

MINIMIZING MULTI-HOMOGENEOUS BÉZOUT NUMBERS BY A LOCAL SEARCH METHOD

TIEJUN LI AND FENGSHAN BAI

ABSTRACT. Consider the multi-homogeneous homotopy continuation method for solving a system of polynomial equations. For any partition of variables, the multi-homogeneous Bézout number bounds the number of isolated solution curves one has to follow in the method. This paper presents a local search method for finding a partition of variables with minimal multi-homogeneous Bézout number. As with any other local search method, it may give a local minimum rather than the minimum over all possible homogenizations. Numerical examples show the efficiency of this local search method.

1. INTRODUCTION

For a polynomial system one may wish to determine the total number of isolated solutions and then find all of them. This kind of problem is very common in many fields of science and engineering. Homotopy continuation has been developed to become an efficient method for approximating the full set of isolated solutions of polynomial systems. This method has been used to find all solutions of polynomial systems by Garcia and Zangwill, Chow, Li and Yorke [3, 1, 6, 8, 10, 11], etc., since the late 1970's. The multi-homogeneous homotopy method [7, 17, 13] can obtain all isolated zeros of a polynomial system with a relatively small amount of computation, and it is parallel in nature.

Consider a polynomial system

$$(1.1) \quad \mathbf{P}(\mathbf{x}) = \begin{pmatrix} p_1(x_1, \dots, x_n) \\ p_2(x_1, \dots, x_n) \\ \vdots \\ p_n(x_1, \dots, x_n) \end{pmatrix} = \mathbf{0},$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{C}^n$. The basic principle of the multi-homogeneous homotopy algorithm is that a partition of variables with a minimal multi-homogeneous Bézout number is first obtained so that a smaller number of solution curves is followed in the continuation. Wampler [16] gives an exhaustive search method on finding the optimal partition of variables. It works well for small scale problems. However, the process of finding that partition is likely an NP hard problem. Hence

Received by the editor September 18, 1998.

2000 *Mathematics Subject Classification.* Primary 65H10.

Key words and phrases. Multi-homogeneous Bézout number, polynomial system, homotopy method, local search method.

Supported by National Science Foundation of China G19871047 and National Key Basic Research Special Fund G1998020306.

this problem is considered from another point of view in this paper. Notice that our final goal is to obtain all solutions of a polynomial system as quickly as possible. All possible partitions of variables form a set. The reason that we spend time on finding an element with the minimal multi-homogeneous Bézout number in this set is to reduce the number of paths one has to follow in the continuation. Hence it is possible to reduce the total computational cost with a partition whose multi-homogeneous Bézout number is not minimal (but close), if that partition could be obtained with a fairly low cost. Four kinds of “neighboring” partitions of variables are defined, which gives a topology for the set. A local search method for minimizing the multi-homogeneous Bézout numbers is constructed. Since the number of the neighboring partitions is only $O(n^2)$ under such a topology, where n is as in (1.1), the computational cost of the local search method is much smaller than that of Wampler’s exhaustive search method. Any local search method may, of course, fall into local minimum, but the computational cost is fairly low, in particular for large scale problems. Many numerical examples indicate that one can obtain satisfactory results for most cases by the local search method.

The plan of this paper is as follows. In the next section, general ideas and basic concepts for homotopy methods are introduced. Basic problems and the method by Wampler [16] for minimizing the multi-homogeneous Bézout number are also given. A local search method for minimizing the multi-homogeneous Bézout numbers is presented in Section 3. The concept of neighboring partitions is crucial for the method. Numerical examples are presented in Sections 4 and 5. Finally, some discussion is given in Section 6.

2. MULTI-HOMOGENEOUS HOMOTOPY

A homotopy for solving (1.1) is

$$(2.1) \quad \mathbf{H}(\mathbf{x}, t) = (1 - t)\mathbf{Q}(\mathbf{x}) + t\mathbf{P}(\mathbf{x}) = 0,$$

where the system $\mathbf{Q}(\mathbf{x}) = (q_1(\mathbf{x}), \dots, q_n(\mathbf{x}))$ is chosen so that solutions for $\mathbf{Q}(\mathbf{x}) = 0$ are known or easy to get. The solution set of (1.1) can be obtained by following solution curves of (2.1) in the real variable t . The construction of $\mathbf{Q}(\mathbf{x})$ is crucial in the homotopy method. If the system $\mathbf{Q}(\mathbf{x})$ is not good enough, solution curves of (2.1) may have turning points or bifurcations, which may be difficult to handle. If the initial system $\mathbf{Q}(\mathbf{x})$ is chosen correctly, the following three properties (known collectively as TSA) hold [7]:

Property 0 (Triviality). The solutions of $\mathbf{Q}(\mathbf{x}) = 0$ are known.

Property 1 (Smoothness). The solution set of $\mathbf{H}(\mathbf{x}, t) = 0$ for $0 \leq t \leq 1$ consists of a finite number of smooth paths, each parameterized by t in $[0, 1)$.

Property 2 (Accessibility). Every isolated solution of $\mathbf{H}(\mathbf{x}, 1) = \mathbf{P}(\mathbf{x}) = 0$ can be reached by some path originating at $t = 0$. It follows that this path starts at a solution of $\mathbf{H}(\mathbf{x}, 0) = \mathbf{Q}(\mathbf{x}) = 0$.

Under the conditions above, the solution curves of (2.1) will be easy to follow, and any solution curve $\mathbf{x}(t)$ of $\mathbf{H}(\mathbf{x}, t) = 0$ must fall into one of the following two cases:

- Case (1). One isolated solution of $\mathbf{P}(\mathbf{x}) = 0$ is obtained at the point where $t = 1$.
- Case (2). The solution curve goes to infinity as t tends to 1.

Obviously, computation spent on following the solution curves of case (2) is a waste of time. Hence we wish to construct $\mathbf{Q}(\mathbf{x})$ so that $\mathbf{Q}(\mathbf{x}) = 0$ has as few solutions as possible while the homotopy $\mathbf{H}(\mathbf{x}, t)$ satisfies the TSA conditions above.

It is well known that the number of the isolated solutions of a polynomial system is bounded above by the Bézout number $D = \prod_{i=1}^n d_i$, where d_i is the degree of the i th equation in the system. The initial system $\mathbf{Q}(\mathbf{x})$ based on this bound has been constructed successfully [1, 3, 12, 6]. But the problem is that the Bézout number is often far larger than the number of isolated solutions that the system actually has. A well-known example is the eigenvalue problem. Let

$$(2.2) \quad \mathbf{Ax} = \lambda\mathbf{x}, \quad \mathbf{x} \in \mathbf{C}^n, \quad \lambda \in \mathbf{C}.$$

We can view it as a polynomial system in the following way:

$$(2.3) \quad \begin{cases} \mathbf{Ax} &= \lambda\mathbf{x}, \quad \mathbf{x} \in \mathbf{C}^n, \quad \lambda \in \mathbf{C}, \\ \eta^T \mathbf{x} &= 1, \end{cases}$$

where $\eta \in \mathbf{C}^n$ is a known vector. Clearly, the Bézout number of this system is 2^n . But we all know that the eigensystem only has n solutions, counting multiplicities. Since $2^n \gg n$, the homotopy method may waste too much time.

With the help of algebraic geometry, great progress has been made on homotopy methods recently. The multi-homogeneous Bézout numbers, which rely on the partition of the variables, are also upper bounds for the number of the isolated zeros of a polynomial system. They are normally less (some times even far less) than the Bézout number. In general, different ways of partitioning the variables produce different multi-homogeneous Bézout numbers. One certainly hopes to find a partition whose multi-homogeneous Bézout number is the smallest among all possible partitions of variables. With this partition, the full set of solutions can be obtained by following a smaller number of paths.

Let variables in a polynomial system be $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{C}^n$, and a partition of it be $\mathbf{T} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$, where

$$\mathbf{x}^{(j)} = (x_{j1}, \dots, x_{jk_j}), \quad j = 1, 2, \dots, m.$$

Here $K = [k_1, \dots, k_m]$ is called the partitioning vector. The subscripts above clearly satisfy $\sum_{j=1}^m k_j = n$, and $m \leq n$.

