AAECC **8**, 257–263 (1997)

AAECC
Applicable Algebra in Engineering, Communication and Computing

Structural Gröbner Basis Detection

Bernd Sturmfels[1], **Markus Wiegelmann**[2]

[1] Department of Mathematics, University of California, Berkeley, CA 94720, USA
(e-mail: bernd@math.berkeley.edu)
[2] Fachbereich Mathematik, Universität Trier, Universitätsring, 15, D-54286 Trier, Germany
(e-mail: wiegelm@uni-trier.de)

Received: May 9, 1996; revised version: December 19, 1996

Abstract. We determine the computational complexity of deciding whether m polynomials in n variables have relatively prime leading terms with respect to some term order. This problem in NP-complete in general, but solvable in polynomial time in two different situations, when m is fixed and when $n - m$ is fixed. Our new algorithm for the second case determines a candidate set of leading terms by solving a maximum matching problem. This reduces the problem to linear programming.

Keywords: Gröbner basis, Variation of term orders, NP-completeness, Polynomial time algorithm, System of polynomial equations.

1 Introduction

Let $S = K[X_1, \ldots, X_n]$ be the polynomial ring in n variables over any field K. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ we write X^α for the monomial $X_1^{\alpha_1} \ldots X_n^{\alpha_n}$. Given a finite set $F = \{f_1, \ldots, f_m\}$ of polynomials in S, any term order $\prec$ on F can be represented by a positive weight vector $w \in \mathbb{R}_+^n$, see [4] or Proposition 1.11 in [8]. This means that

$$X^\alpha \prec X^\beta \;\Leftrightarrow\; w^T\alpha < w^T\beta$$

for any two monomials X^α and X^β occurring in F.

The following *Gröbner Basis Detection* problem was proposed in [1] as a key problem in the analysis of dynamic versions of Buchberger's algorithm.

(GBD) Given $F = \{f_1, \ldots, f_m\} \subset S$, decide if there exists — and if "Yes" find — a term order $w \in \mathbb{R}_+^n$ such that F is a Gröbner basis with respect to w.

In [1] this problem is solved using Minkowski addition of Newton polytopes to obtain the different term orders. The running time of that algorithm, however, depends not just on n and m but also on the maximal number k of terms and the maximal degree R of the polynomials f_i. More precisely, for fixed n the problem

(GBD) can be solved with a number of arithmetic operations which is polynomial in m, k and R. Note that only $log(R)$ and not R itself is part of the input in the sparse encoding of F.

The part of the algorithm where R is involved is the S-pair reduction. The complexity of normal form reduction was studied for example in [1, 3]. The given upper bounds are, however, exponential (in n and $log(R)$) in the sparse encoding of the polynomials and no hardness result is known up to date.

In this paper we focus on the other aspect of the algorithmic solution of the (GBD) problem, which is the variation of term orders. This is done by studying a related problem which can be solved without any reductions, namely, the following *Structural Gröbner Basis Detection* problem.

(**SGBD**) Given $F = \{f_1, \ldots, f_m\} \subset S$, decide if there exists — and if "Yes" find — a term order $w \in \mathbb{R}_+^n$ such that $LT_w(F)$ is a set of pairwise coprime monomials.

Here $LT_w(F)$ denotes the set of w-leading terms of the polynomials in F. Clearly, we can assume w.l.o.g. $m \leqq n$ and that none of the f_i is a constant.

Our aim is to describe the complexity of (SGBD) in the binary model of computation and to give a polynomial time algorithm when one exists. Our main results are the following two theorems.

Theorem 1. *Let $d \in \mathbb{N}_0$ be fixed. Then (SGBD) for $m = n - d$ polynomials in n variables is solvable in polynomial time in the input size of the problem.*

An algorithm for Theorem 1 will be given in Section 2. This is one of the first algorithms in Gröbner basis theory whose running time is polynomial in the number n of variables. The key ingredients are maximum bipartite matching and linear programming. In addition, this result allows to identify a class of square systems of equations for which it is easy to determine the exact number of zeros over the algebraic closure of K (Corollary 8).

