hat{Q}(\mathbf{x}, 0) \equiv 0$, see Figure 1.1. Consequently, the starting points $\mathbf{x}^1(0), \cdots, \mathbf{x}^n(0)$ of those homotopy paths can not be identified, causing the breakdown of the standard homotopy continuation algorithm. This major obstacle can be overcome by the devise we describe below.

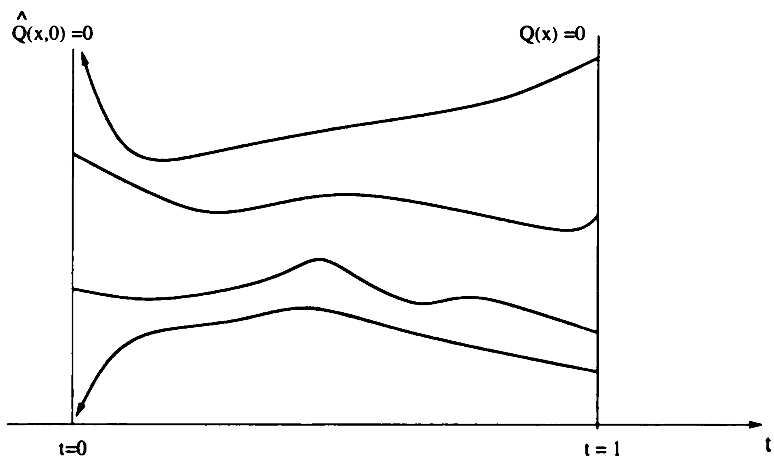

Figure 1.1: Solution curves of $\hat{Q}(x, t) = 0$

For $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$, consider the transformation $\mathbf{y} = t^{-\alpha}\mathbf{x}$ defined by

$$\begin{aligned} y_1 &= t^{-\alpha_1} x_1, \\ &\vdots \\ y_n &= t^{-\alpha_n} x_n. \end{aligned} \tag{1.6}$$

For $\mathbf{a} = (a_1, \cdots, a_n) \in \mathbf{N}^n$, we have

$$\begin{aligned}
\mathbf{x}^{\mathbf{a}} &= x_1^{a_1} \cdots x_n^{a_n}, \\
&= (y_1 t^{\alpha_1})^{a_1} \cdots (y_n t^{\alpha_n})^{a_n} \\
&= y_1^{a_1} \cdots y_n^{a_n}\, t^{\alpha_1 a_n + \cdots + \alpha_n a_n} \\
&= y^{\mathbf{a}} t^{\langle \alpha, \mathbf{a} \rangle}.
\end{aligned} \tag{1.7}$$

Here, $\langle \cdot, \cdot \rangle$ stands for the usual inner product in $\mathbb{R}^n$. Substituting (1.7) into (1.5) yields, for $j = 1, \cdots, n$

$$\begin{aligned}
h_j(\mathbf{y}, t) &\equiv \hat{q}_j(\mathbf{y}t^{\alpha}, t) = \sum_{\mathbf{a} \in S'_j} \bar{c}_{j,a} \mathbf{y}^{\mathbf{a}} t^{\langle \alpha, \mathbf{a} \rangle} t^{w_j(\mathbf{a})} \\
&= \sum_{\mathbf{a} \in S'_j} \bar{c}_{j,\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{\langle (\alpha, 1), (\mathbf{a}, w_j(\mathbf{a})) \rangle} \\
&= \sum_{\mathbf{a} \in S'_j} \bar{c}_{j,\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{\langle \hat{\alpha}, \hat{\mathbf{a}} \rangle},
\end{aligned} \tag{1.8}$$

where $\hat{\alpha} = (\alpha, 1) \in \mathbb{R}^{n+1}$, and $\hat{\mathbf{a}} = (\mathbf{a}, w_j(\mathbf{a}))$ for $\mathbf{a} \in S'_j$. The new homotopy

$$H(\mathbf{y}, t) = (h(\mathbf{y}_1, t), \cdots, h_n(\mathbf{y}, t)) = 0 \tag{1.9}$$

retains most of the properties of the homotopy $\hat{Q}(\mathbf{x}, t) = 0$, in particular, $H(\mathbf{y}, 1) = \hat{Q}(\mathbf{y}, 1) = Q(\mathbf{y})$ and both Properties 2 (Smoothness) and 3 (Accessibility) stand. Let

$$\beta_j = \min_{\mathbf{a} \in S'_j} \langle \hat{\alpha}, \hat{\mathbf{a}} \rangle, \quad j = 1, \cdots, n \tag{1.10}$$

and define the homotopy

$$H_\alpha(\mathbf{y}, t) = (h_1^\alpha(\mathbf{y}, t), \cdots, h_n^\alpha(\mathbf{y}, t)) = 0 \tag{1.11}$$

on $(\mathbb{C}^*)^n \times [0,1]$ where, for $j = 1, \cdots, n$

$$\begin{aligned} h_j^\alpha(\mathbf{y},t) &\equiv t^{-\beta_j} h_j(\mathbf{y},t) = \sum_{\mathbf{a}\in S_j'} \bar{c}_{j,\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{\langle \hat{\alpha},\hat{\mathbf{a}}\rangle - \beta_j} \\ &= \sum_{\substack{\mathbf{a}\in S_j' \\ \langle \hat{\alpha},\hat{\mathbf{a}}\rangle = \beta_j}} \bar{c}_{j,\mathbf{a}} \mathbf{y}^{\mathbf{a}} + \sum_{\substack{\mathbf{a}\in S_j' \\ \langle \hat{\alpha},\hat{\mathbf{a}}\rangle > \beta_j}} \bar{c}_{j,\mathbf{a}} \mathbf{y}^{\mathbf{a}} t^{\langle \hat{\alpha},\hat{\mathbf{a}}\rangle - \beta_j}. \end{aligned} \tag{1.12}$$

Evidently, for any path $\tilde{\mathbf{y}}(t)$ defined on $[0,1]$, we have, for all $t > 0$,

$$H_\alpha(\tilde{\mathbf{y}}(t),t) = 0 \quad \Longleftrightarrow \quad H(\tilde{\mathbf{y}}(t),t) = 0.$$

Therefore, the zero set of $H_\alpha(\mathbf{y},t) = 0$ consists of the same homotopy paths of the homotopy $H(\mathbf{y},t) = 0$ in (1.9). The difference is, the starting points of the homotopy paths considered in the homotopy $H_\alpha(\mathbf{y},t) = 0$ are solutions of the system

$$H_\alpha(\mathbf{y},0) = \begin{cases} h_1^\alpha(\mathbf{y},0) = \displaystyle\sum_{\substack{\mathbf{a}\in S_1' \\ \langle \hat{\alpha},\hat{\mathbf{a}}\rangle = \beta_1}} \bar{c}_{1,\mathbf{a}} \mathbf{y}^{\mathbf{a}} = 0, \\ \quad\vdots \\ h_n^\alpha(\mathbf{y},0) = \displaystyle\sum_{\substack{\mathbf{a}\in S_n' \\ \langle \hat{\alpha},\hat{\mathbf{a}}\rangle = \beta_n}} \bar{c}_{n,\mathbf{a}} \mathbf{y}^{\mathbf{a}} = 0. \end{cases} \tag{1.13}$$

As shown below, when this system is in certain desired form, its isolated nonsingular solutions that lie in $(\mathbb{C}^*)^n$ can be constructively identified. In those situations, Property 1 (Triviality) becomes partially valid for those homotopy paths of $H_\alpha(y,t) = 0$ that emanate from those nonsingular solutions of (1.13) in $(\mathbb{C}^*)^n$, and we may follow those paths to reach a partial set of isolated zeros of $Q(\mathbf{y})$ at $t = 1$.

The system (1.13) is known as the *binomial system* if each $h_j^\alpha(\mathbf{y},0)$ consists of exactly two terms, that is,

$$\begin{aligned} h_1^\alpha(\mathbf{y},0) &= c_1 \mathbf{y}^{\mathbf{a}_1} + c_1' y^{\mathbf{a}_1'} = 0, \\ &\vdots \\ h_n^\alpha(\mathbf{y},0) &= c_n \mathbf{y}^{\mathbf{a}_n} + c_n' y^{\mathbf{a}_n'} = 0, \end{aligned} \tag{1.14}$$

where $\mathbf{a}_j, \mathbf{a}'_j \in S'_j$, $c_j = \bar{c}_{j,\mathbf{a}_j}$ and $c'_j = \bar{c}_{j,\mathbf{a}'_j}$ for $j = 1, \cdots, n$. And in this case, $(\{\mathbf{a}_1, \mathbf{a}'_1\}, \cdots, \{\mathbf{a}_n, \mathbf{a}'_n\})$ is called a **mixed cell** (of type $(1, \cdots, 1)$) of $S' = (S'_1, \cdots, S'_n)$ associated with inner normal $\hat{\alpha}$.

Proposition 1.5 *The binomial system in (1.14) has*

$$k_\alpha \equiv \left| \det \begin{pmatrix} \mathbf{a}_1 - \mathbf{a}'_1 \\ \vdots \\ \mathbf{a}_n - \mathbf{a}'_n \end{pmatrix} \right| \tag{1.15}$$

nonsingular solutions in $(\mathbb{C}^*)^n$.

The number k_α is called the *volume* of the mixed cell $(\{\mathbf{a}_1, \mathbf{a}'_1\}, \cdots, \{\mathbf{a}_n, \mathbf{a}'_n\})$. The proof of this proposition is constructive and therefore provides an algorithm for solving the binomial system (1.14) in $(\mathbb{C}^*)^n$. We will come back to this matter in the next section.

In summary, for given $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$, by changing variables $\mathbf{y} = t^{-\alpha}\mathbf{x}$, as in (1.6), in the homotopy $\hat{Q}(\mathbf{x}, t) = (\hat{q}_1(\mathbf{x}, t), \cdots, \hat{q}_n(\mathbf{x}, t)) = 0$ in (1.5), the homotopy $H(\mathbf{y}, t) = (h_1(\mathbf{y}, t), \cdots, h_n(\mathbf{y}, t)) = 0$ in (1.9) is obtained, where $h_j(\mathbf{y}, t) = \hat{q}_j(\mathbf{y}t^\alpha, t)$. Followed by factoring out the lowest power t^{β_j} of t among all monomials in each individual $h_j(\mathbf{y}, t) = 0$ for $j = 1, \cdots, n$ we arrive at the homotopy $H_\alpha(\mathbf{y}, t) = 0$ in (1.11). When the start system $H_\alpha(\mathbf{y}, 0) = 0$ of this homotopy is binomial, its nonsingular solutions in $(\mathbb{C}^*)^n$, k_α (as given in (1.15)) of them, become available. We may then follow those homotopy paths of $H_\alpha(\mathbf{y}, t) = 0$ originated from those k_α regular solutions of $H_\alpha(\mathbf{y}, 0) = 0$ in $(\mathbb{C}^*)^n$, and reach k_α isolated zeros of $Q(\mathbf{y})$ at $t = 1$. Worth notifying here is the fact that the system $Q(\mathbf{x})$, or $Q(\mathbf{y})$, stays invariant at $t = 1$ during the process. Now, the existence of $\alpha \in \mathbb{R}^n$ for which the start system $H_\alpha(\mathbf{y}, 0) = 0$ is binomial is warranted by the following

Proposition 1.6 *For all the real functions $w_j : S'_j \to \mathbb{R}$, $j = 1, \cdots, n$ being generically chosen, there must exist $\alpha \in \mathbb{R}^n$, for which the start system $H_\alpha(\mathbf{y}, 0) = 0$ of the homotopy $H_\alpha(\mathbf{y}, t) = 0$ in (1.12) is binomial with a nonempty set of nonsingular solutions in $(\mathbb{C}^*)^n$, i.e., $k_\alpha \neq 0$ in (1.15).*

The assertion of this proposition was proved implicitly in [18] with terminologies and machineries developed in combinatorial geometry, such as, *random liftings*, *fine mixed subdivisions*, *lower facets of convex polytopes*, etc., see also [23]. Here, we elect to reinterpret the result without those specialized terms.