Example 2.1. Let $\mathbf{x} = (x_1, \dots, x_8) \in \mathbf{C}^8$. We partition them into four groups as $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(4)})$ where

$$\mathbf{x}^{(1)} = \{x_1, x_2, x_3, x_4\}, \quad \mathbf{x}^{(2)} = \{x_5, x_6\}, \quad \mathbf{x}^{(3)} = \{x_7\}, \quad \mathbf{x}^{(4)} = \{x_8\}.$$

The partitioning vector here is $K = [4, 2, 1, 1]$. For simplicity, we denote the partition as

$$\{1, 2, 3, 4\}, \quad \{5, 6\}, \quad \{7\}, \quad \{8\}.$$

Clearly, the partition above is identical to

$$\{1, 4, 3, 2\}, \quad \{6, 5\}, \quad \{8\}, \quad \{7\}.$$

Assume that the degree of the polynomial $p_i(\mathbf{x})$ with respect to $\mathbf{x}^{(k)}$ in (1.1) is d_{ik} , where $k = 1, 2, \dots, m$. Then the corresponding degree matrix under this

partition of variables is defined as

$$D = \begin{pmatrix} d_{11} & \cdots & d_{1m} \\ d_{21} & \cdots & d_{2m} \\ \vdots & & \\ d_{n1} & \cdots & d_{nm} \end{pmatrix}.$$

The multi-homogeneous Bézout number of the system (1.1) with respect to this partition of variables, denoted by B_m , is defined as the coefficient of $\alpha_1^{k_1} \cdots \alpha_m^{k_m}$ in the following polynomial of $\alpha_1, \dots, \alpha_m$:

$$\varphi(\alpha_1, \dots, \alpha_m) = (d_{11}\alpha_1 + \cdots + d_{1m}\alpha_m) \cdots (d_{n1}\alpha_1 + \cdots + d_{nm}\alpha_m),$$

where d_{ij} ($i = 1, 2, \dots, n, j = 1, 2, \dots, m$) are elements of the degree matrix D . Alternatively, the multi-homogeneous Bézout number B_m under this partition could also be computed through

$$B_m = \sum_{(i_1, \dots, i_n)} d_{1i_1} \cdots d_{ni_n},$$

where i_j ($1 \leq j \leq n$) is chosen from the set $\{1, 2, \dots, m\}$, and the number i is chosen exactly k_i times ($1 \leq i \leq m$). Consider a simple example by Wampler [16].

Example 2.2. Let a polynomial system

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^3,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= x_1^2 + x_2 + 1, \\ p_2(\mathbf{x}) &= x_1x_3 + x_2 + 2, \\ p_3(\mathbf{x}) &= x_2x_3 + x_3 + 3. \end{aligned}$$

The total degree, i.e., the Bézout number, is 8. There are five ways to multi-homogenize this system. It is easy to list all five partitions of the variables for this problem, together with the partitioning vectors K , degree matrices D , and the corresponding multi-homogeneous Bézout number. The first partition of variables is $\mathbf{x}^{(1)} = \{x_1, x_2, x_3\}$. It gives nothing but the classic Bézout number. We list all the others below.

$$\begin{aligned} \mathbf{x}^{(1)} = \{x_1, x_2\}, \mathbf{x}^{(2)} = \{x_3\}, & & \mathbf{x}^{(1)} = \{x_1, x_3\}, \mathbf{x}^{(2)} = \{x_2\}, \\ K = [2, 1], & & K = [2, 1], \\ D = \begin{pmatrix} 2 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, & & D = \begin{pmatrix} 2 & 1 \\ 2 & 1 \\ 1 & 1 \end{pmatrix}, \\ B_2 = 4, & & B_2 = 8, \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{(1)} = \{x_1\}, \mathbf{x}^{(2)} = \{x_2, x_3\}, & & \mathbf{x}^{(1)} = \{x_1\}, \mathbf{x}^{(2)} = \{x_2\}, \mathbf{x}^{(3)} = \{x_3\}, \\ K = [1, 2], & & K = [1, 1, 1], \\ D = \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 2 \end{pmatrix}, & & D = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \\ B_2 = 6, & & B_3 = 5. \end{aligned}$$

Example 2.3. Consider the eigenvalue problem (2.2) again. We know that the Bézout number of (2.3) is 2^n . Let the partition of the variables be $\{x_1, \dots, x_n\}, \{\lambda\}$. Then the corresponding multi-homogeneous Bézout number is exactly n . Hence the multi-homogeneous homotopy method can be used to solve the eigenvalue problem (2.2) efficiently with this partition. See [17, 13, 10, 11, 12] for details.

In order to obtain the minimal Bézout number, Wampler’s method has to go through all possible partitions. The total number of all possible partitions of variables is denoted by $B(n)$, which is called the Bell number. The combinatorics model for the Bell number is: All possible ways of putting n distinct balls into n identical boxes, where some of the boxes could be empty. We can easily conclude the following.

Proposition 2.1. $B(n) > (n/2)^{(n/2)}$ and $B(n) < n!$.

This means that the Bell number $B(n)$, hence the number of all possible partitions of variables, is not very big when n is small. But it increases very quickly as n grows. For example, $B(4) = 15, B(5) = 32$. But $B(10) = 115,975$ and $B(15) = 1,382,958,545$.

Before the local search method for finding the partition of variables with minimal multi-homogeneous Bézout number is presented, let us briefly mention the computation of multi-homogeneous Bézout number for a fixed partition of variables. A recursive algorithm is established by Wampler. The basic idea is as follows. Multi-homogeneous Bézout number B_m is determined uniquely by the partitioning vector $K = [k_1, \dots, k_m]$ and the degree matrix D . In order to use a row expansion algorithm, we denote B_m as $b(D, K, 1)$, and define $e_j = (0, \dots, 0, 1, 0, \dots, 0) \in \mathbf{Z}^n$. Then $b(D, K, i)$ has the recursive property [16]

$$b(D, K, i) = \begin{cases} 1, & \text{if } i = n + 1, \\ \sum_{j=1, k_j \neq 0}^m d_{ij} \times b(D, K - e_j, i + 1), & \text{otherwise,} \end{cases}$$

where $b(D, K, j)$ represents the multi-homogeneous Bézout number while the partitioning vector is K and the degree matrix, which is formed by deleting the first $j - 1$ rows from the matrix D . See details in [16].

From the analysis above, we can clearly see that the computational cost of both multi-homogeneous Bézout number for a fixed partition of variables and exhaustive searching for all the possible partitions is exponentially increasing as n grows. This kind of algorithm is unacceptable for large scale problems.

3. LOCAL SEARCH METHOD

A multi-homogeneous Bézout number is a function defined over the set of all possible partitions of variables. To search locally, the concept of neighborhood of a partition has to be defined first, which gives rise to a topology on this set.

Let $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{C}^n$, and a partition of it $\mathbf{T} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$, where

$$\mathbf{x}^{(j)} = (x_{j1}, \dots, x_{jk_j}), \quad j = 1, 2, \dots, m.$$

The partitioning vector is $K = [k_1, \dots, k_m]$. For simplicity, we often denote the partition above as $\{G_1, \dots, G_m\}$, where

$$G_j = \{j1, j2, \dots, jk_j\}, \quad j = 1, 2, \dots, m.$$

See Example 2.1 for details. To eliminate the nonuniqueness indicated by Example 2.1, we always assume

$$\begin{aligned}
 ij < il & \quad \text{for all } j < l; \\
 i1 < j1 & \quad \text{for all } i < j.
 \end{aligned}$$

That means the variables inside a group are arranged in an increasing order of their subscripts, and the groups in a partition are also in an increasing order of the subscripts of their first variables.

To give the concept of neighborhood for a partition, four kinds of operations on partitions are defined first. Such operations give motion between different partitions of variables. We will describe those operations using the partitions in Example 2.1, where $n = 8$. Assume the partition we start with is

$$\{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\}.$$

Operations on partitions are defined as follows.