Theorem 1 is contrasted by the following hardness result to be proved in Section 3. As a consequence, we finally prove that the (GBD) problem we started with is NP-hard (Corollary 11).

Theorem 2. (*SGBD*) *is strongly NP-complete in the sparse encoding of the input polynomials.*

In order to present the complete picture of the complexity of (*SGBD*) we briefly discuss the case of fixed m. There are at most k^m ways of picking one term from each polynomial in $F = \{f_1, \ldots, f_m\}$, where k is the maximal number of terms occurring in any f_i. We can list all choices in polynomial time. Polyhedral techniques for doing this efficiently are described in [1]. To decide whether a given choice is induced by a term order, we must decide the feasibility of a linear program. This can be done in polynomial time in the binary size of the input, for example by Khachian's Ellipsoid Method [6].

Corollary 3. For fixed m, (SGBD) can be solved in polynomial time.

We do not know if (SGBD) for fixed m can be solved in polynomial time in the RAM model of computation, in which each integer in the input is assumed to occupy unit space. This question leads to the well-known open problem whether there exists a strongly polynomial time algorithm for linear programming. As the new algorithm for fixed d also involves solving a linear program, the same statement holds for (SGBD) in this case.

2 The polynomial time algorithm

The aim of this section is to prove Theorem 1. The most interesting special case is $d = 0$. It will be treated first in Subsection 2.1.

2.1. Square systems

The following class of examples explains the combinatorial structure of our problem. See also Example 3.9 and Exercise (1) on page 29 in [8].

Example 4. Consider a system of n polynomials in n variables of the form

$$f_i := X_1^{a_{i1}} + X_2^{a_{i2}} + \cdots + X_n^{a_{in}} - 1 \quad (i = 1, \ldots, n).$$

For any term order $w \in \mathbb{R}_+^n$ on $F = \{f_1, \ldots, f_n\}$, the leading terms $LT_w(f_i) = \{X_{\sigma(i)}^{a_{i\sigma(i)}}\}$ are indexed by some function σ: $\{1, \ldots, n\} \mapsto \{1, \ldots, n\}$. The set $LT_w(F) = \{LT_w(f_1), \ldots, LT_w(f_n)\}$ is pairwise coprime if and only if σ is a permutation. Structural Gröbner basis detection (SGBD) becomes the question which one of the $n!$ possible permutations σ can be realized by a term order. There is at most one choice by Lemma 5 below.

Lemma 5. Let $F = \{f_1, \ldots, f_n\}$ be as in Example 4. A permutation σ cannot be realized by a term order if there is another permutation ρ such that

$$\prod_{i=1}^{n} a_{i\rho(i)} \geqq \prod_{i=1}^{n} a_{i\sigma(i)}.$$

Proof. Suppose $w \in \mathbb{R}_+^n$ is a term order which picks the leading term $X_{\sigma(i)}^{a_{i\sigma(i)}}$ for f_i. Let $\rho \neq \sigma$ be any other permutation. Since w prefers $X_{\sigma(i)}^{a_{i\sigma(i)}}$, we have

$$w_{\sigma(i)} a_{i\sigma(i)} > w_{\rho(i)} a_{i\rho(i)} \quad \text{for all } i \in \{1, 2, \ldots, n\} \quad \text{with } \sigma(i) \neq \rho(i).$$

There is at least one i with $\sigma(i) \neq \rho(i)$. Thus multiplication of all n inequalities yields

$$\prod_{i=1}^{n} w_{\sigma(i)} a_{i\sigma(i)} > \prod_{i=1}^{a} w_{\rho(i)} a_{i\rho(i)}$$

and thus, by division through the common factor $\prod_{i=1}^{n} w_i > 0$, the assertion. □

Lemma 5 is not restricted to the special case of Example 4 but it applies to any set of n polynomials in n variables. Disregarding terms which are not pure powers of variables, Lemma 5 tells us that the unique candidate for solving (SGBD) is a selection of pure powers $X_{\sigma(1)}^{a_{1\sigma(1)}}, \ldots, X_{\sigma(n)}^{a_{n\sigma(n)}}$ which maximizes the product $a_{1\sigma(1)} \ldots a_{n\sigma(n)}$. Finding such a selection is a bipartite maximum matching problem. The remaining question is whether such a candidate selection is indeed induced by a term order. This amounts to solving a linear program. The situation when more than one optimal assignment exists is taken care of automatically (see the Discussion following Algorithm 7).