Now, different $\alpha \in \mathbb{R}^n$ given in Proposition 1.6 leads to different homotopy $H_\alpha(\mathbf{y}, t) = 0$ in (1.11). Henceforth, following homotopy paths of those different homotopies will reach different sets of isolated zeros of $Q(\mathbf{y})$. By taking the Puiseux series expansions of those homotopy paths of $H_\alpha(\mathbf{y}, t) = 0$ originated at $(\mathbb{C}^*)^n$ into consideration, it is not hard to see that those different sets of isolated zeros of $Q(\mathbf{y})$ reached by different sets of homotopy paths actually disjoint from each other. Most importantly, it can be shown that every isolated zero of $Q(\mathbf{y})$ can be obtained this way by following certain homotopy curve of the homotopy $H_\alpha(y, t) = 0$ associated with certain $\alpha \in \mathbb{R}^n$ given by Proposition 1.6. Thus the total number of isolated zeros of $Q(\mathbf{y})$ must equal to the sum of those k_α's corresponding to all the possible α's provided by Proposition 1.6, respectively. In [18], it was shown that this sum actually equal to the mixed volume of $Q(\mathbf{y})$. This yields an alternative proof of Theorem 1.3, it is very different from Bernshtein's original approach [4].

1.3 Solve Binomial System

Another major step in solving polynomial systems by using the polyhedral homotopy method as we described in the previous section is finding the solutions of the corresponding binomial system

$$
\begin{aligned}
c_1\mathbf{y}^{\mathbf{a}_1} + c_1'y^{\mathbf{a}_1'} &= 0, \\
&\vdots \\
c_n\mathbf{y}^{\mathbf{a}_n} + c_n'y^{\mathbf{a}_n'} &= 0,
\end{aligned}
\tag{1.16}
$$

produced by the mixed cell $(\{\mathbf{a}_1, \mathbf{a}_1'\}, \cdots, \{\mathbf{a}_n, \mathbf{a}_n'\})$ as in (1.14). We now discuss the method for solving (1.16) in $(\mathbb{C}^*)^n$. Let

$$
v_j = \mathbf{a}_j - \mathbf{a}_j', \qquad j = 1, \cdots, n,
$$

and, with $\mathbf{y} \in (\mathbb{C}^*)^n$ in mind, we rewrite the system (1.16) as

$$
\begin{aligned}
\mathbf{y}^{v_1} &= b_1, \\
&\vdots \\
\mathbf{y}^{v_n} &= b_n,
\end{aligned}
\tag{1.17}
$$

where $b_j = -\dfrac{c_j'}{c_j}$ for $j = 1, \cdots, n$. Let

$$
V = \left[\begin{array}{c|c|c|c} v_1^T & v_2^T & \cdots & v_n^T \end{array} \right] \tag{1.18}
$$

and for brevity, write

$$
\mathbf{y}^V = (\mathbf{y}^{v_1}, \cdots, \mathbf{y}^{v_n}) \quad \text{and} \quad \mathbf{b} = (b_1, \cdots, b_n).
$$

Then, (1.17) becomes,

$$
\mathbf{y}^V = \mathbf{b}. \tag{1.19}
$$

With this notation, it is easy to verify that for an $n \times n$ integer matrix U, we have,

$$
(\mathbf{y}^V)^U = \mathbf{y}^{(VU)}.
$$

Now, when the matrix V in (1.18) is an upper triangular matrix, i.e.,

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ 0 & v_{22} & \cdots & v_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & v_{nn} \end{bmatrix},$$

then the equation in (1.19) becomes

$$\begin{aligned} y_1^{v_{11}} &= b_1, \\ y_1^{v_{12}} y_2^{v_{22}} &= b_2, \\ &\vdots \\ y_1^{v_{1n}} y_2^{v_{2n}} \cdots y_n^{v_{nn}} &= b_n. \end{aligned} \tag{1.20}$$

By forward substitutions, all the solutions of the system (1.20) in $(\mathbb{C}^*)^n$ can be found, and the total number of solutions is $|v_{11}| \times \cdots \times |v_{nn}| = |\det V|$.

In general, we may upper triangularize V in (1.18) by the following process. Recall that the greatest common divisor d of two nonzero integers a and b, denoted by $\gcd(a,b)$, can be written as

$$d = \gcd(a,b) = ka + lb,$$

for certain nonzero integers k and l. Let

$$M = \begin{bmatrix} k & l \\ -\frac{b}{d} & \frac{a}{d} \end{bmatrix}.$$

We have $\det(M) = 1$, and

$$M \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} k & l \\ -\frac{b}{d} & \frac{a}{d} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

Similar to using Givens rotation to produce zeros in a matrix for its QR factorization, the matrix M may be used to upper triangularize V as follows. For $v \in Z^n$, let a

and b be its i-th and the j-th (nonzero) components where $i < j$, that is,

$$v = \begin{bmatrix} \vdots \\ a \\ \vdots \\ b \\ \vdots \end{bmatrix} \begin{array}{l} \\ \to i\text{-th} \\ \\ \to j\text{-th.} \\ \\ \end{array}$$

With $d = \gcd(a, b)$, we let

$$U(i,j) = \begin{array}{c} \begin{array}{ccccccccccc} & & & i\text{-th} & & & & j\text{-th} & & & \end{array} \\ \begin{bmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & k & & & & l & & & \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & -\frac{b}{d} & & & & \frac{a}{d} & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \\ i\text{-th} \\ \\ \\ \\ j\text{-th} \\ \\ \\ \\ . \end{array} \tag{1.21}$$

Evidently, $U(i,j)$ is an integer matrix with $|\det(U(i,j))| = 1$ and

$$U(i,j)v = \left[\begin{array}{c} \vdots \\ d \\ \vdots \\ 0 \\ \vdots \end{array}\right] \begin{array}{l} \\ i\text{-th} \\ \\ j\text{-th} \\ \\ \end{array} .$$

Thus a series of matrices in the form of $U(i,j)$ in (1.21) may be used to successively produce zeros in the lower triangular part of the matrix V in (1.18), resulting in an upper triangular matrix. In simple terms, we may construct an integer matrix U , as a product of those $U(i,j)$'s, with $|\det U| = 1$ and UV is an upper triangular integer matrix.

Now, as mentioned above, the solutions of the system

$$(\mathbf{z}^U)^V = \mathbf{z}^{UV} = \mathbf{b} \tag{1.22}$$

in $(\mathbb{C}^*)^n$ can be found by forward substitutions, since UV is an upper triangular integer matrix. And the total number of solutions in $(\mathbb{C}^*)^n$ is

$$|\det(UV)| = |\det(U)| \cdot |\det(V)| = |\det(V)|.$$

By letting $\mathbf{y} = \mathbf{z}^U$ for each solution $\mathbf{z}$ of (1.22) in $(\mathbb{C}^*)^n$, we obtain all the solutions of the system (1.22) in $(\mathbb{C}^*)^n$, and hence, solve the system (1.16) in $(\mathbb{C}^*)^n$.

CHAPTER 2

Linear Programming

As outlined in the last chapter, when the polyhedral homotopy is employed to find all the isolated zeros of a polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ with supports $S_1, \cdots, S_n$, one major step is to indentify the mixed cells $(\{\mathbf{a}_1, \mathbf{a}_1'\}, \cdots, \{\mathbf{a}_n, \mathbf{a}_n'\})$ induced by generic liftings $w_j : S_j \to \mathbb{R}$ for $j = 1, \cdots, n$. As a point of departure in developing our algorithms for finding all mixed cells in the next chapter, we introduce in this chapter some basic terminologies and algorithms in linear programming [3] that will be used in the method.

Consider the model problem

$$\begin{aligned} \min \quad & \langle \mathbf{f}, \mathbf{x} \rangle \\ s.t. \quad & \langle \mathbf{c}_i, \mathbf{x} \rangle \;\leq\; b_i, \quad i = 1, \cdots, m \end{aligned} \tag{2.1}$$

where $\mathbf{f} \in \mathbb{R}^n$, $\mathbf{c}_i \in \mathbb{R}^n$, $\mathbf{b} = (b_1, \cdots, b_m)^T \in \mathbb{R}^m$, $\mathbf{x} = (x_1, \cdots, x_n)$, $m > n$. The feasible region of (2.1), denoted by R, defines a polyhedral set. By a *nondegenerate* extreme point of R we mean a vertex point of R with exactly n active constraints.

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

$$\begin{aligned} \langle \mathbf{c}_i, \mathbf{x}^0 \rangle &= b_i, \text{ if } i \in J \\ \langle \mathbf{c}_i, \mathbf{x}^0 \rangle &< b_i, \text{ if } i \notin J. \end{aligned}$$

Let $D^T = [\mathbf{c}_{j_1}, \cdots . \mathbf{c}_{j_n}]$. Since $\mathbf{x}^0$ is a vertex point, D must be nonsingular. Let $D^{-1} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$. Then for any $\sigma > 0$ and $1 \leq k \leq n$, we have

$$\begin{aligned} \langle \mathbf{c}_{j_i}, \mathbf{x}^0 - \sigma \mathbf{u}_k \rangle &= \langle \mathbf{c}_{j_i}, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{c}_{j_i}, \mathbf{u}_k \rangle = \langle \mathbf{c}_{j_i}, \mathbf{x}^0 \rangle = b_i, \text{ if } i \neq k, \\ \langle \mathbf{c}_{j_k}, \mathbf{x}^0 - \sigma \mathbf{u}_k \rangle &= \langle \mathbf{c}_{j_k}, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{c}_{j_k}, \mathbf{u}_k \rangle = b_{j_k} - \sigma < b_{j_k} \end{aligned} \tag{2.2}$$

and for small $\sigma > 0$,

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

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

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

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

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

and it decreases when $\langle \mathbf{f}, \mathbf{u}_i \rangle > 0$. It can be easily shown that $\mathbf{x}^0$ is an optimal solution of (2.1) if $\langle \mathbf{f}, \mathbf{u}_i \rangle \leq 0$ for all $i = 1, \cdots, n$. If some of the $\langle \mathbf{f}, \mathbf{u}_i \rangle$'s are positive, then the greatest rate of decrease of the cost function is obtained by choosing k such that

$$\langle \mathbf{f}, \mathbf{u}_k \rangle = \max \{ \langle \mathbf{f}, \mathbf{u}_i \rangle \mid 1 \leq i \leq n \}.$$

Let $\mathbf{s} = \mathbf{u}_k$ be the next search direction. From (2.2), for all positive σ, the i-th 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_k-th constraint becomes inactive but stays feasible. To make $\mathbf{x}^1$ feasible, we must choose $\sigma > 0$ such that

$$\langle \mathbf{c}_i, \mathbf{x}^0 - \sigma \mathbf{s} \rangle = \langle \mathbf{c}_i, \mathbf{x}^0 \rangle - \sigma \langle \mathbf{c}_i, \mathbf{s} \rangle \leq b_i, \text{ for } i \notin J. \tag{2.3}$$

If $\langle \mathbf{c}_i, \mathbf{s} \rangle \geq 0$ for all $i \notin J$, then the inequalities in (2.3) are valid for all $\sigma > 0$ and problem (2.1) is unbounded from below with no solution. Otherwise, from (2.3), the largest possible σ for $\mathbf{x}^1$ to stay feasible is

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

Let l be the smallest integer such that

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

Then $\mathbf{x}^1 = \mathbf{x}^0 - \sigma_0 \mathbf{s}$ is a new extreme point of the feasible region R in (2.1) with reduced value of the objective function. This procedure can be continued until either an optimal solution is reached or the problem is determined to be unbounded from below.

We summarize the above discussion in Algorithm 1 below [3].

Algorithm 1 *Solving the model problem (2.1) .*

Step 0: ***Initialization.***

Start with an extreme point $\mathbf{x}^0$ of (2.1), $J = \{i_1, \cdots, i_n\}$, And $D^{-1} = [u_{ij}] = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$, where $D^T = [\mathbf{d}_1, \cdots, \mathbf{d}_n] = [\mathbf{c}_{i_1}, \cdots, \mathbf{c}_{i_n}]$ is nonsingular.

Step 1: ***Computation of the search direction $\mathbf{s}$.***

Determine the smallest index k such that

$$\langle \mathbf{f}, \mathbf{u}_k \rangle = \max\{\langle \mathbf{f}, \mathbf{u}_i \rangle \mid i = 1, \cdots, n\}.$$

If $\langle \mathbf{f}, \mathbf{u}_k \rangle \leq 0$, stop with optimal solution $\mathbf{x}^0$. Otherwise, set $\mathbf{s} = \mathbf{u}_k$ and go to Step 2.

Step 2: ***Compute the maximum feasible step size σ.***

If $\langle \mathbf{c}_i, \mathbf{s} \rangle \geq 0$ for all $i = 1, \cdots, m$, print the message "problem is unbounded from below" and stop. Otherwise, compute the smallest index l

and σ such that

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

and go to Step 3.

Step 3: ***Update.***

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

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

CHAPTER 3

Find Mixed Cells

In this chapter we will elaborate our algorithms for finding all mixed cells by solving a series of linear programming problems.

