2000年度日本オペレーションズ・リサーチ学会
春季研究発表会

A Combinatorial Problem Arising from Polyhedral Homotopies for Solving Polynomial Systems

02302370 東京工業大学 *武田 朗子 TAKEDA Akiko
01103520 東京工業大学 小島 政和 KOJIMA Masakazu

1 Introduction

Homotopy continuation is 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.

Let $P(x) = 0$ be a system of n polynomial equations in n unknowns. Denoting $P = (p_1, \dots, p_n)$, we want to find all isolated solutions $x = (x_1, \dots, x_n)$ of

$$\left.\begin{array}{r} p_1(x_1,\dots,x_n) = 0 \\ \vdots \\ p_n(x_1,\dots,x_n) = 0. \end{array}\right\} \quad (1)$$

The classical homotopy continuation method for solving (1) is to define a trivial system $Q(x) = (q_1(x), \dots, q_n(x))$ and then follow the curves in the real variable t which make up the solution set of

$$0 = H(x,t) = (1-t)Q(x) + tP(x).$$

A typical choice of the *start* system $Q(x)$ generates tremendously many initial points for solutions of the original problem $P(x) = 0$. However, in the last few years, a new technique for constructing $Q(x)$ has emerged, which provides a much tighter bound for the number of isolated zeros of $P(x)$. The so called *polyhedral homotopy* is then established for the new method and the homotopy paths so produced is much fewer. According to the recent article [2], we describe a problem involved in the construction of a new polynomial system $Q(x)$.

2 Formulation

Let us look at the following example of a system of polynomial equations:

$$P(x) \equiv \begin{pmatrix} x_1x_2^2x_3 - 2x_1^3x_3 + 3x_2^2x_3^4 + 6 \\ -x_1^2x_2^4 + x_2^2x_3^2 - 3 \\ 2x_1^3 - 3x_1x_2^2 + 4 \end{pmatrix}$$

For the jth term of the ith equation (say, $dx_1^{c^1}x_2^{c^2}x_3^{c^3}$), we define $c_{ij} \equiv (c^1, c^2, c^3)$. That is,

$$\begin{array}{r} c_{11} = (1,2,1), c_{12} = (3,0,1), c_{13} = (0,2,4), \\ c_{14} = (0,0,0), \\ c_{21} = (2,4,0), c_{22} = (0,2,2), c_{23} = (0,0,0), \\ c_{31} = (3,0,0), c_{32} = (1,2,0), c_{33} = (0,0,0). \end{array}$$

Let $S_i \equiv \{1, \dots, m_i\}$ for $i = 1, 2, \dots, n$, and in the above case, $n = 3$, $m_1 = 4$, $m_2 = 3$, $m_3 = 3$. Given real numbers $\omega_{ij} (i = 1, 2, \dots, n, \ \forall j \in S_i)$ chosen generically, we consider the system of linear inequalities:

$$\begin{array}{l} \beta_i - \langle c_{ij}, \alpha \rangle \leq \omega_{ij} \\ (i = 1, 2, \dots, n, \ \forall j \in S_i), \end{array} \quad (2)$$

where $\alpha, \beta \in R^n$, and formulate our problem as

Problem 2.1 Find all (α, β) which satisfies (2) with exactly two equalities for each $i = 1, 2, \dots, n$.

By solving Problem 2.1, we can construct a start system $Q(x)$ whose $q_i(x)$, $i = 1, 2, \dots, n$ consists of exactly two terms. We can algebraically solve such a system of polynomial equations (see [1]).

3 Transformation

Define $b_i \in R$ $(i = 1, 2, \dots, n)$ and $d \in R^n$ arbitrarily, and consider the linear program:

$$\begin{array}{lll} \text{P:} & \max & \sum_{i=1}^{n} b_i\beta_i + \langle d, \alpha \rangle \\ & \text{s.t.} & \beta_i - \langle c_{ij}, \alpha \rangle \leq \omega_{ij} \\ & & (i = 1, 2, \dots, n, \ \forall j \in S_i). \end{array}$$

Note that the set of constraint linear inequalities in P coincides with the system (2) of linear inequalities.