1) *Splitting*. Factor out one element (i.e., one variable) from a group which has more than one element to form a new group by itself. For example,

$$\begin{array}{c}
 \{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\} \\
 \parallel \\
 \parallel \text{ splitting} \\
 \downarrow \\
 \{1, 2, 4\}, \{3\}, \{5, 6\}, \{7\}, \{8\}
 \end{array}$$

or

$$\begin{array}{c}
 \{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\} \\
 \parallel \\
 \parallel \text{ splitting} \\
 \downarrow \\
 \{1, 2, 3, 4\}, \{5\}, \{6\}, \{7\}, \{8\}.
 \end{array}$$

2) *Merging*. Combine one group which has only one element with another group to form a new group. For example,

$$\begin{array}{c}
 \{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\} \\
 \parallel \\
 \parallel \text{ merging} \\
 \downarrow \\
 \{1, 2, 3, 4, 8\}, \{5, 6\}, \{7\}.
 \end{array}$$

3) *Interchanging*. Exchange two elements in two different groups to form a new partition. For example,

$$\begin{array}{c}
 \{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\} \\
 \parallel \\
 \parallel \text{ interchanging} \\
 \downarrow \\
 \{1, 2, 3, 5\}, \{4, 6\}, \{7\}, \{8\}.
 \end{array}$$

4) *Transporting*. Move one element from one group which has more than one element to another group. For example,

$$\begin{array}{c} \{1, 2, 3, 4\}, \quad \{5, 6\}, \quad \{7\}, \quad \{8\} \\ \parallel \\ \parallel \quad \text{transporting} \\ \downarrow \\ \{1, 2, 3\}, \quad \{4, 5, 6\}, \quad \{7\}, \quad \{8\}. \end{array}$$

Definition 3.1. A partition $\mathbf{T} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$ of the variable \mathbf{x} is said to be connected with another partition $\mathbf{T}_1 = (\bar{\mathbf{x}}^{(1)}, \dots, \bar{\mathbf{x}}^{(l)})$ if \mathbf{T}_1 can be obtained by applying one of the four kinds operations defined above to the partition \mathbf{T} .

Note that operations 1) and 2) above establish relations between partitions which have a different number of groups, whereas 3) and 4) establish relations between partitions which have the same number of groups. They are all elementary operations connecting different partitions. All the other connections between different partitions can be obtained by compositing some of those elementary operations. Of course, there are many more options to define elementary operations. For example, it is not necessary to restrict the elementary operations moving only one element each time. The reason for doing so here is to reduce the number of neighboring partitions for any single partition. That makes our local search method easier.

Let the partitioning vector be $K = [k_1, k_2, \dots, k_m]$, where

$$k_i \begin{cases} > 2, & \text{for } 1 \leq i \leq t, \\ = 2, & \text{for } t + 1 \leq i \leq s, \\ = 1, & \text{for } s + 1 \leq i \leq m. \end{cases}$$

The following results give the number of partitions connected by each of the operations defined above.

Proposition 3.1. Assume a partition $\mathbf{T} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$ of variable \mathbf{x} is given.

- (1) The number of partitions connected with the partition \mathbf{T} through the splitting operation is

$$k_1 + k_2 + \dots + k_t + s - t = n - (m - t) \leq n \sim O(n).$$

- (2) The number of partitions connected with the partition \mathbf{T} through the merging operation is

$$s(m - s) + C_{m-s}^2 = \frac{(m - s)(m + s - 1)}{2} \leq C_n^2 \sim O(n^2).$$

- (3) The number of partitions connected with the partition \mathbf{T} through the interchanging operation is

$$\begin{aligned} k_1(k_2 + \dots + k_s + m - s) + k_2(k_3 + \dots + k_s + m - s) \\ + \dots + k_s(m - s) < n^2 \sim O(n^2). \end{aligned}$$

- (4) The number of partitions connected with the partition \mathbf{T} through the transporting operation is

$$\begin{aligned} k_1(m - 1) + k_2(m - 1) + \dots + k_s(m - 1) \\ = (n - m + s)(m - 1) \leq C_n^2 \sim O(n^2). \end{aligned}$$

Remark 3.1. The proof of these results is straightforward. Take result (1) as an example. The ways of splitting a group which has more than three elements are the same as the number of elements the group actually has. For a group with only two elements, it makes no difference which element is split out. Hence the result follows.

With the neighboring partitions defined above, a local search method can be constructed:

- 1) Generate a partition randomly. Search its neighboring partitions in a fixed order by applying one of the operations, which is chosen in the order of splitting first, then merging, interchanging, and finally transporting.
- 2) If one obtains a partition whose multi-homogeneous Bézout number is less than that of the present partition, set it as the initial partition. Restart the local search procedure above again until a partition whose multi-homogeneous Bézout number is less than any of its neighboring partition is obtained.
- 3) In order to improve the result, our local search method is performed k times, starting from k different randomly chosen initial partitions.

Algorithm 3.1.

- Step 0: Input k the number of times that the local search is performed.
Set a counter $nt := 1$.
- Step 1: If $nt \leq k$, then an initial partition T_0 of variables is randomly generated or chosen by the user, the multi-homogeneous Bézout number B_0 of the initial partition is computed;
else goto step 5.
- Step 2: All the neighboring partitions of partition T_0 are ordered as a sequence $\{T_1, T_2, \dots, T_N\}$, where N is the total number of the neighboring partitions of T_0 . Partitions in the sequence all result by applying one of the elementary operations, where the order of them are those by splitting first, then merging, interchanging, and finally transporting.
- Step 3: Compute the multi-homogeneous Bézout number B_1 of the partition T_1 .
 $i := 1$,
While $i \leq N$ and $B_i \geq B_0$
 $i := i + 1$;
 If $i \leq N$ compute the multi-homogeneous Bézout number B_i of the partition T_i ;
 else break;
end;
If $i \leq N$, then
 $T_0 := T_i$, $B_0 := B_i$,
 goto step 2;
else goto step 4;
- Step 4: Store the local minimal multi-homogeneous Bézout number and the corresponding partition. Set $nt := nt + 1$, and go to Step 1.
- Step 5: Find the smallest multi-homogeneous Bézout number over all k local minima obtained, which is taken as an approximation of the global minimal multi-homogeneous Bézout number, and the corresponding partition is used in further continuation.

Remark 3.2. The strategy of the local search method could be altered in various ways. For example, one could search the neighborhoods of a given partition at random rather than using a fixed order as above.

Remark 3.3. Assume the probability of finding the global optimal partition through a single time of local search is p . Then the probability of finding the optimal partition with minimal multi-homogeneous Bézout number through k times of local search is $1 - (1 - p)^k$, when the scale of the polynomial system n is sufficiently large. Hence it is reasonable to choose a suitable number k to make the probability of success larger. We will take $k = n$ in our computations later.

4. NUMERICAL EXAMPLES

Our local search method has been implemented successfully. All of our computations in this paper were performed on 486/DX/66, with C as the programming language. Computational results of our algorithm on several well-known polynomial systems are given in this section. These results give a clear picture of how our method works. More numerical results will be given in the next section. The efficiency of the local search method is quite evident from these computations.

Example 4.1. Let us consider the well-known eigenvalue problem first. We view it as a polynomial system as in (2.3) again, where $A \in \mathbf{C}^{7 \times 7}$ and $\lambda = x_8$. Computational results are as follows.

A partition is randomly generated at step 1 of Algorithm 3.1. The partition and the corresponding multi-homogeneous Bézout number are

$$\{1, 2, 3, 8\}, \quad \{4, 6, 7\}, \quad \{5\}, \quad B_3 = 900.$$

For simplicity here and throughout this section, we will not list out all partitions obtained by the local search method, but only those whose multi-homogeneous Bézout numbers decrease. They are

$$\begin{aligned} \{1, 2, 3, 8\}, \quad \{4, 5, 6, 7\}, & \quad B_2 = 225, \\ \{1, 4, 5, 6, 7\}, \quad \{2, 3, 8\}, & \quad B_2 = 104, \\ \{1, 2, 4, 5, 6, 7\}, \quad \{3, 8\}, & \quad B_2 = 35, \\ \{1, 2, 3, 4, 5, 6, 7\}, \quad \{8\}, & \quad B_2 = 7. \end{aligned}$$

The multi-homogeneous Bézout number of the final partition equals 7, which is the global minimum over all possible partitions.