For $i = 1, \cdots, n$, let S_i be the support of $p_i(\mathbf{x})$ in the polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ and $w_i : S_i \to \mathbb{R}$ be a generically chosen function. Let

$$\hat{S}_i = \left\{ \hat{\mathbf{a}} = (\mathbf{a}, w_i(\mathbf{a})) \;\middle|\; \mathbf{a} \in S_i \right\}, \text{ for } i = 1, \cdots, n$$

and for $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$, write $\hat{\alpha} = (\alpha, 1)$. Recall that a mixed cell of $S = (S_1, \cdots, S_n)$ induced by the lifting $w = (w_1, \cdots, w_n)$ is a collection of pairs

$$\{\mathbf{a}_1, \mathbf{a}'_1\}, \cdots, \{\mathbf{a}_n, \mathbf{a}'_n\}, \text{ with } \mathbf{a}_i, \mathbf{a}'_i \in S_i, \; i = 1, \cdots, n$$

such that there exists an $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$ for which

$$\langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}'_i, \hat{\alpha} \rangle, \; i = 1, \cdots, n$$

and

$$\langle \hat{\mathbf{a}}, \hat{\alpha} \rangle > \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle \text{ for } \hat{\mathbf{a}} \in \hat{S}_i \backslash \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}'_i\}, \; i = 1, \cdots, n.$$

The geometric meaning of finding those mixed cells is that with generic *lifting* w_i on lattice points $S_i \subset \mathbb{N}^n$ for each $i = 1, \cdots, n$, we are looking for hyperplanes with

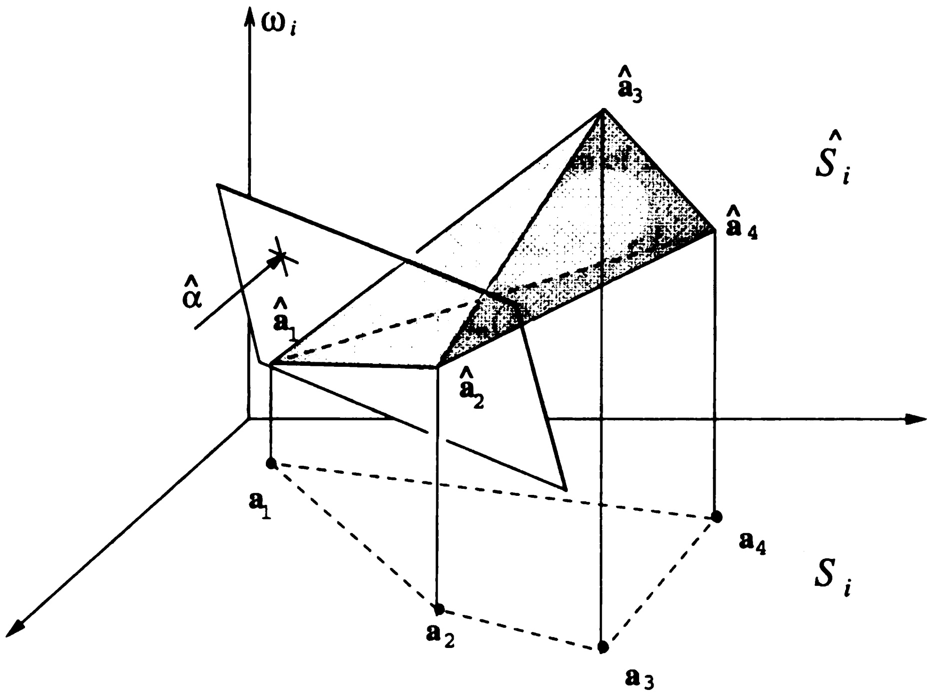

Figure 3.1: A lifting on lattice points

normal $\hat{\alpha} = (\alpha, 1)$ where $\alpha \in \mathbb{R}^n$, and each hyperplane supports the convex hull of $\hat{S}_i$ at exactly two points $\{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_i'\}$ of $\hat{S}_i$, for each $i = 1, \cdots, n$, as shown in Figure 3.1.

For $1 \leq i \leq n$, $\hat{\mathbf{e}} = \{\hat{\mathbf{a}}, \hat{\mathbf{a}}'\} \subset \hat{S}_i$ is called a *lower edge* of $\hat{S}_i$ if there is a vector $\hat{\alpha} = (\alpha, 1)$ with $\alpha \in \mathbb{R}^n$ such that

$$\begin{aligned} \langle \hat{\mathbf{a}}, \hat{\alpha} \rangle &= \langle \hat{\mathbf{a}}', \hat{\alpha} \rangle \\ \langle \hat{\mathbf{a}}, \hat{\alpha} \rangle &\leq \left\langle \hat{\mathbf{b}}, \hat{\alpha} \right\rangle, \quad \hat{\mathbf{b}} \in \hat{S}_i \backslash \{\hat{\mathbf{a}}, \hat{\mathbf{a}}'\}. \end{aligned}$$

For $1 \leq k \leq n$, $\hat{E}_k = (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_k)$ where $\hat{\mathbf{e}}_i = \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_i'\} \subset \hat{S}_i$, for $i = 1, \cdots, k$, is called a *level-k subface* of $\hat{S} = (\hat{S}_1, \cdots, \hat{S}_n)$ if there is a vector $\hat{\alpha} = (\alpha, 1)$ with $\alpha \in \mathbb{R}^n$ such that for all $i = 1, \cdots, k$,

$$\begin{aligned} \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle &= \langle \hat{\mathbf{a}}_i', \hat{\alpha} \rangle \\ \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle &\leq \left\langle \hat{\mathbf{b}}, \hat{\alpha} \right\rangle, \quad \hat{\mathbf{b}} \in \hat{S}_i \backslash \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_i'\}. \end{aligned}$$

Obviously, a level-1 subface of $\hat{S}$ is just a lower edge of $\hat{S}_1$ and a level-n subface of $\hat{S}$ induces a mixed cell of S. Thus, to find the mixed cells of S, one may proceed by finding all the lower edges of $\hat{S}$ for $i = 1, \cdots, n$ in the first place, followed by extending the level-k subfaces of $\hat{S}$ from $k = 1$ to $k = n$.

It can be shown that the mixed volume of S

$$\begin{aligned} \mathcal{M}(S) &= (-1)^{n-1} \sum_{i=1}^{n} \mathrm{Vol}_n(K_i) + (-1)^{n-2} \sum_{i<j} \mathrm{Vol}_n(K_i + K_j) \\ &\quad + \cdots + \mathrm{Vol}_n(K_1 + \cdots + K_n), \end{aligned}$$

where $K_i = \mathrm{conv}(S_i), i = 1, \cdots, n$. Thus non-extreme points of any of the S_i's play no role in the mixed volume of S.

To identify non-extreme points of S_i, notice that a non-extreme point of S_i is a convex combination of other points of S_i. Namely, if $\mathbf{a}_{ik} \in S_i$ is a non-extreme point of S_i, the following system of equations

$$\begin{gathered} \lambda_1 \mathbf{a}_{i1} + \cdots + \lambda_{k-1} \mathbf{a}_{ik-1} + \lambda_{k+1} \mathbf{a}_{ik+1} \cdots + \lambda_{m_i} \mathbf{a}_{im_i} = \mathbf{a}_{ik} \\ \lambda_1 + \cdots + \lambda_{k-1} + \lambda_{k+1} + \cdots + \lambda_{m_i} = 1 \\ \lambda_1, \cdots, \lambda_{k-1}, \lambda_{k+1}, \cdots, \lambda_{m_i} \geq 0. \end{gathered}$$

must have a solution. Testing the existence of solutions of the above system constitutes a standard Phase I problem in linear programming, and algorithms for this problem can be found in many standard Linear Programming books, e.g. [35]. When we compute the mixed cells of S we will eliminate all those non-extreme points in the first place and assume throughout this chapter that S_i has only extreme points for all $i = 1, \cdots, n$.

3.1 Find all lower Edges of a lifted Lattice Set

For $w = (w_1, \cdots, w_n)$ with generically chosen w_i, $i = 1, \cdots, n$, and

$$\hat{S}_i = \left\{ \hat{\mathbf{a}} = (\mathbf{a}, w_i(\mathbf{a})) \;\middle|\; \mathbf{a} \in S_i \right\} \quad \text{for } i = 1, \cdots, n,$$

denote the set of all lower edges of $\hat{S}_i$ by $\mathcal{L}(\hat{S}_i)$. In this section, we will describe our algorithm for finding $\mathcal{L}(\hat{S}_i)$ for $i = 1, \cdots, n$ efficiently. For this purpose, let

$\mathcal{B} = \{\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_m\} \subset \mathbb{N}^n$ represent general S_i's, and $w : \mathcal{B} \to \mathbb{R}$ be a generic lifting function. Let $\hat{\mathcal{B}}(w) = \{\hat{\mathbf{a}} = (\mathbf{a}, w(\mathbf{a})) \mid \mathbf{a} \in \mathcal{B}\}$. Consider the following system in the $n+1$ unknowns $\alpha_0, \alpha_1, \cdots, \alpha_n$

$$\langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle \geq \alpha_0, \; i = 0, \cdots, m \tag{3.1}$$

where $\hat{\alpha} = (\alpha_1, \cdots, \alpha_n, 1) \in \mathbb{R}^{n+1}$. Immediately, we have

Lemma 3.1 *If system (3.1) has a solution $\alpha_0, \alpha_1, \cdots, \alpha_n$ such that $\langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}_j, \hat{\alpha} \rangle = \alpha_0$ for $0 \leq i, j \leq m$, then $\{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j\}$ is a lower edge of $\hat{\mathcal{B}}$.*

With $\mathbf{a}_i = (a_{i,1}, \cdots, a_{i,n})$ for $i = 1, \cdots, m$, rewrite system (3.1) as

$$\begin{bmatrix} 1 & -a_{1,1} & \cdots & -a_{1,n} \\ 1 & -a_{2,1} & \cdots & -a_{2,n} \\ \vdots & \vdots & & \vdots \\ 1 & -a_{m,1} & \cdots & -a_{m,n} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \leq \begin{bmatrix} w(\mathbf{a}_1) \\ w(\mathbf{a}_2) \\ \vdots \\ w(\mathbf{a}_m) \end{bmatrix}. \tag{3.2}$$

Suppose the rank of the coefficient matrix in (3.2) is $v \leq n$. Without loss of generality, we may assume the first v rows are linearly independent. By Gaussian eliminations, there exists $L \in \mathbb{R}^{(n+1)\times(n+1)}$ such that

$$\begin{bmatrix} 1 & -a_{1,1} & \cdots & -a_{1,n} \\ 1 & -a_{2,1} & \cdots & -a_{2,n} \\ \vdots & \vdots & & \vdots \\ 1 & -a_{v,1} & \cdots & a_{v,n} \\ \vdots & \vdots & & \vdots \\ 1 & -a_{m,1} & \cdots & -a_{m,n} \end{bmatrix} \cdot L = \begin{bmatrix} c_{1,1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ c_{2,1} & c_{2,2} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ c_{v,1} & c_{v,2} & \cdots & c_{v,v} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,v} & 0 & \cdots & 0 \end{bmatrix},$$

where $c_{i,i} \neq 0$ for $i = 1, \cdots, v$. With $(x_1, \cdots, x_{n+1})^T := L^{-1}(\alpha_0, \alpha_1 \cdots, \alpha_n)^T$ in

$$\begin{bmatrix} 1 & -a_{1,1} & \cdots & -a_{1,n} \\ 1 & -a_{2,1} & \cdots & -a_{2,n} \\ \vdots & \vdots & & \vdots \\ c_{v,1} & c_{v,2} & \cdots & c_{v,n} \\ \vdots & \vdots & & \vdots \\ 1 & -a_{m,1} & \cdots & -a_{m,n} \end{bmatrix} L \cdot L^{-1} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \leq \begin{bmatrix} w(\mathbf{a}_1) \\ w(\mathbf{a}_2) \\ \vdots \\ w(\mathbf{a}_v) \\ \vdots \\ w(\mathbf{a}_m) \end{bmatrix} \tag{3.3}$$

we obtain the following system:

$$\begin{bmatrix} c_{1,1} & 0 & \cdots & 0 \\ c_{2,1} & c_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{v,1} & c_{v,2} & \cdots & c_{v,v} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,v} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_v \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_v \\ \vdots \\ b_m \end{bmatrix}, \tag{3.4}$$

with $b_i = w(\mathbf{a}_i)$, $i = 0, \cdots, m$.