Lemma 6. Let $F = \{f_1, \ldots, f_n\} \subset S$ and X^{α_i} a distinguished monomial in f_i for $i = 1, \ldots, n$. For any monomial $X^{\beta_i} \neq X^{\alpha_i}$ occurring in f_i consider the difference vector $\alpha_i - \beta_i$ and let Γ be the matrix whose rows are all these vectors for all i. Therefore exists a term order w such that $LT_w(f_i) = X^{\alpha_i}$ for $i = 1, \ldots, n$ if and

only if the linear system of inequalities $\Gamma w > 0$, $w > 0$ has a solution. Moreover, if a solution exists, there also exists a solution of binary size which is polynomial in the binary size of the sparsely encoded input polynomials.

Proof. Since every term order can be represented by a positive weight vector, the "only if" direction is clear. For the "if" part refine any feasible solution w of the linear program to a term order $\prec$ on F. The conditions $\Gamma w > 0$ exactly imply that the X^{α_i} become the leading terms with respect to $\prec$. The polynomial bound on the binary size of an appropriate weight vector follows by Cramer's rule. □

Algorithm 7. (solving (SGBD) for $m = n$).
INPUT: Polynomials $f_1, \ldots, f_n \in S$ in sparse encoding.

1. For $1 \leqq i, j \leqq n$ do:
   Let a_{ij} be the maximal exponent of X_j in f_i.
   (Set $a_{ij} = 0$ if no power of X_j appears in f_i)
   If there is a term $X_j^{a_{ij}} X^{\alpha}$ for $\alpha \neq 0$ in f_i, then set $a_{ij} = 0$.
2. Construct a bipartite graph $G = (V, E)$ with $2n$ vertices as follows:
   Let $V = \{v_1, \ldots, v_n, w_1, \ldots, w_n\}$ and $(v_i, w_j) \in E$ if and only if $a_{ij} > 0$.
3. Compute a maximum matching M of G which maximizes the product over all a_{ij} with $(v_i, w_j) \in M$.
4. If $|M| < n$ then stop, OUTPUT: No;
   else write $M = \{(v_i, w_{\sigma(i)}) | i = 1, \ldots, n\}$ for a permutation σ.
5. Let Γ be the matrix with n columns whose row vectors are $\alpha_{i\sigma(i)} - \beta_i$, where $i = 1, \ldots, n$ and X^{β_i} runs over all terms of f_i with $\beta_i \neq \alpha_{i\sigma(i)}$
6. Decide the linear programming feasibility problem $\Gamma w > 0$, $w > 0$.
   If no feasible solution w exists, then OUTPUT: No;
   else OUTPUT: Yes, any feasible solution w solves (SGBD).

Discussion: The correctness of Algorithm 7 follows from Lemma 5 and Lemma 6. In particular, if the permutation σ in step 4 is not uniquely determined, then the answer to the linear feasibility problem in step 6 will be *No* for every choice of σ by Lemma 5. Thus any choice is fine.

The main algorithmic subroutines are steps 3 and 6. The calculation of a maximum bipartite matching can be done in polynomial time, for example by the Hungarian method [2]. The linear programming feasibility problem only involves binary data which are part of the input. It can be solved in polynomial time in the size of the input, for example by Khachian's Ellipsoid method [6]. It follows that Algorithm 7 runs in polynomial time in the binary model of computation. □

We summarize a consequence which is specific to square systems.

Corollary 8. For $m = n$ there is at most one term order w up to equivalence such that F is a structural Gröbner basis with respect to w. If $X_i^{a_{i\sigma(i)}}$ is the w-leading term of f_i for $i = 1, \ldots, n$, then the system $f_1 = \cdots f_n = 0$ has exactly $\prod_{i=1}^{n} a_{i\sigma(i)}$ solutions in the algebraic closure of K counted with multiplicities.