Next we generate another partition randomly. The partition and the corresponding multi-homogeneous Bézout number are

$$\{1, 5, 6\}, \quad \{2, 3, 7, 8\}, \quad \{4\}, \quad B_3 = 900.$$

Partition sequence and the corresponding multi-homogeneous Bézout numbers obtained by the local search method are

$$\begin{aligned} \{1, 4, 5, 6\}, \quad \{2, 3, 7, 8\}, & \quad B_2 = 225, \\ \{1, 2, 4, 5, 6\}, \quad \{3, 7, 8\}, & \quad B_2 = 104, \\ \{1, 2, 3, 4, 5, 6\}, \quad \{7, 8\}, & \quad B_2 = 35, \\ \{1, 2, 3, 4, 5, 6, 7\}, \quad \{8\}, & \quad B_2 = 7. \end{aligned}$$

The multi-homogeneous Bézout number of the final partition is again 7.

For this eigenvalue problem, we start from any partition, our local search method always ends up with the partition whose multi-homogeneous Bézout number is 7.

Running local search method eight times (here $k = n$, the size of the system) only costs 1.65 seconds CPU time.

Example 4.2. Consider the Cassou-Nouges System, Equation 18 in the Appendix.

A partition is generated randomly at step 1. The partition and the corresponding multi-homogeneous Bézout number are

$$\{1, 3\}, \{4\}, \{2\}, \quad B_3 = 1511.$$

Partition sequence and the corresponding multi-homogeneous Bézout numbers obtained by the local search method are

$$\begin{aligned} \{1\}, \{2\}, \{3\}, \{4\}, & \quad B_4 = 752, \\ \{1\}, \{2, 3\}, \{4\}, & \quad B_3 = 440, \\ \{1\}, \{2, 3, 4\}, & \quad B_2 = 368. \end{aligned}$$

The multi-homogeneous Bézout number of the final partition is 368, which is, in fact, the global minimum.

Another partition is generated randomly. The partition and the corresponding multi-homogeneous Bézout number are

$$\{1, 3\}, \{2, 4\}, \quad B_2 = 1760.$$

Partition sequence and the corresponding multi-homogeneous Bézout numbers obtained by the local search method are

$$\begin{aligned} \{1\}, \{2, 4\}, \{3\}, & \quad B_3 = 876, \\ \{1\}, \{2\}, \{3\}, \{4\}, & \quad B_4 = 752, \\ \{1\}, \{2, 3\}, \{4\}, & \quad B_3 = 440, \\ \{1\}, \{2, 3, 4\}, & \quad B_2 = 368. \end{aligned}$$

The multi-homogeneous Bézout number of the final partition is 368 again.

For this system, we also tried to start from some other partitions. Our local search method always leads to the partition whose multi-homogeneous Bézout number is the global minimum. Running the local search method four times for this system only costs 0.05 seconds CPU time.

Example 4.3. Consider the Moeller-4 System, Equation 7 in the Appendix.

The partition is generated randomly at step 1. The partition and the corresponding multi-homogeneous Bézout number are

$$\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \quad B_4 = 135.$$

Partition sequence and the corresponding multi-homogeneous Bézout numbers obtained by the local search method are

$$\begin{aligned} \{1\}, \{2, 4\}, \{3\}, \{5\}, \{6\}, & \quad B_5 = 87, \\ \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, & \quad B_6 = 80, \\ \{1\}, \{2\}, \{3\}, \{4, 6\}, \{5\}, & \quad B_5 = 64, \\ \{1\}, \{2\}, \{3\}, \{4, 5, 6\}, & \quad B_4 = 41, \\ \{1\}, \{2\}, \{3, 4, 5, 6\}, & \quad B_3 = 31, \\ \{1, 3, 4, 5, 6\}, \{2\}, & \quad B_2 = 30, \\ \{1, 2, 3, 4, 5, 6\}, & \quad B_1 = 8. \end{aligned}$$

The multi-homogeneous Bézout number of the final partition is 8, which is again the global minimum.

For this system, we tried to start from some other partitions. Our local search method always gives the partition whose multi-homogeneous Bézout number is the

global minimum. Running the local search method six times for this system only costs 0.22 seconds CPU time.

5. FURTHER NUMERICAL RESULTS

To illustrate the efficiency of our local search algorithm, a large amount of computation has been done. Some remarks are given by comparing the computational results with those of Wampler’s method.

The local search method is tested with twenty polynomial systems, which are chosen from concrete problems in science and engineering. They could be considered as the benchmark of the polynomial system solver. Some symbols used below are listed:

- Eq(i) — The i th Equation in the Appendix.
- TD — Total Degree, i.e., the classical Bézout number.
- MB — Minimal multi-homogeneous Bézout number.
- SMV — Stable Mixed Volume, which also gives an upper bound for the number of isolated solutions, and is normally smaller than the minimal multi-homogeneous Bézout number. See details in [2].
- BRE — Time for Basic Row Expansion method, the basic approach used by Wampler for computing minimal multi-homogeneous Bézout number. See details in [16].
- EC — Time for Early Cutoff algorithm, an improved method by Wampler for computing the minimal multi-homogeneous Bézout number. See details in [16].
- DMC — Time for early cutoff with Degree Matrix Condensation, a further improved method by Wampler for computing the minimal multi-homogeneous Bézout number. See [16].

Basic properties of those twenty problems are given in Table 5.1. To compare with Wampler’s methods, some of the numerical results by Wampler’s methods are also given in Table 5.1.

Remark 5.1. From Table 5.1, it is very obvious that $BRE < EC < DMC$, which clearly shows that the DMC algorithm is the best among all three methods by Wampler. When n , the size of the problem, is less than 8, time spent for the computations with Wampler’s methods is acceptable. But when $n \geq 8$, the computer time for Wampler’s methods increases very quickly. Hence they are no longer efficient. The next table shows the results of our local search method with notations given below.

- MBLS — Optimal Multi-homogeneous Bézout number found by our Local Search method.
- TMLS — Time spent for obtaining the optimal Multi-homogeneous Bézout number above with our Local Search method.
- NLS — Number of the Local Search performed, i.e., the number k in Step 0 of Algorithm 3.1.
- Best? — Does the numerical result obtained by the local search method equal the minimal multi-homogeneous Bézout number?
(Y=Yes, N=No)

TABLE 5.1. Basic Properties and Results by Wampler’s Methods
 (Note that $< 0.01''$ shown in the table below means that the time spent for such computation cannot be measured by the computer we used.)