Lemma 3.2 *System (3.1) has a solution* $\alpha_0, \alpha_1, \cdots, \alpha_n$ *satisfying* $\langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}_j, \hat{\alpha} \rangle = \alpha_0$ *if and only if System (3.4) has a solution* $x_1, \cdots, x_v$ *satisfying* $c_{i,1}x_1 + c_{i,2}x_2 + \cdots + c_{i,v}x_v = b_i$ *and* $c_{j,1}x_1 + c_{j,2}x_2 + \cdots + c_{j,v}x_v = b_j$.

Inequality system in (3.4) defines a polyhedron R in $\mathbb{R}^v$, and for an extreme point, or a vertex, of R, there are at least v active constraints. It follows from Lemma 3.1 and Lemma 3.2 that

Lemma 3.3 *If $\mathbf{x}_0$ is an extreme point of R and $J = \{i_1, \cdots, i_u\}$ is the indices of active constraints of the system at $\mathbf{x}_0$ with $u \geq v$, then $\{\hat{\mathbf{a}}_{i_k}, \hat{\mathbf{a}}_{i_l}\}$ is a lower edge of $\hat{\mathcal{B}}$ for any $i_k, i_l \in J$.*

On the other hand, if $\{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j\}$ is a lower edge of $\hat{\mathcal{B}}$, there is a lower facet of conv$(\hat{\mathcal{B}})$ with inner normal $\hat{\alpha} = (\alpha, 1)$ where $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$ which contains the line segment of $\{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j\}$. Let $\alpha_0 = \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}_j, \hat{\alpha} \rangle$ and $\mathbf{x}_0$ be the first v components $x_1, \cdots, x_v$ of $L^{-1}(\alpha_0, \alpha_1, \cdots, \alpha_n)$ in (3.3). Apparently, $\mathbf{x}_0$ is an extreme point of R.

Therefore, in order to find all the lower edges of $\hat{\mathcal{B}}$, it suffices to locate all the extreme points of the polyhedral R defined by the inequalities in (3.4). To reach this goal, our main strategy is to find an initial extreme point of R at the first step and generate all other extreme points of R from this extreme point thereafter.

To find an initial extreme point of R, we may first solve the triangular system

$$\begin{array}{ccccccccc}
c_{1,1}x_1 & & & & & & & = & b_1 \\
c_{2,1}x_1 & + & c_{2,2}x_2 & & & & & = & b_2 \\
\vdots & & & \ddots & & & & & \vdots \\
c_{v,1}x_1 & + & c_{v,2}x_2 & + & \cdots & + & c_{v,v}x_v & = & b_v
\end{array}$$

in (3.4). Let the solution be $\mathbf{x}_0 = (x_{01}, x_{02}, \cdots, x_{0v})$, and let

$$d_i = b_i - (c_{i,1}x_{01} + c_{i,2}x_{02} + \cdots + c_{i,v}x_{0v}) \quad \text{for} \quad i = v+1, \cdots, m.$$

If $d_l = \min\limits_{v+1 \leq i \leq m} d_i \geq 0$, then $\mathbf{x}_0$ is already an extreme point of R. Otherwise we apply Algorithm 1 to the following linear programming problem

$$
\begin{array}{rcccccccccl}
\min & \epsilon & & & & & & & & & \\
c_{1,1}x_1 & & & & & & & & & \leq & b_1 \\
c_{2,1}x_1 & + & c_{2,2}x_2 & & & & & & & \leq & b_2 \\
\vdots & & & \ddots & & & & & & & \vdots \\
c_{v,1}x_1 & + & c_{v,2}x_2 & + \cdots + & c_{v,v}x_v & & & & & \leq & b_v \\
c_{v+1,1}x_1 & + & c_{v+1,2}x_2 & + \cdots + & c_{v+1,v}x_v & - & c_{v+1,v+1}\epsilon & & & \leq & b_{v+1} \\
\vdots & & & & & & & & & & \vdots \\
c_{m,1}x_1 & + & c_{m,2}x_2 & + \cdots + & c_{m,v}x_v & - & c_{m,v+1}\epsilon & & & \leq & b_m \\
 & & & & & & -\epsilon & & & \leq & 0
\end{array} \tag{3.5}
$$

$$
\text{where } c_{i,v+1} = \begin{cases} 0, & \text{if } d_i \geq 0 \\ 1, & \text{if } d_i < 0 \end{cases}, \text{ for } v \leq i \leq m,
$$

in the variable $(x_1, \cdots, x_v, \epsilon)$ with initial extreme point $(\mathbf{x}, \epsilon) = (\mathbf{x}_0, -d_l)$ of (3.5) and initial indices of constraints $J = \{1, 2, \cdots, v, l\}$. The optimal solution of this problem gives an initial extreme point of R that we need.

Let $\mathbf{x}_0$ be an initial extreme point of R. To generate all other extreme points, we first introduce the following linear programming problem:

Two-Point Test Problem:

$$
\begin{array}{rl}
\min & -(c_{i_0,1}+c_{j_0,1})x_1 - (c_{i_0,2}+c_{j_0,2})x_2 - \cdots - (c_{i_0,v}+c_{j_0,v})x_v \\
& \begin{array}{rcccccl}
c_{1,1}x_1 & & & & & \leq & b_1 \\
c_{2,1}x_1 & + & c_{2,2}x_2 & & & \leq & b_2 \\
\vdots & & & & & & \vdots \\
c_{v,1}x_1 & + & c_{v,2}x_2 & + \cdots + & c_{v,v}x_v & \leq & b_v \\
\vdots & & & & & & \vdots \\
c_{m,1}x_1 & + & c_{m,2}x_2 & + \cdots + & c_{m,v}x_v & \leq & b_m
\end{array}
\end{array} \tag{3.6}
$$

where $1 \leq i_0, j_0 \leq m$.

Lemma 3.4 *Given $1 \leq i_0, j_0 \leq m$, if the optimal value of the problem in (3.6) is $-b_{i_0} - b_{j_0}$, then $\{\hat{\mathbf{a}}_{i_0}, \hat{\mathbf{a}}_{j_0}\}$ is an lower edge of $\hat{\mathcal{B}}$.*

PROOF: If the optimal value $-b_{i_0} - b_{j_0}$ is attained at $(x_0, \cdots, x_v)$, then

$$\begin{aligned}
&-(c_{i_0,1} + c_{j_0,1})x_1 - (c_{i_0,2} + c_{j_0,2})x_2 - \cdots - (c_{i_0,v} + c_{j_0,v})x_v \\
&= (-c_{i_0,1}x_1 - c_{i_0,2}x_2 - \cdots - c_{i_0,v}x_v) + (-c_{j_0,1}x_1 - c_{j_0,2}x_2 - \cdots - c_{j_0,v}x_v) \\
&= -b_{i_0} - b_{j_0}.
\end{aligned}$$

But $(x_0, \cdots, x_v)$ also satisfies constraints

$$\begin{aligned}
c_{i_0,1}x_1 + c_{i_0,2}x_2 + \cdots + c_{i_0,v}x_v &\leq b_{i_0}, \\
c_{j_0,1}x_1 + c_{j_0,2}x_2 + \cdots + c_{j_0,v}x_v &\leq b_{j_0}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
c_{i_0,1}x_1 + c_{i_0,2}x_2 + \cdots + c_{i_0,v}x_v &= b_{i_0}, \\
c_{j_0,1}x_1 + c_{j_0,2}x_2 + \cdots + c_{j_0,v}x_v &= b_{j_0}.
\end{aligned}$$

By Lemma 3.1 and Lemma 3.2, $\{\hat{\mathbf{a}}_{i_0}, \hat{\mathbf{a}}_{j_0}\}$ is a lower edge of $\hat{\mathcal{B}}$. □

The constraint in (3.6) is the same inequality system in (3.4) which defines polyhedron R. Since an initial extreme point $\mathbf{x}_0$ of R is available, we may use Algorithm 1 on the Two-Point Test Problem in (3.6) to test if $\{\hat{\mathbf{a}}_{i_0}, \hat{\mathbf{a}}_{j_0}\}$ for given $1 \leq i_0, j_0 \leq m$ is a lower edge of $\hat{\mathcal{B}}$. By using Algorithm 1, the optimal value of the problem is reached by moving from one extreme point of R to another extreme point of R in the direction where the objective function decreases. By Lemma 3.3, a newly obtained extreme point of R in the process provides a new collection of lower edges of $\hat{\mathcal{B}}$. This important feature keeps us away from exhaustive testings on all the possible pairs in $\hat{\mathcal{B}}$ for identifying all the lower edges of $\hat{\mathcal{B}}$.

The details of our algorithm for finding all lower edges of $\hat{\mathcal{B}}$ is given in the following

Algorithm 2 *Given $\hat{\mathcal{B}} = \{\hat{\mathbf{a}}_0, \hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_m\}$, construct $\mathcal{L}(\hat{\mathcal{B}})$.*

Step 0: ***Initialization.***

Set up inequality system (3.4). Let $\mathcal{P} = \{\{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j\} \big| 1 \le i, j \le m\}$ be all the possible pairs of $\hat{\mathcal{B}}$. If $v = m$, set $\mathcal{L}(\hat{\mathcal{B}}) := \mathcal{P}$ and stop. Otherwise find an initial extreme point $\mathbf{x}_0$ of system (3.4) with $J = \{i_1, \cdots, i_v\}$ and $D^{-1} = [\mathbf{u}_1, \cdots, \mathbf{u}_v]$, where $D^T = [\mathbf{c}_{i_1}, \cdots, \mathbf{c}_{i_v}]$ by applying Algorithm 1 to the optimization problem (3.5). Set $\mathcal{L}(\hat{\mathcal{B}}) = \emptyset$, go to step 1.

Step 1: ***Set up objective function for the Two-Point Test***

If $\mathcal{P} = \emptyset$, stop. Otherwise select $\{\hat{\mathbf{a}}_{i_0}, \hat{\mathbf{a}}_{j_0}\} \in \mathcal{P}$, set $\mathbf{f} := (-c_{i_0,1} - c_{j_0,1}, \cdots, -c_{i_0,v} - c_{j_0,v})$, and $\mathcal{P} := \mathcal{P} \backslash \{\{\hat{\mathbf{a}}_{i_0}, \hat{\mathbf{a}}_{j_0}\}\}$, go to Step 2.

Step 2: ***Apply Algorithm 1***

Determine the smallest index k such that

$$\langle \mathbf{f}, \mathbf{u}_k \rangle = \max\{\langle \mathbf{f}, \mathbf{u}_i \rangle \mid i = 1, \cdots, v\}$$

If $\langle \mathbf{f}, \mathbf{u}_k \rangle \le 0$, go to Step 1. Otherwise, set $\mathbf{s} = \mathbf{u}_k$ and go to Step 3.

Step 3: *Compute the smallest index l and σ such that*

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

Go to Step 4.

Step 4: *Set $\mathbf{x}_0 := \mathbf{x}_0 - \sigma \mathbf{s}$ and update $J = \{i_1, \cdots, i_v\}$ and D^{-1}. Set $\mathcal{L}(\hat{\mathcal{B}}) := \mathcal{L}(\hat{\mathcal{B}}) \cup (\mathcal{P} \cap \{\{\hat{\mathbf{a}}_k, \hat{\mathbf{a}}_l\} | k, l \in J\})$, and $\mathcal{P} := \mathcal{P} \backslash \{\{\hat{\mathbf{a}}_k, \hat{\mathbf{a}}_l\} | k, l \in J\}$ Go to Step 2.*

3.2 Extend level-k Subfaces

For a level-k subface $\hat{E}_k = (\hat{e}_1, \cdots, \hat{e}_k)$ of $\hat{S} = (\hat{S}_1, \cdots, \hat{S}_n)$ with $1 \le k < n$ where $\hat{e}_i = \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}'_i\} \in \mathcal{L}(\hat{S}_i)$ for $i = 1, \cdots, k$, we say $\hat{e}_{k+1} = \{\hat{\mathbf{a}}_{k+1}, \hat{\mathbf{a}}'_{k+1}\} \in \mathcal{L}(\hat{S}_{k+1})$ *extends*

$\hat{E}_k$ if $\hat{E}_{k+1} = (\hat{e}_1, \cdots, \hat{e}_{k+1})$ is a level-$(k+1)$ subface of $\hat{S}$. Let

$$\mathcal{E}(\hat{E}_k) = \left\{ \{\hat{\mathbf{a}}_{k+1}, \hat{\mathbf{a}}'_{k+1}\} \in \mathcal{L}(\hat{S}_{k+1}) \middle| \{\hat{\mathbf{a}}_{k+1}, \hat{\mathbf{a}}'_{k+1}\} \text{ extends } \hat{E}_k \right\}.$$

$\hat{E}_k$ is called *extendible* if $\mathcal{E}(\hat{E}_k) \neq \emptyset$, it is *nonextendible* otherwise. To find all mixed cells of $S = (S_1, \cdots, S_n)$, we will start from $k = 1$ and extend $\hat{E}_k$ step by step. If $\hat{E}_k$ is nonextendible, there is no mixed cell of S which contains $(\{\mathbf{a}_1, \mathbf{a}'_1\}, \cdots, \{\mathbf{a}_k, \mathbf{a}'_k\})$, and extension attempt will be repeated on the next $\hat{E}_k$. Obviously, when $k = n-1$, an extendible $\hat{E}_k$ yields mixed cells of S with elements in $\mathcal{E}(\hat{E}_k)$ (possibly several).