Let

$$\mathcal{F} = \{\boldsymbol{F} = (F_1, F_2, \ldots, F_n) : F_i \subset S_i,\ \sharp F_i \leq 2\ (i = 1, 2, \ldots, n)\}.$$

For each $\boldsymbol{F} = (F_1, F_2, \ldots, F_n) \in \mathcal{F}$, we consider a subproblem P($\boldsymbol{F}$) of P:

$$\begin{array}{lll} \mathrm{P}(\boldsymbol{F}): & \max & \sum_{i=1}^{n} b_i\beta_i + \langle \boldsymbol{d}, \boldsymbol{\alpha} \rangle \\ & \text{s.t.} & \beta_i - \langle \boldsymbol{c}_{ij_1}, \boldsymbol{\alpha} \rangle \leq \omega_{ij_1} \\ & & \beta_i - \langle \boldsymbol{c}_{ij_2}, \boldsymbol{\alpha} \rangle = \omega_{ij_2} \\ & & (i = 1, 2, \ldots, n, \\ & & \quad \forall j_1 \in S_i \backslash F_i,\ \forall j_2 \in F_i). \end{array}$$

Define $\mathcal{F}^*$ as

$$\left\{ \boldsymbol{F} \in \mathcal{F} : \begin{array}{l} \sharp F_i = 2\ (i = 1, 2, \ldots, n), \\ \mathrm{P}(\boldsymbol{F}) \text{ is feasible} \end{array} \right\}.$$

Thus, finding all solutions of Problem 2.1 has been reduced to computing optimal solutions of P($\boldsymbol{F}$) for all $\boldsymbol{F} \in \mathcal{F}^*$.

In order to enumerate all P($\boldsymbol{F}$) ($\boldsymbol{F} \in \mathcal{F}^*$), we introduce a tree structure into the subproblems $\{P(\boldsymbol{F}) : \boldsymbol{F} \in \mathcal{F}\}$. For every $k = 0, 1, 2, \ldots, n$, define $\mathcal{F}^k =$

$$\left\{ \boldsymbol{F} \in \mathcal{F} : \begin{array}{l} \sharp F_i = 2\ (i = 1, 2, \ldots, k), \\ \sharp F_j = 0\ (j = k+1, k+2, \ldots, n) \end{array} \right\}.$$

Now we regard each subproblem P($\boldsymbol{F}$) ($\boldsymbol{F} \in \mathcal{F}^k$) as a node at the kth level of the tree which we construct. A node P($\boldsymbol{F}'$) at the $(k+1)$th level is a child node of a node P($\boldsymbol{F}$) at the kth level if and only if $F'_j = F_j$ $(j = 1, 2, \ldots, k)$. We now apply the depth-first search to the tree. If a node P($\boldsymbol{F}$) at the kth level of the tree is infeasible, then all of its descendants are infeasible. Hence we terminate the node P($\boldsymbol{F}$) at the kth level in this case. For practical computational efficiency, we will propose to deal with the duals D($\boldsymbol{F}$) of P($\boldsymbol{F}$) ($\boldsymbol{F} \in \mathcal{F}$).

4 Implementation

We consider all possible distinct pairs $\{p, q\}$ of S_i with $1 \leq p < q \leq m_i$ and arrange them in the lexicographical order, i.e.,

$$L(S_i) \equiv \{\{1,2\}, \{1,3\}, \ldots, \{m_i - 1, m_i\}\},$$

where $1, 2, \ldots, m_i \in S_i$. For every $F_i = \{p, q\}$ in the list $L(S_i)$, we define $\mathrm{succ}(F_i; L(S_i)) =$

$$\begin{cases} \emptyset \text{ if } F_i \text{ is the last element in the list } L(S_i), \\ \text{the element succeeding to } F_i \text{ in the list } L(S_i) \\ \hfill \text{otherwise,} \end{cases}$$

and let $\mathrm{succ}(\emptyset; L(S_i))$ = the first element in the list $L(S_i)$.

Algorithm 4.1

Step 0: Let $\boldsymbol{F} = (\emptyset, \emptyset, \ldots, \emptyset) \in \mathcal{F}^0$, $\tilde{S}_i = S_i$ $(i = 1, 2, \ldots, n)$ and $k = 1$.

Step 1: If $k = 0$ then terminate. Otherwise, let

$$F_i = \begin{cases} F_i & \text{if } 1 \leq i \leq k-1, \\ \mathrm{succ}(F_k, L(\tilde{S}_k)) & \text{if } i = k, \\ \emptyset & \text{if } k+1 \leq i \leq n. \end{cases}$$

Step 2: If $F_k = \emptyset$, then let $\tilde{S}_k = S_k$, $k = k - 1$ and go to Step 1. Otherwise, go to Step 3.

Step 3: Solve D($\boldsymbol{F}$) to compute a basic optimal solution or detect the unboundedness of D($\boldsymbol{F}$). If D($\boldsymbol{F}$) is unbounded, go to Step 1. Otherwise, go to Step 4.

Step 4: If $k = n$, then output the optimal solution of P($\boldsymbol{F}$). Otherwise let $k = k + 1$. Go to Step 1.

5 Numerical Results

In this talk, we also present our numerical results on the widely considered benchmark system.

References

[1] M.Grötschel, L.Lovász and A. Schrijver, Geometric algorithms and combinatorial optimization (Springer, New York, 1988).

[2] Tien-Yien Li, "Solving polynomial systems by polyhedral homotopies", Taiwan Journal of Mathematics, 3, 251-279.