Eq(#)	n	TD	MB	SMV	BRE	EC	DMC
Eq(1)	2	16	10	6	$<0.01''$	$<0.01''$	$<0.01''$
Eq(2)	4	625	384	97	0.05''	0.05''	0.05''
Eq(3)	4	256	96	96	0.05''	0.05''	0.05''
Eq(4)	4	144	62	56	0.05''	$<0.01''$	$<0.01''$
Eq(5)	4	900	450	150	0.05''	0.05''	0.05''
Eq(6)	5	16	16	16	0.05''	0.05''	0.05''
Eq(7)	6	8	8	8	0.17''	0.11''	0.11''
Eq(8)	8	5764801	645120	–	18.56''	15.76''	10.0''
Eq(9)	8	128	16	–	9.60''	5.71''	5.16''
Eq(10)	6	64	20	–	0.30''	0.22''	0.15''
Eq(11)	11	2048	320	–	2°47'20.16''	39'36.68''	31'32.30''
Eq(12)	8	576	193	–	10.89''	2.58''	1.76''
Eq(13)	7	4608	1361	–	1.59''	1.32''	0.99''
Eq(14)	10	3628800	3628800	–	17°51.43''	15'52.63''	13'59.37''
Eq(15)	9	362880	362880	–	1°53.26''	1'44.58''	1'30.52''
Eq(16)	6	1024	216	–	0.16''	0.16''	0.16''
Eq(17)	5	108	56	–	0.05''	0.05''	0.05''
Eq(18)	4	1344	368	–	$<0.01''$	$<0.01''$	$<0.01''$
Eq(19)	10	64	56	–	15°45.32''	10'25.49''	9'53.69''
Eq(20)	8	256	160	–	13.01''	6.48''	4.70''

Remark 5.2. Comparing the results of Table 5.1 and those of Table 5.2, it is clear that the time spent for the local search method (the maximal time is less than 1 minute) is always acceptable, and is far less than those of Wampler’s methods for large scale problems. Using our local search method, global minimal multi-homogeneous Bézout numbers are missed only three times out of twenty. Even for those three whose global minima are missed, the results obtained by our local search method are not far away from the global minima. In other words, we obtain satisfactory results for those three problems.

In order to exploit the general behavior of the local search method, we choose several problems with larger dimension n . Starting from all possible partitions of variables, our local search method is applied. Table 5.3 was compiled from the results obtained, with more notations given below.

- nP — Number of all possible Partitions for the polynomial system, i.e., the Bell number $B(n)$.
- nPLSM — Total number of Partitions used as initial points, our Local Search method obtains a partition with minimal Multi-homogeneous Bézout number.
- LSR — Percentage of nPLSM over nP.
- nPM — Total number of Partitions with Minimal multi-homogeneous Bézout number.
- MBR — Percentage of nPM over nP.

TABLE 5.2. Results by the Local Search Method

Eq(#)	n	MBLS	TMLS	NLS	Best?
Eq(1)	2	10	0.11"	2	Y
Eq(2)	4	384	0.11"	4	Y
Eq(3)	4	96	0.22"	4	Y
Eq(4)	4	62	0.17"	4	Y
Eq(5)	4	450	0.17"	4	Y
Eq(6)	5	16	0.22"	5	Y
Eq(7)	6	8	0.33"	6	Y
Eq(8)	8	645120	5.55"	8	Y
Eq(9)	8	16	4.50"	8	Y
Eq(10)	6	20	0.55"	6	Y
Eq(11)	11	576	51.41"	11	N
Eq(12)	8	193	2.25"	8	Y
Eq(13)	7	1361	1.98"	7	Y
Eq(14)	10	3628800	30.53"	10	Y
Eq(15)	9	362880	9.51"	9	Y
Eq(16)	6	344	0.55"	6	N
Eq(17)	5	56	0.22"	5	Y
Eq(18)	4	368	0.16"	4	Y
Eq(19)	10	64	18.95"	10	N
Eq(20)	8	160	2.20"	8	Y

TABLE 5.3. Efficiency of Local Search Method

Eq(#)	Eq(9)	Eq(10)	Eq(12)	Eq(13)	Eq(16)	Eq(17)
n	8	6	8	7	6	5
nP	4140	203	4140	877	203	52
nPLSM	3938	178	3163	869	190	46
LSR	95%	88%	76%	99%	94%	89%
nPM	8	1	1	1	1	1
MBR	0.19%	0.49%	0.024%	0.11%	0.49%	1.92%

Remark 5.3. The results of Table 5.3 show that among all possible partitions of variables, the ratio of partitions with the minimal multi-homogeneous Bézout number is usually very low (normally less than 1%, shown by MBR in Table 5.3). However our local search method can obtain the partitions with the global minimal multi-homogeneous Bézout number by taking most of the partitions as initial points (around 90% as shown by LSR in Table 5.3). This shows that the local search method performs very well.

6. CONCLUSIONS AND DISCUSSIONS

A local search method has been developed in this paper. This method, supported by a large amount of numerical computation, can be seen to be fairly efficient. All the problems with 11 variables or fewer that we have tried can be solved by this method within one minute on an out-of-date personal computer. We are able to

make the size of the problems larger in our computations, but not without limitation. The reason is that the expensive computational cost of multi-homogeneous homotopy method comes from two aspects. One is finding a partition with the global minimal multi-homogeneous Bézout number, which is our focus of this paper. Another is the computation of the multi-homogeneous Bézout number for a given partition of variables. This makes the whole computation still too costly for large systems at present. Wampler's algorithm with row expansion recursion [16], which is the best method so far to our knowledge, is used to compute multi-homogeneous Bézout number for a given partition in our computations. Hence an efficient algorithm on computing multi-homogeneous Bézout number for a given partition remains an open question.

We conjecture that minimizing multi-homogeneous Bézout number is an NP hard problem. For such problems, it is only possible to get the exact solution for small scale problems. As the size of the problem becomes larger, we can only expect to get an approximate solution. Some heuristic approximate algorithms have been developed for the classical NP hard problems, such as traveling salesman problem (TSP) and 0-1 packing problem. We formulate the local search method for minimizing multi-homogeneous Bézout numbers.

Combinatorial geometry has come to play a more and more important role in homotopy method in the past five years [7, 4, 2, 5, 14, 15]. With some combinatorial geometric techniques based on algebraic geometry, one can get the so-called BKK bound (the upper bound of the isolated zeros in $(\mathbf{C}^*)^n$) and the stable mixed volume (the upper bound of the isolated zeros in \mathbf{C}^n), where $\mathbf{C}^* = \mathbf{C} \setminus \{0\}$. They normally give a tighter bound for the number of the isolated zeros than the multi-homogeneous Bézout number. That is certainly true for Equations 1 to 7, as shown in Table 5.1. Huber and Sturmfels [4] constructed the starting system with random lifting and convex hull techniques. Such a procedure is called polyhedral homotopy. It is, however, very expensive to obtain the starting system for the current polyhedral homotopy method.

Therefore the main point of this paper can be summarized as follows. For some polynomial systems, particularly large scale problems, we may obtain a partition with a satisfactory multi-homogeneous Bézout number (i.e., reasonably close to the global minimal multi-homogeneous Bézout number) within a reasonable amount of time by our local search method. This number could be larger than the stable mixed volume which is given by the polyhedral homotopy method. Consequently we have to follow some more homotopy paths. But the total computational cost may still be less.

APPENDIX

Polynomial systems used in our numerical computations are listed below. They are all concrete problems in applications.

Equation 1 ([2]). The bivariate system.

$$\mathbf{P}(x, y) = \begin{cases} ay + by^2 + cxy^3 & = 0, \\ dx + ex^2 + fyx^3 & = 0, \end{cases}$$

with generic coefficients a, b, c, d, e, f .

Equation 2 ([2]). System of E. R. Speer.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_4(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^4,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= 4\beta(n + 2a_1 - 8x_1)(a_2 - a_3) - x_2x_3x_4 + x_2 + x_4, \\ p_2(\mathbf{x}) &= 4\beta(n + 2a_1 - 8x_2)(a_2 - a_3) - x_1x_3x_4 + x_1 + x_3, \\ p_3(\mathbf{x}) &= 4\beta(n + 2a_1 - 8x_3)(a_2 - a_3) - x_1x_2x_4 + x_2 + x_4, \\ p_4(\mathbf{x}) &= 4\beta(n + 2a_1 - 8x_4)(a_2 - a_3) - x_1x_2x_3 + x_1 + x_3. \end{aligned}$$

The coefficients above satisfy the following:

$$\begin{aligned} a_1 &= x_1 + x_2 + x_3 + x_4, \\ a_2 &= x_1x_2x_3x_4, \\ a_3 &= x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1. \end{aligned}$$