In this section, we describe our algorithm to calculate $\mathcal{E}(\hat{E}_k)$ efficiently for a given level-k subface $\hat{E}_k = (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_k)$ where $\hat{\mathbf{e}}_i = \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}'_i\} \subset \hat{S}_i$ for $i = 1, \cdots, k$.

Now, consider the following system in the $n+1$ unknowns $\alpha_0, \alpha_1, \cdots, \alpha_n$

$$\begin{aligned} \langle \hat{\mathbf{a}}, \hat{\alpha} \rangle &\geq \alpha_0, & \hat{\mathbf{a}} \in \hat{S}_{k+1} \\ \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle &\leq \langle \hat{\mathbf{a}}, \hat{\alpha} \rangle, & \hat{\mathbf{a}} \in \hat{S}_i, \text{ for } i = 1, \cdots, k \\ \langle \hat{\mathbf{a}}_i, \hat{\alpha} \rangle &= \langle \hat{\mathbf{a}}'_i, \hat{\alpha} \rangle, & i = 1, \cdots, k, \end{aligned} \tag{3.7}$$

where $\hat{\alpha} = (\alpha_1, \cdots, \alpha_n, 1) \in \mathbb{R}^{n+1}$.

The following lemma is obvious.

Lemma 3.5 *For a level-k subface $\hat{E}_k = (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_k)$, where $\hat{\mathbf{e}}_i = \{\hat{\mathbf{a}}_i, \hat{\mathbf{a}}'_i\} \subset \hat{S}_i$, if system (3.7) has a solution $(\alpha_0, \alpha_1, \cdots, \alpha_n)$ such that $\langle \hat{\mathbf{a}}_l, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}_j, \hat{\alpha} \rangle = \alpha_0$ for $\hat{\mathbf{a}}_l, \hat{\mathbf{a}}_j \in \hat{S}_{k+1}$, then $\{\hat{\mathbf{a}}_l, \hat{\mathbf{a}}_j\}$ extends $\hat{E}_k$.*

With $S_i = \{a_{i,1}, \cdots, a_{i,m_i}\}$ for $i = 1, \cdots, n$ and $\alpha = (\alpha_1, \cdots, \alpha_n)$ we may rewrite system (3.7) as

$$\begin{aligned} \langle \mathbf{a}_{k+1,j}, \alpha \rangle - \alpha_0 &\leq -w_{k+1}(\mathbf{a}_{k+1,j}) & & j = 1, \cdots, m_{k+1} \\ \langle \mathbf{a}_i - \mathbf{a}_{i,j}, \alpha \rangle &\leq w_i(\mathbf{a}_{i,j}) - w_i(\mathbf{a}_i) & & j = 1, \cdots, m_i,\ \mathbf{a}_{i,j} \in S_i \backslash \{\mathbf{a}_i, \mathbf{a}'_i\},\ i = 1, \cdots, k \\ \langle \mathbf{a}_i - \mathbf{a}'_i, \alpha \rangle &= w_i(\mathbf{a}'_i) - w_i(\mathbf{a}_i) & & i = 1, \cdots, k. \end{aligned}$$

By using the last k equality constraints to eliminate k variables of α, the above system can be reduced to the following general inequality system :

$$
\begin{array}{cccccccc}
c'_{1,j_1}\alpha_{j_1} & + & c'_{1,j_2}\alpha_{j_2} & + \cdots + & c'_{1,j_{\eta'}}\alpha_{j_{\eta'}} & \leq & b_1 \\
c'_{2,j_1}\alpha_{j_1} & + & c'_{2,j_2}\alpha_{j_2} & + \cdots + & c'_{2,j_{\eta'}}\alpha_{j_{\eta'}} & \leq & b_2 \\
\vdots & & & & & & \vdots \\
c'_{\mu,j_1}\alpha_{j_1} & + & c'_{\mu,j_2}\alpha_{j_2} & + \cdots + & c'_{\mu,j_{\eta'}}\alpha_{j_{\eta'}} & \leq & b_\mu
\end{array}
\tag{3.8}
$$

where $\mu = \sum_{i=1}^{k+1} m_i - 2k$ and $\eta' = n - k + 1$.

As before, by a coordinate transformation $(x_1, \cdots, x_{\eta'}) = (\alpha_{j_1}, \cdots, \alpha_{j_{\eta'}})L$, where $L \in \mathbb{R}^{\eta' \times \eta'}$ is nonsingular, the system can be further reduced to the following inequality system:

$$
\begin{array}{cccccccc}
c_{1,1}x_1 & & & & & \leq & b_1 \\
c_{2,1}x_1 & + & c_{2,2}x_2 & & & \leq & b_2 \\
\vdots & & & \ddots & & & \vdots \\
c_{\eta,1}x_1 & + & c_{\eta,2}x_2 & + \cdots + & c_{\eta,\eta}x_\eta & \leq & b_\eta \\
\vdots & & & & & & \vdots \\
c_{\mu,1}x_1 & + & c_{\mu,2}x_2 & + \cdots + & c_{\mu,\eta}x_\eta & \leq & b_\mu .
\end{array}
\tag{3.9}
$$

Lemma 3.6 *System (3.9) has a solution* $(\alpha_0, \alpha_1, \cdots, \alpha_n)$ *satisfying* $\langle \hat{\mathbf{a}}_{k+1.i}, \hat{\alpha} \rangle = \langle \hat{\mathbf{a}}_{k+1,j}, \hat{\alpha} \rangle = \alpha_0$ *for* $i \leq i, j \leq m_{k+1}$ *if and only if system (3.9) has a solution* $x_1, \cdots, x_\eta$ *such that* $c_{i,1}x_1 + c_{i,2}x_2 + \cdots + c_{i,\eta}x_\eta = b_i$ *and* $c_{j,1}x_1 + c_{j,2}x_2 + \cdots + c_{j,\eta}x_\eta = b_j$.

Inequalities in (3.9) defines a polyhedron $\bar{R}$ in $\mathbb{R}^\eta$, and for an extreme point, or a vertex, of $\bar{R}$, there are at least η active constraints. From Lemma 3.5 and Lemma 3.6, it follows that

Lemma 3.7 *If* $\mathbf{x}_0$ *is an extreme point of* $\bar{R}$ *and* $J = \{i_1, \cdots, i_t\}$ *is the indices of active constraints of the system at* $\mathbf{x}_0$ *with* $t \geq \eta$, *then* $\{\hat{\mathbf{a}}_{k+1,i_p}, \hat{\mathbf{a}}_{k+1,i_q}\}$ *extends* $\hat{E}_k$ *for any* $i_p, i_q \in J \cap \{1, \cdots, m_{k+1}\}$.

Similar to the discussion following Lemma 3.3, if $\{\hat{\mathbf{a}}_{k+1,i}, \hat{\mathbf{a}}_{k+1,j}\} \in \mathcal{L}(\hat{S}_{k+1})$ for $\{i,j\} \subset I \equiv \{1, \cdots, m_{k+1}\}$, it will lead to a corresponding point $\mathbf{x}_0$ of $\bar{R}$, its indices of active constraints includes $\{i,j\}$. Hence, to construct $\mathcal{E}(\hat{E}_k) \subset \mathcal{L}(\hat{S}_{k+1})$, we may look for all those extreme points of $\bar{R}$ whose indices of active constraints contain at least a pair of $\{i,j\}$ in I. To achieve this, we may certainly apply the Two-Point Test introduced in the last section and confine to I the indices of the "two points" to be tested. However, it is very likely that most of the $\hat{\mathbf{a}}_{k+1,i}$'s that appeare in the pairs in $\mathcal{L}(\hat{S}_{k+1})$ fail to extend $\hat{E}_k$ with their associated pairs in $\mathcal{L}(\hat{S}_{k+1})$. Namely, those points do not exist in any of the pairs in $\mathcal{E}(\hat{E}_k)$. This phenomenon never occurs when we compute the set of lower edges $\mathcal{L}(\hat{\mathcal{B}})$ of $\hat{\mathcal{B}}$ in the last section since all the points in $\mathcal{B}$ are extreme points. Consequently, every point of $\hat{\mathcal{B}}$ appears in certain pairs of $\mathcal{L}(\hat{\mathcal{B}})$. From this important observation, we introduce the following One-Point Test to be used in additional to the Two-Point Test in our algorithm.

One-Point Test Problem:

$$\begin{array}{llllllll}
min & -c_{i_0,1}x_1 & - & c_{i_0,2}x_2 & - \cdots - & c_{i_0,\eta}x_\eta & & \\
 & c_{1,1}x_1 & & & & & \leq & b_1 \\
 & c_{2,1}x_1 & + & c_{2,2}x_2 & & & \leq & b_2 \\
 & & \vdots & & \ddots & & & \vdots \\
 & c_{\eta,1}x_1 & + & c_{\eta,2}x_2 & + \cdots + & c_{\eta,\eta}x_\eta & \leq & b_\eta \\
 & & \vdots & & & & & \vdots \\
 & c_{\mu,1}x_1 & + & c_{\mu,2}x_2 & + \cdots + & c_{\mu,\eta}x_\eta & \leq & b_\mu
\end{array} \qquad (3.10)$$

where $1 \leq i_0 < m_{k+1}$.

Lemma 3.8 *Given $1 \leq i_0 < m_{k+1}$, if the optimal value of system (3.10) is greater than $-b_{i_0}$, then $\{\hat{\mathbf{a}}_{k+1,i_0}, \hat{\mathbf{a}}_{k+1,i}\}$ does not extend $\hat{E}_k$ for all $i \in \{1, \cdots, m_{k+1}\} \backslash \{i_0\}$.*

PROOF: Suppose there exists $1 \leq j_0 \leq m_{k+1}$ for which $\{\hat{\mathbf{a}}_{k+1,i_0}, \hat{\mathbf{a}}_{k+1,j_0}\}$ extends $\hat{E}_k$.

By Lemmas 3.5 and 3.6, system (3.9) has a solution $(x_1, \cdots, x_\eta)$ satisfying

$$c_{i_0,1}x_1 + c_{i_0,2}x_2 \cdots + c_{i_0,\eta}x_\eta = b_{i_0},$$
$$c_{j_0,1}x_1 + c_{j_0,2}x_2 \cdots + c_{j_0,\eta}x_\eta = b_{j_0}.$$

Hence the objective function value at $(x_1, \cdots, x_\eta)$ is

$$-c_{i_0,1}x_1 - c_{i_0,2}x_2 - \cdots - c_{i_0,\eta}x_\eta = -b_{i_0},$$

which contradicts the fact that the optimal value of the system (3.10) is greater than $-b_{i_0}$. □

From the above lemma, points appeared in the pairs in $\mathcal{L}(\hat{S}_{k+1})$ may be tested systematically by using One-Point Test to check the possibilities of their appearances in the pairs in $\mathcal{E}(\hat{E}_k)$. When the optimal value obtained is not as desired for a particular point $\hat{\mathbf{a}}_{k+1,i_0}$, all the pairs associated with $\hat{\mathbf{a}}_{k+1,i_0}$ in $\mathcal{L}(\hat{S}_{k+1})$ should be deleted from further considerations. In the meantime, in the process of reaching the optimal value of the problem, newly obtained extreme points of $\bar{R}$ provide a collection of new pairs of $\mathcal{E}(\hat{E}_k)$ as long as their active constraints contain a pair of $\{i, j\}$ in $I = \{1, \cdots, m_{k+1}\}$. Furthermore, we no longer test points $\hat{\mathbf{a}}_{k+1,i}$ in $\mathcal{L}(\hat{S}_{k+1})$ whose index i have appeared in any of the indices of the active constraints of the extreme points of $\bar{R}$ being obtained.

The system of constraints in problem (3.10) is the same inequality system in (3.9) which defines the polyhedron $\bar{R}$ in $\mathbb{R}^\eta$. To find an initial extreme point of $\bar{R}$ to start Algorithm 1 on the problem, we may employ the same strategy by augmenting a new variable $\epsilon \geq 0$ as in calculating $\mathcal{L}(\hat{\mathcal{B}})$ of $\hat{\mathcal{B}}$ in the last section.