2.2. Almost square systems

Suppose we have $m \leqq n$ polynomials where $d = n - m$ is fixed. The idea is to reduce (SGBD) to the case $n = m$. This can be done in the following way.

Lemma 9. Let $S_1 \cup \ldots \cup S_m$ be a partition of $\{X_1, \ldots, X_n\}$ and π the map from $k[X_1, \ldots, X_n]$ to $k[Y_1, \ldots, Y_m]$ defined by $\pi(X_i) = Y_j$ for $X_i \in S_j$. Let σ be a permutation of $\{1, \ldots, m\}$. There is a term order $\prec_X$ on $k[X_1, \ldots, X_n]$ such that $LT_{\prec_X}(f_i)$ has support in $S_{\sigma(i)}$ if and only if there is a term order $\prec_Y$ on $k[Y_1, \ldots, Y_m]$ such that $LT_{\prec_Y}(\pi(f_i))$ is a power of $Y_{\sigma(i)}$.

Proof. Given a term order $\prec_X$ by a weight vector w_X, we define w_Y to have j-th entry $w_Y(j) = \sum_{i \in S_j} w_X(i)$. For the if-direction, given $\prec_Y$ define $\prec_X$ by any refinement of the partial order $X^\alpha \prec_X X^\beta$ if $\pi(X^\alpha) \prec_Y \pi(X^\beta)$. □

Lemma 9 implies that we can answer the following question in polynomial time using Algorithm 1.

(SGBD)$_P$ Given a partition P of the variables $\{X_1, \ldots, X_n\}$ into m parts, does there exist a term order which picks as the leading terms of $f_1, \ldots, f_m$ monomials whose supports are contained in distinct parts of P?

A term order w solves (SGBD) if and only if there is a partition P such that w solves (SGBD)$_P$ for this P. The question how many such subproblems have to be considered is answered by the following combinatorial lemma.

Lemma 10. For $d = n - m$ fixed, the number of partitions of a set of n elements into m disjoint nonempty subsets is a polynomial of degree $2d$ in n.

Proof. The number of partitions is given by the Stirling numbers $S(n, m)$ of the second kind, which fulfill the recursion

$$S(n, m) = S(n-1, m-1) + mS(n-1, m),$$

see [7]. Since $S(n, n-1) = \binom{n}{2}$ the assertion is true for $d = 1$. Now proceed by induction on d using the recursion. □

In order to prove Theorem 1, we now just have to put things together.

Proof of Theorem 1. Let $d = n - m$ be fixed. By Lemma 10 there is a polynomial number of partitions P of $\{1, \ldots, n\}$ into m nonempty subsets. In order to decide (SGBD), it suffices to test (SGBD)$_P$ for all these P. By Lemma 9 this can be done in polynomial time using Algorithm 1. □

3 Hardness for the general case

In this section, we turn to the general case of m polynomials in n variables. Our aim is to show the NP-completeness result stated in Theorem 2.

Proof of Theorem 2. First, we show, that (SGBD) is in NP. Suppose the given set $F = \{f_1, \ldots, f_m\}$ has pairwise coprime leading monomials $X^{\alpha_i}, \ldots, X^{\alpha_m}$ for $f_1, \ldots, f_m$ with respect to some term order. Then the linear system $w^T(\alpha_i - \beta_i) > 0$ for all exponents $\beta_i \neq \alpha_i$ occurring in f_i and $i, \ldots, m$ has a solution w in $\mathbb{N}^n$ with binary size bounded by the polynomial in the size of F. Guess such a w and check that $\alpha_1, \ldots, \alpha_m$ have pairwise disjoint supports. This proves that (SGBD) is in NP.

In order to prove that (SGBD) is NP-hard, we reduce from the following known NP-complete problem, see for example [5], page 201.

(**SET PACKING**) Given a family $S = \{S_1, \ldots, S_k\}$ of subsets of $\{1, \ldots, v\}$, and a goal $m \in \mathbb{N}$. Are there m pairwise disjoint sets in S?