Equation 3 ([2]). Symmetric four-bar mechanism.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_4(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^4,$$

where

$$\begin{aligned} p_l(x) &= a_{l1}x_1^2x_3^2 + a_{l2}x_1^2x_3x_4 + a_{l3}x_1^2x_3 + a_{l4}x_1^2x_4^2 + a_{l5}x_1^2x_4 \\ &+ a_{l6}x_1^2 + a_{l7}x_1x_2x_3^2 + a_{l8}x_1x_2x_3x_4 + a_{l9}x_1x_2x_3 \\ &+ a_{l10}x_1x_2x_4^2 + a_{l11}x_1x_2x_4 + a_{l12}x_1x_3^2 + a_{l13}x_1x_3x_4 \\ &+ a_{l14}x_1x_3 + a_{l15}x_1x_4^2 + a_{l16}x_1x_4 + a_{l17}x_2^2x_3^2 \\ &+ a_{l18}x_2^2x_3x_4 + a_{l19}x_2^2x_3 + a_{l20}x_2^2x_4^2 + a_{l21}x_2^2x_4 \\ &+ a_{l22}x_2^2 + a_{l23}x_2x_3^2 + a_{l24}x_2x_3x_4 + a_{l25}x_2x_3 \\ &+ a_{l26}x_2x_4^2 + a_{l27}x_2x_4 + a_{l28}x_3^2 + a_{l29}x_4^2, \end{aligned}$$

$$l = 1, \dots, 4.$$

Equation 4 ([2]). Caprasse system from PoSSo test suite.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_4(\mathbf{x})), \quad \mathbf{x} = (x, y, z, t) \in \mathbf{C}^4,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= y^2z + 2xyt - 2x - z, \\ p_2(\mathbf{x}) &= -x^3z + 4xy^2z + 4x^2yt + 2y^3t + 4x^2 \\ &\quad - 10y^2 + 4xz - 10yt + 2, \\ p_3(\mathbf{x}) &= 2yzt + xt^2 - x - 2z, \\ p_4(\mathbf{x}) &= -xz^3 + 4yz^2t + 4xzt^2 + 2yt^3 + 4xz \\ &\quad + 4z^2 - 10yt - 10t^2 + 2. \end{aligned}$$

Equation 5 ([2]). Cohn-2 system from PoSSo test suite.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_4(\mathbf{x})), \quad \mathbf{x} = (x, y, z, t) \in \mathbf{C}^4,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= x^3y^2 + 4x^2y^2z - x^2yz^2 + 288x^2y^2 + 207x^2yz + 1152xy^2z \\ &\quad + 156xyz^2 + xz^3 - 3456x^2y + 20736xy^2 + 19008xyz \\ &\quad + 82944y^2z + 432xz^2 - 497664xy + 62208xz + 2895984x, \\ p_2(\mathbf{x}) &= y^3t^3 + 4y^3t^2 - y^2zt^2 + 4y^2t^3 - 48y^2t^2 - 5yzt^2 + 108yzt \\ &\quad + z^2t + 144zt - 1728z, \\ p_3(\mathbf{x}) &= -x^2z^2t + 4xz^2t^2 + z^3t^2 + x^3z + 156x^2zt + 207xz^2t \\ &\quad + 1152xzt^2 + 288z^2t^2 + 432x^2z + 19008xzt - 3456z^2t \\ &\quad + 82944xt^2 + 20736zt^2 + 62208xz - 497664zt + 2985984z, \\ p_4(\mathbf{x}) &= y^3t^3 - xy^2t^2 + 4y^3t^2 + 4y^2t^3 - 5xy^2t - 48y^2t^2 + x^2y \\ &\quad + 108xyt + 144xy - 1728x. \end{aligned}$$

Equation 6 ([2]). Katsura-4 system from PoSSo test suite.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_5(\mathbf{x})), \quad \mathbf{x} = (x, y, z, t, u) \in \mathbf{C}^5,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= 2x^2 + 2y^2 + 2z^2 + 2t^2 + u^2 - u, \\ p_2(\mathbf{x}) &= xy + 2yz + 2zt + 2tu - t, \\ p_3(\mathbf{x}) &= 2xz + 2yt + t^2 + 2zu - z, \\ p_4(\mathbf{x}) &= 2xt + 2zt + 2yu - y, \\ p_5(\mathbf{x}) &= 2x + 2y + 2z + 2t + u - 1. \end{aligned}$$

Equation 7 ([2]). Moeller-4 system from PoSSo test suite.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_6(\mathbf{x})), \quad \mathbf{x} = (x, y, z, t, u, v) \in \mathbf{C}^6,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= y + u + v - 1, \\ p_2(\mathbf{x}) &= z + t + 2u - 3, \\ p_3(\mathbf{x}) &= y + t + 2v - 1, \\ p_4(\mathbf{x}) &= x - y - z - t - u - v, \\ p_5(\mathbf{x}) &= \alpha y z^3 + x^2 t u, \\ p_6(\mathbf{x}) &= \beta y t + z v, \end{aligned}$$

and

$$\alpha = -\frac{1569}{31250}, \quad \beta = -\frac{587}{15625}.$$

Equation 8.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_8(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^8,$$

where

$$\begin{aligned} p_j(\mathbf{x}) &= (x_2 x_3 x_5 x_8 - x_1 x_4 x_6 x_7) \\ &\quad \times [(x_6 x_7 - x_5 x_8)(C_j(x_2 - x_4)C_j'(x_1 - x_3)) \\ &\quad + (x_1 x_4 - x_2 x_3)(C_j(x_6 - x_8) + C_j'(x_5 - x_7))], \\ &\quad j = 1, \dots, 8. \end{aligned}$$

Equation 9 ([15]). A PUMA robot.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_8(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^8,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= x_1^2 + x_2^2 - 1, \\ p_2(\mathbf{x}) &= x_3^2 + x_4^2 - 1, \\ p_3(\mathbf{x}) &= x_5^2 + x_6^2 - 1, \\ p_4(\mathbf{x}) &= x_7^2 + x_8^2 - 1, \\ p_5(\mathbf{x}) &= 0.004731x_1x_3 - 0.3578x_2x_3 - 0.1238x_1 \\ &\quad - 0.001637x_2 - 0.9338x_4 + x_7 - 0.3571, \\ p_6(\mathbf{x}) &= 0.2238x_1x_3 + 0.7623x_2x_3 + 0.2638x_1 \\ &\quad - 0.007745x_2 - 0.6734x_4 - 0.6022, \\ p_7(\mathbf{x}) &= x_6x_8 + 0.3578x_1 + 0.004731x_2, \\ p_8(\mathbf{x}) &= -0.7623x_1 + 0.2238x_2 + 0.3461. \end{aligned}$$

This polynomial system models hand position and orientation of a PUMA robot.

Equation 10 ([15]). Camera motion from point matches.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_6(\mathbf{x})), \quad \mathbf{x} = (d_1, d_2, d_3, q_1, q_2, q_3) \in \mathbf{C}^6,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= -3.6d_1q_1 + 4.1d_1q_2 + 2.0d_1q_3 + 0.1d_1 + 4.1d_2q_1 \\ &\quad + 1.8d_2q_2 + 3.7d_2q_3 - 0.2d_2 + 2.0d_3q_1 + 3.7d_3q_2 \\ &\quad - 4.0d_3q_3 + 0.3d_3 + 0.1q_1 - 0.2q_2 + 0.3q_3 + 5.8, \\ p_2(\mathbf{x}) &= -2.140796d_1q_1 - 3.998792d_1q_2 + 3.715992d_1q_3 - 0.2828d_1 - 3.998792d_2q_1 \\ &\quad - 1.575196d_2q_2 - 3.998792d_2q_3 + 3.715992d_3q_1 - 3.998792d_3q_2 \\ &\quad - 2.140796d_3q_3 + 0.2828d_3 - 0.2828q_1 + 0.2828q_3 + 5.856788, \\ p_3(\mathbf{x}) &= 0.3464d_1q_1 + 0.1732d_1q_2 - 5.999648d_1q_3 - 0.1732d_1 + 0.1732d_2q_1 \\ &\quad - 5.999648d_2q_2 - 0.1732d_2q_3 + 0.3464d_2 - 5.999648d_3q_1 - 0.1732d_3q_2 \\ &\quad - 0.3464d_3q_3 - 0.1732d_3 - 0.1732q_1 + 0.3464q_2 - 0.1732q_3 + 5.999648, \\ p_4(\mathbf{x}) &= -5701.3d_1q_1 - 2.9d_1q_2 + 3796.7d_1q_3 - 1902.7d_1 - 2.9d_2q_1 \\ &\quad - 5698.7d_2q_2 + 1897.3d_2q_3 + 3803.3d_2 + 3796.7d_3q_1 + 1897.3d_3q_2 \\ &\quad + 5703.1d_3q_3 + 0.7d_3 - 1902.7q_1 - 3803.3q_2 + 0.7q_3 + 5696.9, \\ p_5(\mathbf{x}) &= -6.8d_1q_1 - 3.2d_1q_2 + 1.3d_1q_3 + 5.1d_1 - 3.2d_2q_1 \\ &\quad - 4.8d_2q_2 - 0.7d_2q_3 - 7.1d_2 + 1.3d_3q_1 - 0.7d_3q_2 \\ &\quad + 9.0d_3q_3 - d_3 + 5.1q_1 - 7.1q_2 - q_3 + 2.6, \\ p_6(\mathbf{x}) &= -d_1q_1 - d_2q_2 - d_3q_3 + 1. \end{aligned}$$