Two-Point Tests will be used only after One-Point Tests have exhausted all the testings on possible candidates. Our experiences show that the Two-Point Test only plays a minor role in constructing $\mathcal{E}(\hat{E}_k)$, namely, when we finish the One-Point Tests, most of the pairs in $\mathcal{E}(\hat{E}_k)$ have been found.

Combining the One-Point Test and the Two-Point Test, we list the following algorithm for constructing $\mathcal{E}(\hat{E}_k)$.

Algorithm 3 *Given $\hat{E}_k$, construct $\mathcal{E}(\hat{E}_k)$.*

Step 0: ***Initialization.***

Set up the inequality system (3.9). Start from an extreme point $\mathbf{x}_0$ with $J = \{i_1, \cdots, i_\eta\}$ and $D^{-1} = [\mathbf{u}_1, \cdots, \mathbf{u}_\eta]$, where $D^T = [\mathbf{c}_{i_1}, \cdots, \mathbf{c}_{i_\eta}]$, and set $\hat{F}_{k+1} := \mathcal{L}(\hat{S}_{k+1})$.

Step 1: ***One-Point Test Problems.***

Step 1.0 *Set $i_0 := 0$, go to Step 1.1.*

Step 1.1 *Set up objective function*

Find $\tau = \min\left\{j \;\middle|\; j > i_0 \text{ and } \{\hat{\mathbf{a}}_{k+1,j}, \hat{\mathbf{a}}_{k+1,j'}\} \subset \hat{F}_{k+1} \text{ for some } j'\right\}$. If such τ does not exists, go to Step 2. Otherwise set $i_0 := \tau$ and $\mathbf{f} = (-c_{i_0,1}, \cdots, -c_{i_0,\eta})$, go to Step 1.2.

Step 1.2 *Determine the smallest index k such that*

$$\langle \mathbf{f}, \mathbf{u}_k \rangle = \max\{\langle \mathbf{f}, \mathbf{u}_i \rangle \mid i = 1, \cdots, \eta\}.$$

If $\langle \mathbf{f}, \mathbf{u}_k \rangle \leq 0$, go to Step 1.5. Otherwise, set $\mathbf{s} = \mathbf{u}_k$ and go to Step 1.3.

Step 1.3 *Compute the smallest index l and σ such that*

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

Go to Step 1.4.

Step 1.4 *Set $\mathbf{x}_0 := \mathbf{x}_0 - \sigma\mathbf{s}$ and update $J = \{i_1, \cdots, i_\eta\}$ and D^{-1}. If $l < m_{k+1}$, check if any lower edge $\{\hat{\mathbf{a}}_{k+1,l}, \hat{\mathbf{a}}_{k+1,j}\}$ in $\hat{F}_{k+1}$*

extends $\hat{F}_{k+1}$. Collect these lower edges, if they exist, and delete them from $\hat{F}_{k+1}$.

Go to Step 1.2.

Step 1.5 *If the current value of objective function is not equal to $-b_{i_0}$, delete all lower edges containing point $\hat{\mathbf{a}}_{k+1,i_0}$ from $\hat{F}_{k+1}$.*

Go to Step 1.1.

Step 2: ***Two-point Test Problems.***

Step 2.1 *Set up objective function.*

If $\hat{F}_{k+1} = \emptyset$, stop. Otherwise select a lower edge $\{\hat{\mathbf{a}}_{k+1,i_0}, \hat{\mathbf{a}}_{k+1,j_0}\} \in \hat{F}_{k+1}$. Set $\mathbf{f} := (-c_{i_0,1} - c_{j_0,1}, \cdots, -c_{i_0,\eta} - c_{j_0,\eta})$, and $\hat{F}_{k+1} := \hat{F}_{k+1} \backslash \{\hat{\mathbf{a}}_{k+1,i_0}, \hat{\mathbf{a}}_{k+1,j_0}\}$, go to Step 2.2.

Step 2.2 *Determine the smallest index k such that*

$$\langle \mathbf{f}, \mathbf{u}_k \rangle = \max\{\langle \mathbf{f}, \mathbf{u}_i \rangle \mid i = 1, \cdots, \eta\}.$$

If $\langle \mathbf{f}, \mathbf{u}_k \rangle \leq 0$, go to Step 2.1. Otherwise, set $\mathbf{s} = \mathbf{u}_k$ and go to Step 2.3.

Step 2.3 *Compute the smallest index l and σ such that*

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

Go to Step 2.4.

Step 2.4 *Set $\mathbf{x}_0 := \mathbf{x}_0 - \sigma \mathbf{s}$ and update $J = \{i_1, \cdots, i_\eta\}$ and D^{-1}. If $l < m_{k+1}$, check if any lower edge $\{\hat{\mathbf{a}}_{k+1,l}, \hat{\mathbf{a}}_{k+1,j}\}$ in $\hat{F}_{k+1}$ extends $\hat{F}_{k+1}$. Collect those lower edges, if they exist, and delete them from $\hat{F}_{k+1}$.*

Go to Step 2.2.

Remark 1 *Numerical testing shows that setting up the inequality system (3.9) is very time consuming. One strategy we employ is to save the inequality systems at all previous levels. Thus the inequality system in the current level can be set up by using the inequality system that already exist.*

3.3 Find All Fine Mixed Cells

For $S = (S_1, \cdots, S_n)$ with $S_i = \{\mathbf{a}_{i,1}, \cdots, a_{i,m_i}\} \subset \mathbb{N}^n$, $i = 1, \cdots, n$ and generically chosen $w = (w_1, \cdots, w_n)$ with

$$\hat{S}_i = \left\{ \hat{\mathbf{a}} = (\mathbf{a}, w_i(\mathbf{a})) \;\middle|\; \mathbf{a} \in S_i \right\}, \quad i = 1, \cdots, n,$$

we combine our algorithms described in the last two sections in the following algorithm for finding all the mixed cells in $\hat{S} = (\hat{S}_1, \cdots, \hat{S}_n)$.

Algorithm 4 *Find all mixed cells in $S = (S_1, \cdots, S_n)$.*

Step 0: ***Initialization***.

Find $\mathcal{L}(\hat{S}_i)$, for all $i = 1, \cdots, n$.

Set $\mathcal{F}_1 := \mathcal{L}(\hat{S}_1)$, $k := 1$.

Step 1: ***Backtracking***.

If $k = 0$ Stop.

If $\hat{\mathcal{F}}_k = \emptyset$, set $k := k - 1$ and go to Step 1.

Otherwise go to Step 2.

Step 2: ***Select next level-k subface to extend***.

Select $\hat{\mathbf{e}}_k \in \hat{\mathcal{F}}_k$, and set $\hat{\mathcal{F}}_k := \hat{\mathcal{F}}_k \backslash \{\hat{\mathbf{e}}_k\}$.

Let $\hat{E}_k = (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_k)$ and go to Step 3.

Step 3: ***Extending the level-k subface***.

Find $\mathcal{E}(\hat{E}_k)$.

If $\mathcal{E}(\hat{E}_k) = \emptyset$, go to Step 1, otherwise set $\hat{\mathcal{F}}_{k+1} = \mathcal{E}(\hat{E}_k)$, $k := k+1$ then go to Step 4.

Step 4: ***Collect mixed cells.***

If $k = n$, all $C = (\mathbf{e}_1, \cdots, \mathbf{e}_{n-1}, \mathbf{e}), \hat{\mathbf{e}} \in \hat{\mathcal{F}}_n$ are fine mixed cells, pick up all these mixed cells, then set $k := k-1$, go to Step 1.

Otherwise go to Step 2.

Remark 2 *In finding $\mathcal{E}(\hat{E}_k)$ at Step 3, inequalities associated with the points in $\hat{S}_i$ which never appear in $\mathcal{E}(\hat{E}_i)$ for $i = 1, \cdots, k-1$ should not be considered as constraints, since these points will never enter the level-k subface.*

CHAPTER 4

Numerical Implementation

4.1 Algorithm

For finding all isolated zeros of a polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_n(\mathbf{x}))$ in $\mathbb{C}^n$, where

$$p_i(\mathbf{x}) = \sum_{\mathbf{a} \in S_i} c_{i,\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \quad \text{for } i = 1, \cdots, n,$$

we outline the major steps in brief terms as follows:

(A) Set up the polyhedral homotopy $Q(\mathbf{x}, t) : \mathbb{C}^n \times [0,1] \to \mathbb{C}^n$ as

$$q_i(\mathbf{x}, t) = \sum_{\mathbf{a} \in S_i \cup \{0\}} (c_{i,\mathbf{a}} + (1-t)\epsilon_{i,\mathbf{a}}) t^{w_i(\mathbf{a})} \mathbf{x}^{\mathbf{a}}, \quad \text{for } i = 1, \cdots, n,$$

where $w_i : S_i \cup \{0\} \to \mathbb{R}$ are chosen generically and $\epsilon_{i,\mathbf{a}}$ are randomly chosen complex numbers.

(B) Find all mixed cells of extended support $S_1 \cup \{0\}, \cdots, S_n \cup \{0\}$. For each inner normal α associated with a mixed cell, define the homotopy

$$H_\alpha(\mathbf{y}, t) \equiv t^{-\beta} Q(\mathbf{y} t^{\alpha}, t) = 0,$$

where $\beta = (\beta_1, \cdots, \beta_n)$ and β_i is the lowest order in t among all the terms in $q_i(\mathbf{y}t^{\alpha}, t)$.

(C) Solve the binomial system $H_\alpha(\mathbf{y},0)=0$ in $\mathbb{C}^n$, then follow homotopy paths of $H_\alpha(\mathbf{y},t)=0$ to find all the isolated zeros of $P(\mathbf{x})$.

We set up our initial system $\bar{Q}(\mathbf{x})=(\bar{q}_1(\mathbf{x}),\cdots,\bar{q}_n(\mathbf{x}))$ by perturbating the coefficients of $P(\mathbf{x})$, that is

$$\bar{q}_i(\mathbf{x})=\sum_{\mathbf{a}\in S_i\cup\{0\}}(c_{i,\mathbf{a}}+\epsilon_{i,\mathbf{a}})\mathbf{x}^{\mathbf{a}},\quad i=1,\cdots,n$$

where $\epsilon_{i,\mathbf{a}}\in\mathbb{R}^n$ are randomly chosen small complex numbers, and let

$$\tilde{Q}(\mathbf{x},t)=(1-t)\bar{Q}(\mathbf{x})+tP(\mathbf{x}).$$

The homotopy $Q(\mathbf{x},t)$ in (A) is obtained by setting up the polyhedral homotopy for $\tilde{Q}(\mathbf{x},t)$ instead with the same variable t.

Apparently, we have $Q(\mathbf{x},1)=P(\mathbf{x})$ and for each $t\in[0,1]$, $Q(\mathbf{x},t)$ and $\bar{Q}(\mathbf{x})$ have the same support. Let

$$\hat{q}_i(\mathbf{x},t)=\sum_{\mathbf{a}\in S_i\cup\{0\}}(c_{i,\mathbf{a}}+\epsilon_{i,\mathbf{a}})\mathbf{x}^{\mathbf{a}}t^{w_i(\mathbf{a})},\ i=1,\cdots,n.$$

It is clear that, for any $\alpha\in\mathbb{R}^n$, the lowest order terms in t of both $q_j(\mathbf{x}t^\alpha,t)$ and $\hat{q}_j(\mathbf{y}t^\alpha,t)$ are the same. Hence the mixed cells and their associated inner normals stay invariant, and the start system of the homotopy

$$H_\alpha(y,t)\equiv t^{-\beta}Q(\mathbf{y}t^\alpha,t)=0 \tag{4.1}$$

is the same as that of the $\hat{H}_\alpha(y,t)\equiv t^{-\beta}\hat{Q}(\mathbf{y}t^\alpha,t)=0$ with $\hat{Q}(\mathbf{y}t^\alpha,t)=(\hat{q}_1(\mathbf{y}t^\alpha,t),\cdots,\hat{q}_n(\mathbf{y}t^\alpha,t))$. Here, again, $\beta=(\beta_1,\cdots,\beta_n)$ and for $j=1,\cdots,n$, β_j is the lowest order in t among all the terms in $q_j(\mathbf{y}t^\alpha,t)$. Thus, when nonsingular solutions of $H_\alpha(\mathbf{y},t)$ in $(\mathbb{C}^*)^n$ are available, we may follow those homotopy paths of $H_\alpha(\mathbf{y},t)$ in (4.1) instead with those starting points.