For a given instance (v, S, m) of (SET PACKING) we construct an instance of (SGBD) as follows. We take the polynomial ring

$$k[X_1, \ldots, X_v, Y_{11}, \ldots, Y_{1k}, \ldots, Y_{m1}, \ldots, Y_{mk}]$$

in $v + mk$ variables, and we encode S_j by the monomial $M_j := \prod_{i \in S_j} X_i$. Then we define m polynomials

$$f_1 = \sum_{j=1}^{k} Y_{1j} M_j, \ldots, f_m = \sum_{j=1}^{k} Y_{mj} M_j.$$

We claim that $F = \{f_1, \ldots, f_m\}$ is a structural Gröbner basis if and only if (v, S, m) is a "Yes"-instance of (SET PACKING).

To prove the "only if"-direction, let $F = \{f_1, \ldots, f_m\}$ be a structural Gröbner basis with leading terms $Y_{1i_1} M_{i_1}, \ldots, Y_{mi_m} M_{i_m}$. Then $M_{i_1}, \ldots, M_{i_m}$ must have disjoint support, and the m sets $S_{i_1}, \ldots, S_{i_m}$ are disjoint. For the "if"-direction, let $S_{i_1}, \ldots, S_{i_m}$ be m disjoint subsets of $\{1, \ldots, v\}$ in S. Define the coordinates of a weight vector $w \in N^{v+mk}$ as 1 for all variables except for $Y_{1i_1}, \ldots, Y_{mi_m}$, which get weight $v + 1$. Then the leading terms of $f_1, \ldots, f_m$ with respect to w are $Y_{1i_1} M_{i_1}, \ldots, Y_{mi_m} M_{i_m}$. Since they are pairwise coprime F is a structural Gröbner basis with respect to w. The proof is complete. □

As a consequence we obtain the following hardness result for the Gröbner basis detection problem (GBD) stated in the beginning of the introduction.

Corollary 11. (GBD) is NP-hard.

Proof. It suffices to argue that the set F constructed in the hardness part of the proof of Theorem 2 is a Gröbner basis if and only if it is a structural Gröbner basis. The "if"-part being clear, we assume that F is a Gröbner basis with respect to some term order w. We have to show that this implies that $LT(f_i)$ and $LT(f_j)$ are coprime for all i and j. The S-polynomial $s = S(f_i, f_j)$ reduces to zero with respect to F. Since all polynomials f_k for $k \notin \{i, j\}$ involve a variable Y_{lk} in their leading term, they do not occur in any reduction of s. Thus s reduces to zero by $\{f_i, f_j\}$ only. But since f_i and f_j do not have a common factor, their minimal syzygy is $(f_j, -f_i)$. Thus their leading terms have to be coprime since every syzygy on $(LT(f_i), LT(f_j))$ has to come from a syzygy on (f_i, f_j). □

Acknowledgements. Bernd Sturmfels supported in part by an NSF National Young Investigator Fellowship and a David and Lucile Packard Fellowship. Markus Wiegelmann is supported by the DFG through the graduate program "Mathematische Optimierung", Universität Trier. He also wishes to thank the International Computer Science Institute (ICSI) and its members for their great hospitality during his stay in Berkeley.

References

1. Gritzmann, P., Sturmfels, B.: Minkowski Addition of Polytopes: Computational Complexity and Applications to Gröbner Bases. SIAM J. Discrete Math. **6**, 246–269 (1993)
2. Lovasz, L., Plummer, M. D.: Matching Theory. Mathematics Studies, Vol. 121. Amsterdam: North-Holland 1986

3. Mishra, B., Yap, C.: Notes on Gröbner Bases. Inf. Sci. **48**, 219–252 (1989)
4. Ostrowski, A.: Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra. Jahresberichte Deutsche Math. Verein. **30**, 98–99 (1921)
5. Papadimitriou, C. H.: Computational Complexity, New York: Addison-Wesley 1994
6. Schrijver, A.: Theory of Linear and Integer Programming. New York: Wiley 1986
7. Stanley, R.: Enumerative Combinatorics. Monterey. CA: Wadsworth & Brooks 1986
8. Sturmfels, B.: Gröbner Bases and Convex Polytopes. Providence, RI: American Mathematical Society 1995