This polynomial system models the displacement of a camera between two positions.

Equation 11 ([15]). An inverse position problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_{11}(\mathbf{x})),$$

where

$$\mathbf{x} = (z_{21}, z_{22}, z_{31}, z_{32}, z_{33}, z_{41}, z_{42}, z_{43}, z_{51}, z_{52}, z_{53}) \in \mathbf{C}^{11}$$

and

$$\begin{aligned} p_1(\mathbf{x}) &= c_1^2 + z_{21}^2 + z_{22}^2 - 1, \\ p_2(\mathbf{x}) &= z_{31}^2 + z_{32}^2 + z_{33}^2 - 1, \\ p_3(\mathbf{x}) &= z_{41}^2 + z_{42}^2 + z_{43}^2 - 1, \\ p_4(\mathbf{x}) &= z_{51}^2 + z_{52}^2 + z_{53}^2 - 1, \\ p_5(\mathbf{x}) &= c_1z_{33} - c_2 + z_{21}z_{31} + z_{22}z_{32}, \\ p_6(\mathbf{x}) &= -c_3 - c_2 + z_{31}z_{41} + z_{32}z_{42} + z_{33}z_{43}, \\ p_7(\mathbf{x}) &= -c_4 - c_2 + z_{41}z_{51} + z_{42}z_{52} + z_{43}z_{53}, \\ p_8(\mathbf{x}) &= -c_1 - c_2 + z_{51}z_{61} + z_{52}z_{62} + z_{53}z_{63}, \\ p_9(\mathbf{x}) &= -c_1e_2z_{32} + d_2z_{21} + d_3z_{31} + d_4z_{41} + d_5z_{51} - e_1z_{22} \\ &\quad + e_2z_{22}z_{33} + e_3z_{32}z_{43} - e_3z_{33}z_{42} + e_4z_{42}z_{53} \\ &\quad - e_4z_{43}z_{52} + e_5z_{52}z_{63} - e_5z_{53}z_{62} - p_{61}, \\ p_{10}(\mathbf{x}) &= c_1e_2z_{31} + d_2z_{22} + d_3z_{32} + d_4z_{42} + d_5z_{52} + e_1z_{21} \\ &\quad - e_2z_{21}z_{33} - e_3z_{31}z_{43} + e_3z_{33}z_{41} - e_4z_{41}z_{53} \\ &\quad + e_4z_{43}z_{51} - e_5z_{51}z_{63} + e_5z_{53}z_{61} - p_{62}, \\ p_{11}(\mathbf{x}) &= c_1d_2 + d_3z_{33} + d_4z_{43} + d_5z_{53} + e_2z_{21}z_{32} \\ &\quad - e_2z_{22}z_{31} + e_3z_{31}z_{42} - e_3z_{32}z_{41} + e_4z_{41}z_{52} \\ &\quad - e_4z_{42}z_{51} + e_5z_{51}z_{62} - e_5z_{52}z_{61} - p_{63}. \end{aligned}$$

This system represents an inverse position problem for six-joint robot arms.

Equation 12 ([15]). A Heart-Dipole problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_8(\mathbf{x})), \quad \mathbf{x} = (a, b, c, d, t, u, v, w) \in \mathbf{C}^8,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= a + b - 0.6325, \\ p_2(\mathbf{x}) &= c + d - 0.8465, \\ p_3(\mathbf{x}) &= ta + ub - vc - wd - 0.1245, \\ p_4(\mathbf{x}) &= va + wb + tc + ud - 5.3452, \\ p_5(\mathbf{x}) &= at^2 - av^2 - 2ctv + bu^2 - bw^2 - 2duw - 1.4352, \\ p_6(\mathbf{x}) &= ct^2 - cv^2 + 2atv + du^2 - dw^2 + 2buw - 0.9896, \\ p_7(\mathbf{x}) &= at^3 - 3atv^2 + cv^3 - 3cvt^2 + bu^3 \\ &\quad - 3buw^2 + dw^3 - 3duw^2 - 0.3464, \\ p_8(\mathbf{x}) &= ct^3 - 3ctv^2 - av^3 + 3avt^2 + du^3 \\ &\quad - 3duw^2 - bw^3 + 3buw^2 - 3.1345. \end{aligned}$$

The numbers in the last terms of each equation are parameters for the model. They are chosen randomly in our computations.

Equation 13 ([15]). Butcher’s problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_7(\mathbf{x})), \quad \mathbf{x} = (x, y, z, u, v, w, t) \in \mathbf{C}^7,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= zu + yv + tw - w^2 - \frac{1}{2}w - \frac{1}{2}, \\ p_2(\mathbf{x}) &= zu^2 + yv^2 - tw^2 + w^3 + w^2 - \frac{1}{3}t + \frac{4}{3}w, \\ p_3(\mathbf{x}) &= xzv - tw^2 + w^3 - \frac{1}{2}tw + w^2 - \frac{1}{6}t + \frac{2}{3}w, \\ p_4(\mathbf{x}) &= zu^3 + yv^3 + tw^3 - w^4 - \frac{3}{2}w^3 + tw - \frac{5}{2}w^2 - \frac{1}{4}w - \frac{1}{4}, \\ p_5(\mathbf{x}) &= xzuv + tw^3 - w^4 + \frac{1}{2}tw^2 - \frac{3}{2}w^3 + \frac{1}{2}tw \\ &\quad - \frac{7}{4}w^2 - \frac{3}{8}w - \frac{1}{8}, \\ p_6(\mathbf{x}) &= xzv^2 + tw^3 - w^4 + tw^2 - \frac{3}{2}w^3 + \frac{2}{3}tw \\ &\quad - \frac{7}{6}w^2 - \frac{1}{12}w - \frac{1}{12}, \\ p_7(\mathbf{x}) &= -tw^3 + w^4 - tw^2 - \frac{3}{2}w^3 - \frac{1}{3}tw \\ &\quad - \frac{13}{12}w^2 + \frac{7}{24}w + \frac{1}{24}. \end{aligned}$$

Equation 14 ([15]). The cyclic n -roots problem where ($n = 10$).

$$\mathbf{P}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^n,$$

where

$$\begin{aligned} f_k(\mathbf{x}) &= \sum_{i=1}^n \prod_{j=1}^k x_{(i+j) \bmod n}, \quad k = 1, \dots, n-1, \\ f_n(\mathbf{x}) &= \prod_{j=1}^n x_j - 1. \end{aligned}$$

Equation 15 ([15]). The cyclic n -roots problem where ($n = 9$).

The equations are exactly the same as above but $n = 9$ instead of $n = 10$.