4.2 Implementation

In this section, we will briefly describe our software environment that has been developed.

Currently there are several publically available software packages dedicated to solving polynomial systems by homotopy continuations. HOMPACK [45] and CONSOL [31] are written in FORTRAN77, pss [28] and Pelican [16] are written in C and PHC [42] is written in Ada. Some of these software packages are integrated multi-purpose packages.

Our package is designed as a high-performance polynomial system solver, focused on better efficiency, portability and simplicity. Our program is written in C++, a standard programming language which provides excellent support for object-oriented programming, abstraction, and encapsulation.

The following UML diagrams illustrate the structure of our polynomial system solver system. Diagram 4.2 shows that our polynomial system solver relys on two packages. One of them provides utility tools such as linear programming and linear system solvers. The other one may not be critical, it is mainly used for parsing polynomial expressions and performing simple polynomial manipulation to provide certain degree of user friendly interface. The class diagram 4.2 shows our PolynomialSolver uses four major components: PolyhedralHomotopy, MixedCell, BinomialSystemSolver, and Continuation (which uses Newton's iteration). The state diagram 4.2 shows the transitions among states in the execution of our program.

4.3 Numerical Results

Our software package has solved many well-known polynomial systems successfully. In this section we present some of our numerical results.

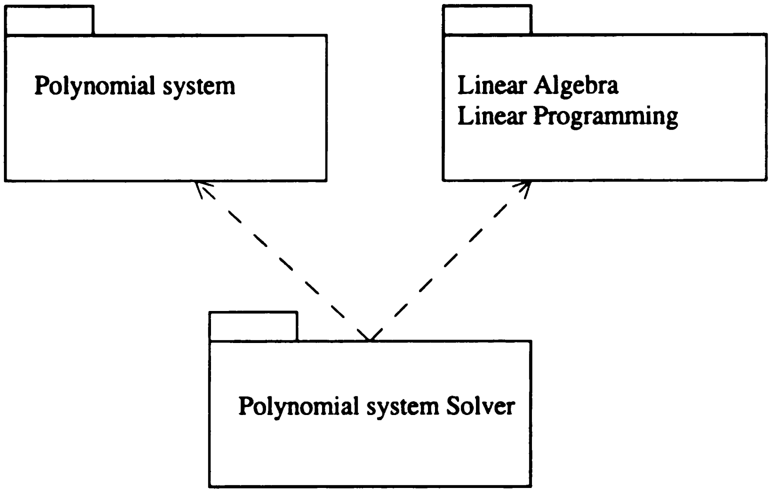

Figure 4.1: Package Diagram

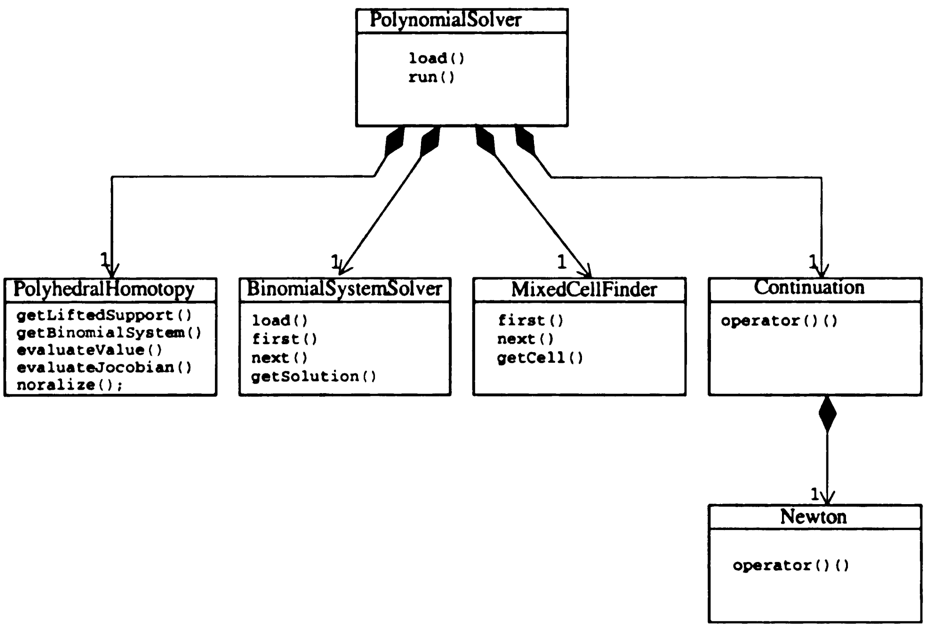

Figure 4.2: Class Diagram

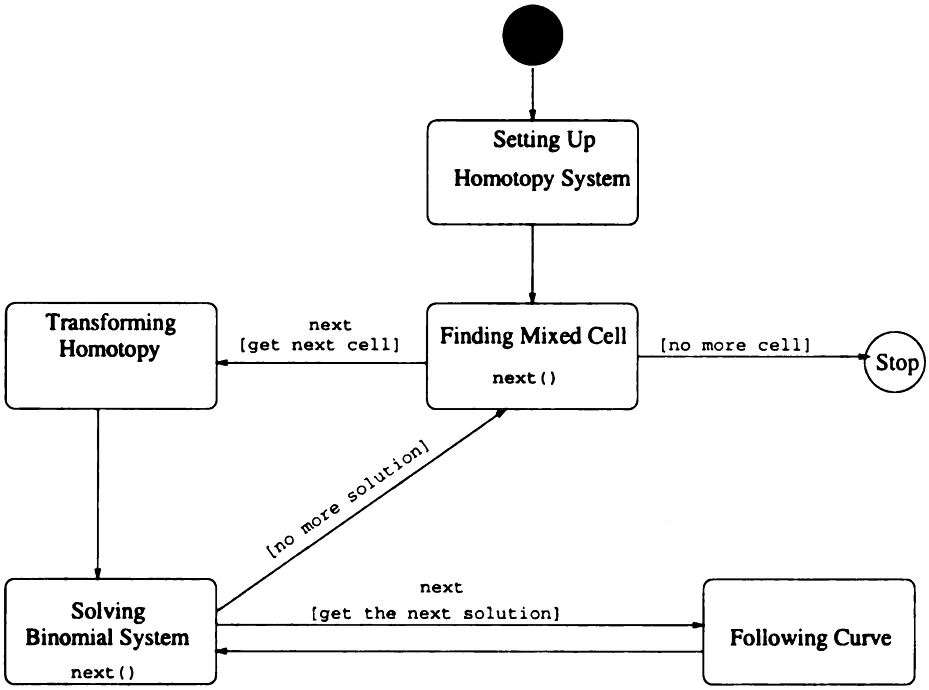

Figure 4.3: State Diagram

1. **System of Trinks from the PoSSo test suite**[40]

$$P(x,y,z,u,v,t) = \begin{cases} 45y + 35u - 165v - 36, \\ 35y + 25z + 40t - 27u, \\ 25yu - 165v^2 + 15x - 18z + 30t, \\ 15yz + 20tu - 9x, \\ -11v^3 + xy + 2zt, \\ -11uv + 3v^2 + 99x, \end{cases}$$

2. **Generalized eigenvalue problem**[9]

$$P(\mathbf{x}) = \begin{cases} -10x_1x_6^2 + 2x_2x_6^2 - x_3x_6^2 + x_4x_6^2 + 3x_5x_6^2 + x_1x_6 + 2x_2x_6 + x_3x_6 \\ +2x_4x_6 + x_5x_6 + 10x_1 + 2x_2 - x_3 + 2x_4 - 2x_5, \\ 2x_1x_6^2 - 11x_2x_6^2 + 2x_3x_6^2 - 2x_4x_6^2 + x_5x_6^2 + 2x_1x_6 + x_2x_6 + 2x_3x_6 \\ +x_4x_6 + 3x_5x_6 + 2x_1 + 9x_2 + 3x_3 - x_4 - 2x_5, \\ -x_1x_6^2 + 2x_2x_6^2 - 12x_3x_6^2 - x_4x_6^2 + x_5x_6^2 + x_1x_6 + 2x_2x_6 - 2x_4x_6 \\ -2x_5x_6 - x_1 + 3x_2 + 10x_3 + 2x_4 - x_5, \\ x_1x_6^2 - 2x_2x_6^2 - x_3x_6^2 - 10x_4x_6^2 + 2x_5x_6^2 + 2x_1x_6 + x_2x_6 - 2x_3x_6 \\ +2x_4x_6 + 3x_5x_6 + 2x_1 - x_2 + 2x_3 + 12x_4 + x_5, \\ 3x_1x_6^2 + x_2x_6^2 + x_3x_6^2 + 2x_4x_6^2 - 11x_5x_6^2 + x_1x_6 + 3x_2x_6 - 2x_3x_6 \\ +3x_4x_6 + 3x_5x_6 - 2x_1 - 2x_2 - x_3 + x_4 + 10x_5, \\ x_1 + x_2 + x_3 + x_4 + x_5 - 1, \end{cases}$$

3. **A system from economics modeling** [31]

The following formulates the system for general dimension n.

$$P(\mathbf{x}) = \begin{cases} \left(x_k + \sum_{l=1}^{n-k-1} x_l x_{l+k}\right) x_n - c_k, \; k = 1, \cdots, n-1. \\ \sum_{l=1}^{n-1} x_l + 1. \end{cases}$$

The constant c_k can be chosen at random. This problem has solved for dimension up 12.

4. **Totally mixed Nash equilibria for 5 players with two pure strategies**[29]

$$P(\mathbf{p}) = \begin{cases} 1.390350657p_3 + 0.4641393136p_3p_2 - 0.6266605268p_2 - 1.400171891p_4 \\ -2.090683800p_4p_3p_2 + 4.089263882p_4p_3 + 1.129827638p_4p_2 \\ +1.881614464p_5p_4p_3p_2 + 0.4716169661p_5p_3p_2 - 0.8625849122p_5p_4p_2 \\ -1.398871056p_5p_2 + 0.9599693844p_5 + 0.0714025397p_5p_4 \\ +0.1073802376p_5p_3 - 0.9259664538p_5p_4p_3 - 0.2067814278, \\ -1.196136754p_3 - 0.9249804195p_4 + 0.3188761009p_4p_3 - 1.045301323p_3p_1 \\ -0.0306661782p_1 + 0.5987012929p_4p_3p_1 - 0.4448182692p_4p_1 \\ -0.3908068031p_5p_4p_3p_1 - 1.212939725p_5 + 2.586129779p_5p_4 \\ -0.1180169224p_5p_3 - 1.051519507p_5p_4p_3 + 2.134979375p_5p_3p_1 \\ -1.337061849p_5p_4p_1 - 0.2961272671p_5p_1 + 0.7316111016, \\ 2.272943163p_2 - 0.4131564265p_4 - 1.920446680p_4p_2 + 0.0080509234p_1 \\ +1.342851102p_4p_1 - 2.979502184p_2p_1 + 3.391571834p_4p_2p_1 \\ -0.5975693742p_5p_4p_2 + 0.3002794716p_5p_2 - 0.7893445350p_5 \\ +1.276948001p_5p_4 - 4.601376311p_5p_4p_1 + 2.356804322p_5p_1 \\ +3.498840190p_5p_4p_2p_1 - 1.355375015p_5p_2p_1 - 1.231070236, \\ -2.206336116p_3 + 2.318673689p_3p_2 - 1.267478048p_2 + 1.110654516p_3p_1 \\ +1.533592098p_1 - 1.872504375p_2p_1 + 0.3299103675p_3p_2p_1 \\ -3.400750472p_5p_3p_2 + 2.093674516p_5p_2 - 1.772874182p_5 \\ +2.993821915p_5p_3 - 1.356762392p_5p_3p_1 + 0.0637534233p_5p_1 \\ +0.5870371377p_5p_2p_1 + 1.018269743p_5p_3p_2p_1 + 1.400431557, \\ -2.522718869p_3 + 0.8323646978p_3p_2 - 1.375039881p_2 - 0.3055443755p_4 \\ +0.6760632172p_4p_3p_2 - 0.4262974456p_4p_3 + 1.268255245p_3p_1 \\ +0.5352674901p_4p_2 - 1.024495558p_1 + 1.818275404p_4p_3p_1 \\ -1.354832512p_4p_1 - 1.595112039p_2p_1 + 2.237956242p_4p_2p_1 \\ +3.370102170p_3p_2p_1 - 3.465040669p_4p_3p_2p_1 + 2.132631128, \end{cases}$$

5. **Benchmark i1 from the Interval Arithmetics Benchmarks** [15]

$$P(\mathbf{x}) = \begin{cases} x_1 - 0.25428722 - 0.18324757x_4x_3x_9, \\ x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6, \\ x_3 - 0.27162577 - 0.16955071x_1x_2x_{10}, \\ x_4 - 0.19807914 - 0.15585316x_7x_1x_6, \\ x_5 - 0.44166728 - 0.19950920x_7x_6x_3, \\ x_6 - 0.14654113 - 0.18922793x_8x_5x_{10}, \\ x_7 - 0.42937161 - 0.21180484x_2x_5x_8, \\ x_8 - 0.07056438 - 0.17081208x_1x_7x_6, \\ x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8, \\ x_{10} - 0.42651102 - 0.21466544x_4x_8x_1, \end{cases}$$

This system is very sparse, the mixed volume is much smaller than the total degree.