Equation 16 ([14]). Neurophysiology problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_6(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^6,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= x_1^2 + x_3^2 - 1, \\ p_2(\mathbf{x}) &= x_2^2 + x_4^2 - 1, \\ p_3(\mathbf{x}) &= x_5x_3^3 + x_6x_4^3 - C_1, \\ p_4(\mathbf{x}) &= x_5x_1^3 + x_6x_2^3 - C_2, \\ p_5(\mathbf{x}) &= x_5x_3^2x_1 + x_6x_4^2x_2 - C_3, \\ p_6(\mathbf{x}) &= x_5x_3x_1^2 + x_6x_4x_2^2 - C_4. \end{aligned}$$

Equation 17 ([14]). Chemical equilibrium problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_5(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^5,$$

where

$$\begin{aligned} p_1(\mathbf{x}) &= x_1x_2 + x_1 - 3x_5, \\ p_2(\mathbf{x}) &= 2x_1x_2 + x_1 + 2R_{10}x_2^2 + x_2x_3^2 \\ &\quad + R_7x_2x_3 + R_9x_2x_4 + R_8x_2 - Rx_5, \\ p_3(\mathbf{x}) &= 2x_2x_3^2 + R_7x_2x_3 + 2R_5x_3^2 + R_6x_3 - 8x_5, \\ p_4(\mathbf{x}) &= R_9x_2x_4 + 2x_4^2 + 4Rx_5, \\ p_5(\mathbf{x}) &= x_1x_2 + x_1 + R_{10}x_2^2 + x_2x_3^2 + R_7x_2x_3 \\ &\quad + R_9x_2x_4 + R_8x_2 + R_5x_3^2 + R_6x_3 + x_4^2 - 1. \end{aligned}$$

Equation 18 ([14]). Cassou-Nogues system.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_4(\mathbf{x})), \quad \mathbf{x} = (a, b, c, d) \in \mathbf{C}^4,$$

where

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

Equation 19 ([14]). A combustion chemistry problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_{11}(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^{11},$$

where

$$\begin{aligned}
 p_1(\mathbf{x}) &= x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5}, \\
 p_2(\mathbf{x}) &= x_3 + x_8 - 3.0 \times 10^{-5}, \\
 p_3(\mathbf{x}) &= x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} - 5.0 \times 10^{-5}, \\
 p_4(\mathbf{x}) &= x_4 + 2x_7 - 10^{-5}, \\
 p_5(\mathbf{x}) &= 0.5140437 \times 10^{-7}x_5 - x_1^2, \\
 p_6(\mathbf{x}) &= 0.1006932 \times 10^{-6}x_6 - x_2^2, \\
 p_7(\mathbf{x}) &= -0.7816278 \times 10^{-15}x_7 - x_4^2, \\
 p_8(\mathbf{x}) &= 0.1496236 \times 10^{-6}x_8 - x_1x_3, \\
 p_9(\mathbf{x}) &= 0.6194411 \times 10^{-7}x_9 - x_1x_2, \\
 p_{10}(\mathbf{x}) &= 0.2089296 \times 10^{-14}x_{10} - x_1x_2^2.
 \end{aligned}$$

Such a sparse system is very typical in a chemical equilibrium system.

Equation 20 ([14]). Inverse position problem.

$$\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_8(\mathbf{x})), \quad \mathbf{x} \in \mathbf{C}^8,$$

where

$$p_i(\mathbf{x}) = x_i^2 + x_{i+1}^2 - 1, \quad i = 1, 2, 3, 4,$$

$$\begin{aligned}
 p_i(\mathbf{x}) &= a_1x_1x_3 + a_2x_1x_4 + a_3x_2x_3 + a_4x_2x_4 + a_5x_5x_7 + a_6x_5x_8 \\
 &\quad + a_6x_5x_8 + a_7x_6x_7 + a_8x_6x_8 + a_9x_1 + a_{10}x_2 + a_{11}x_3 \\
 &\quad + a_{12}x_4 + a_{13}x_5 + a_{14}x_6 + a_{15}x_7 + a_{16}x_8 + a_{17}, \\
 &\quad i = 5, 6, 7, 8.
 \end{aligned}$$

This system is related to six-revolution-joint problem in mechanics.

Acknowledgments. The authors thank the referees for their careful reading and for their helpful remarks and suggestions.

REFERENCES

- [1] S. N. Chow, J. Mallet-Paret, J. A. Yorke (1979), A homotopy method for locating all zeros of a system of polynomials, In "Functional differential equations and approximation of fixed points" (H. O. Peitgen, H. O. Walther, eds.) Lecture Notes in Mathematics Vol. 730, Springer-Verlag (Berlin, Heidelberg), 77-88. MR **80k**:58012
- [2] T. A. Gao, T. Y. Li, X. S. Wang (1999), Finding isolated zeros of polynomial systems in \mathbf{C}^n with stable mixed volumes, J. Symbolic Comput. **28**, 187-211. CMP 2000:01
- [3] C. B. Garcia, W. I. Zangwill (1979), Finding all solutions to polynomial systems and other systems of equations, Math. Programming, **16**, 159-176. MR **80f**:65057
- [4] B. Huber, B. Sturmfels (1995), A polyhedral method for solving sparse polynomial systems, Math. Comp., **64**, 1541-1555.
- [5] B. Huber, B. Sturmfels (1997), Bernstein's theorem in affine space, Discrete Comput. Geom., **17**, 137-141. MR **98b**:52014
- [6] T. Y. Li (1983), On Chow, Mallet-Paret and Yorke homotopy for solving system of polynomials, Bull. Inst. Math. Acad. Sinica, **11** No. 3, 433-437. MR **85k**:58015
- [7] T. Y. Li (1997), Numerical solution of multivariate polynomial systems by homotopy continuation methods, Acta Numerica, 399-436. CMP 98:06
- [8] T. Y. Li, C. B. Garcia (1980), On the number of solutions to polynomial systems of equations, SIAM J. Numer. Anal. **17**, 540-546. MR **81m**:65074
- [9] T. Y. Li, T. Sauer, J. A. Yorke (1987), The random product homotopy and deficient polynomial systems, Numer. Math. **51**, 481-500. MR **88j**:65108
- [10] T. Y. Li, T. Sauer, J. A. Yorke (1987), Numerical solution of a class of deficient polynomial systems, SIAM J. Numer. Anal. **24**, 435-451. MR **89e**:90181

- [11] T. Y. Li, T. Sauer, J. A. Yorke (1989), A simple homotopy for solving deficient polynomial systems, *Japan J. Math. Appl. Math.* **6**, 409-419.
- [12] T. Y. Li, T. Sauer, J. A. Yorke (1989), The cheater's homotopy: an efficient procedure for solving systems of polynomial equations, *SIAM J. Numer. Anal.* **26**, 1241-1251. MR **90m**:65105
- [13] A. P. Morgan, A. J. Sommese (1987), A homotopy for solving general polynomial systems that respect m -homogeneous structures, *Appl. Math. Comput.*, **24**, 101-113. MR **88j**:65110
- [14] J. Verschelde, P. Verlinden, R. Cools (1994) Homotopies exploiting Newton polytopes for solving sparse polynomial systems, *SIAM J. Numer. Anal.*, **31**, 915-930. MR **94m**:65084
- [15] J. Verschelde, K. Gatermann, R. Cools (1996) Mixed-volume computation by dynamic lifting applied to polynomial systems solving, *Discrete Comput. Geom.* **16**, 69-112. MR **98g**:68171
- [16] C. W. Wampler (1992), Bézout number calculations for multi-homogeneous polynomial systems, *Appl. Math. Comput.*, **51**, 143-157. MR **93f**:65047
- [17] C. W. Wampler (1994), An efficient start system for multihomogeneous polynomial continuation, *Numer. Math.*, **66**, 517-523. MR **94i**:65065

SCHOOL OF MATHEMATICAL SCIENCES, PEKING UNIVERSITY, BEIJING, P. R. CHINA

DEPARTMENT OF MATHEMATICS, TSINGHUA UNIVERSITY, BEIJING, 100084, P. R. CHINA

E-mail address: fbai@math.tsinghua.edu.cn