6. **Cyclic-n problem** [13]

The general formulation goes as follows:

$$P(\mathbf{x}) = \begin{cases} \sum_{i=1}^{n}\prod_{j=1}^{k} x_{(i+j)\, mod\, n}, \; k = 1, \cdots, n-1 \\ \prod_{j=1}^{n} x_j - 1, \end{cases}$$

This system is widely considered as a benchmark problem for polynomial system solving. We have solved this system up to dimension 11.

7. **The construction of Virasoro algebras**[38]

$$P(\mathbf{x}) = \begin{cases} 8x_1^2 + 8x_1x_2 + 8x_1x_3 + 2x_1x_4 + 2x_1x_5 + 2x_1x_6 + 2x_1x_7 \\ -8x_2x_3 - 2x_4x_7 - 2x_5x_6 - x_1, \\ 8x_1x_2 - 8x_1x_3 + 8x_2^2 + 8x_2x_3 + 2x_2x_4 + 2x_2x_5 + 2x_2x_6 \\ +2x_2x_7 - 2x_4x_6 - 2x_5x_7 - x_2, \\ -8x_1x_2 + 8x_1x_3 + 8x_2x_3 + 8x_3^2 + 2x_3x_4 + 2x_3x_5 + 2x_3x_6 \\ +2x_3x_7 - 2x_4x_5 - 2x_6x_7 - x_3, \\ 2x_1x_4 - 2x_1x_7 + 2x_2x_4 - 2x_2x_6 + 2x_3x_4 - 2x_3x_5 + 8x_4^2 \\ +8x_4x_5 + 2x_4x_6 + 2x_4x_7 + 6x_4x_8 - 6x_5x_8 - x_4, \\ 2x_1x_5 - 2x_1x_6 + 2x_2x_5 - 2x_2x_7 - 2x_3x_4 + 2x_3x_5 + 8x_4x_5 \\ -6x_4x_8 + 8x_5^2 + 2x_5x_6 + 2x_5x_7 + 6x_5x_8 - x_5, \\ -2x_1x_5 + 2x_1x_6 - 2x_2x_4 + 2x_2x_6 + 2x_3x_6 - 2x_3x_7 + 2x_4x_6 \\ +2x_5x_6 + 8x_6^2 + 8x_6x_7 + 6x_6x_8 - 6x_7x_8 - x_6, \\ -2x_1x_4 + 2x_1x_7 - 2x_2x_5 + 2x_2x_7 - 2x_3x_6 + 2x_3x_7 + 2x_4x_7 \\ +2x_5x_7 + 8x_6x_7 - 6x_6x_8 + 8x_7^2 + 6x_7x_8 - x_7, \\ -6x_4x_5 + 6x_4x_8 + 6x_5x_8 - 6x_6x_7 + 6x_6x_8 + 6x_7x_8 + 8x_8^2 - x_8, \end{cases}$$

The test results are summarized in the following table. In the table, $\mathcal{M}(\mathcal{A}^0)$ denotes the mixed volume of the extended support $\mathcal{A}^0 = (\mathcal{A}_1 \cup \{\mathbf{0}\}, \ldots, \mathcal{A}_n \cup \{\mathbf{0}\})$. The number of zeros in the examples are obtained from the numerical results of our algorithm. It is well-known that homotopy curves may converge to solutions in an algebraic variety with nonzero dimension, i.e., they may lead to non-isolated zeros of the target polynomial systems. In our root count, we do not exclude those numerical solutions at which the Jacobian matrices of the corresponding polynomial systems are almost singular. And our computation were carried out on a 400Mhz Intel Pentium II CPU with 256 MB of RAM, running SunOS 5.6.

System	Total Degree	$\mathcal{M}(A^0)$	# of Zeros in $\mathbb{C}^n$	CPU Time
Trinks	24	10	10	290ms
Eigenvalue	243	10	10	430ms
Economics-13	354294	2048	2048	8m2s200ms
Economics-14	1062882	4096	4096	22m35s620ms
Nash equilibia	1024	44	44	4s240ms
Benchmark i1	59049	66	50	4s100ms
Cyclic-10	3628800	35940	34940	1h33m46s
Cyclic-11	39916800	184756	184756	8h55m36s
Virasoro	256	256	256	37s 690 ms

Table 4.1: Numerical Results

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. L. Allgower and K. Georg (1990), *Numerical Continuation Methods, an Introduction*, Springer Series in Comput. Math., Vol. 13, Springer-Verlag(Berlin Herdelberg, New York).

[2] E. L. Allgower and K. Georg (1993), "Continuation and path following", *Acta Numerica*, 1-64.

[3] M. J. Best and K.Ritter (1985), *Linear Programming: Active Set Analysis and Computer Programs*, Prentice-Hall, Inc. New Jersey.

[4] 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.

[5] B. Buchberger (1985), "Gröbner basis: An algorithmic method in polynomial ideal theory", In *Multidimensional System Theory* (N.K. Bose, ed.), D. Reidel Publishing Company (Dordrecht Boston Lancaster), 184-232.

[6] J. Canny and J. M. Rojas (1991), "An optimal condition for determining the exact number of roots of a polynomial system", In *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, ACM, 96-101.

[7] S. N. Chow, J. Mallet-Paret, and J. A. Yorke (1978), "Finding zeros of maps: homotopy methods that are constructive with probability one", *Math. Comput.*, **32**, 887-899.

[8] S. N. Chow, J. Mallet-Paret, and J. A. Yorke (1979), "Homotopy method for locating all zeros of a system of polynomials", In *Functional differential equations and approximation of fixed points* (H. O. Peitgen and H. O. Walther, eds.), Lecture Notes in Mathematics Vol. 730, Springer-Verlag (Berlin Heidelberg), 77-88.

[9] M. Chu, T.-Y. Li and T. Sauer (1988) "Homotopy method for general lambda-matrix problems", *SIAM J. Matrix Anal. Appl.*, vol. 9, No. 4, pp 528-536.

[10] H.H. Chung (1998), *Polyhedral homotopy and its applications to polynomial system solving*, Ph.D. dissertation, Michigan State University.

[11] F. J. Drexler (1977), "Eine Methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen", *Numer. Math.* **29**, 45-58.

[12] I. Emiris (1994), *Sparse Elimination and Applications in Kinematics*, Ph.D. thesis, Computer Science Division, Dept. of Electrical Engineering and Computer Science, University of California (Berkeley).

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

[14] C. B. Garcia and W. I. Zangwill (1979), "Finding all solutions to polynomial systems and other systems of equations", *Math. Programming*, **16**, 159-176.

[15] P. Van Hentenryck, D. McAllester, and D. Kapur (1995), "Solving polynomial systems using a branch and prune approach". Technical Report No. CS-95-01, Department of Computer Science, Brown University.

[16] B. Huber, "Pelican manual", available via the author's Web page.

[17] B. Huber (1996), *Solving sparse polynomial systems*, Ph.D. dissertation, Cornell University.

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

[19] B. Huber and B. Sturmfels (1997), "Bernstein's theorem in affine space", *Discrete Comput. Geom.*, **7**, no. 2, 137-141.

[20] A. G. Khovanskií (1978), "Newton polyhedra and the genus of complete intersections", *Functional Anal. Appl.*, **12**(1), 38-46. Translated from *Funktsional. Anal. i Prilozhen.*, **12**(1), 51-61.

[21] A. G. Kushnirenko (1976), "Newton Polytopes and the Bèzout Theorem", *Functional Anal. Appl.*, **10**(3), 233-235. Translated from *Funktsional. Anal. i Prilozhen.*, **10**(3), 82-83.

[22] T. Y. Li (1983), "On Chow, Mallet-Paret and Yorke homotopy for solving systems of polynomials", *Bulletin of the Institute of Mathematics, Acad. Sin.*, **11**, 433-437.

[23] T. Y. Li (1997), "Numerical solution of multivariate polynomial systems by homotopy continuation methods", *Acta Numerica*, **6**, 399-436.

[24] T. Y. Li and T. Sauer (1989), "A simple homotopy for solving deficient polynomial systems", *Japan J. Appl. Math.*, **6**, 409-419.

[25] T. Y. Li, T. Sauer and J. A. Yorke (1989), "The cheater's homotopy: an efficient procedure for solving systems of polynomial equations", *SIAM J. Numer. Anal.*, **26**, 1241-1251.

[26] T. Y. Li and X. Wang (1992), "Nonlinear homotopies for solving deficient polynomial systems with parameters", *SIAM J. Numer. Anal.*, **29**, 1104-1118.

[27] T. Y. Li and X. Wang (1996), "The BKK root count in $\mathbf{C}^n$", *Math. Comp.*, **65**, no. 216, 1477-1484.

[28] G. L. Malajovich. pss 2. alpha, polynomial system solver, version 2. alpha. Distributed by the author through gopher.

[29] Richard D. McKelvey and Andrew McLennan (1997), "The maximal number of regular totally mixed Nash equilibria", *Journal of Economic Theory*, Volume 72, pages 411-425.

[30] A. P. Morgan (1986), "A homotopy for solving polynomial systems", *Appl. Math. Comput.*, **18**, 173-177.

[31] A. P. Morgan (1987), *Solving polynomial systems using continuation for engineering and scientific problems*, Prentice-Hall (Englewood Cliffs, N. J.).

[32] A. P. Morgan and A. J. Sommese (1987), "A homotopy for solving general polynomial systems that respect m-homogeneous structures", *Appl. Math. Comput.*, **24**, 101-113.

[33] A. P. Morgan and A. J. Sommese (1989:1), "Coefficient-parameter polynomial continuation", *Appl. Math. Comput.*, **29**, 123-160. Errata: *Appl. Math. Comput.*, **51**, 207 (1992).

[34] A.P. Morgan and C.W. Wampler (1990), "Solving a planar four-bar design problem using continuation", *ASME J. of Mechanical Design*, **112**, 544-550.

[35] C. H. Papadimitriou and K. Steiglitz (1982) *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Inc. New Jersey.

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

[37] J. M. Rojas and X. Wang (1996), "Counting affine roots of polynomial systems via pointed Newton polytopes", *J. of Complexity*, **12**, no. 2, 116-133.

[38] S. Schrans and and W. Troost (1990) "Generalized Virasoro Constructions for SU(3)", *Nuclear Phys. B*, Vol. 345, No. 2–3, pp. 584–606.

[39] I. R. Shafarevich (1977), *Basic Algebraic Geometry*, Springer-Verlag (New York).

[40] C. Traverso (1997), "The PoSSo test suite examples", [Online] Available at http://www.inria.fr/safir/POL/index.html.

[41] J. Verschelde and K. Gatermann (1995), "Symmetric Newton polytopes for solving sparse polynomial systems", *Adv. Appl. Math.*, **16**, 95-127.

[42] J. Verschelde (1995), "PHC and MVC: two programs for solving polynomial systems by homotopy continuation". Presented at the PoSSo workshop on software, Paris. Available by anonymous ftp to ftp.cs.kuleuven.ac.be in the directory /pub/NumAnal-ApplMath/PHC.

[43] J. Verschelde (1996), *"Homotopy continuation methods for solving polynomial systems*, Ph. D. thesis, Department of computer S cience, Katholieke Universiteit Leuven (Leuven, Belgium).

[44] J. Verschelde, K. Gatermann, and R. Cools (1996), "Mixed-volume computation by dynamic lifting applied to polynomial system solving", *Discrete Comput. Geom.*, **16**, no. 1, 69-112.

[45] L. T. Watson, S. C. Billups, and A. P. Morgan. "Algorithm 653: HOMPACK: a suite of codes for globbally convergent homotopy algorithms." *ACM Trans. Math. Softw.*, **13**(3): 281-310, 1987.

[46] A. H. Wright (1985), "Finding all solutions to a system of polynomial equations", *Math. Comput.*, **44**, 125-133.

[47] W. Zulener (1988), "A simple homotopy method for determining all isolated solutions to polynomial systems", *Math. Comput.*, **50**, 167-177.