Homogenity, Pseudo-Homogenity, and Gröbner Basis Computations

Thomas BECKER

Fakultät für Mathematik und Informatik, Universität Passau,
Postfach 2540, D-8390 Passau, FR Germany

Abstract

Let $K[\mathbf{X}]$ be a multivariate polynomial ring over a field K, and let $\mathbf{U} \subseteq \mathbf{X}$. We call $f \in K[\mathbf{X}]$ pseudo-homogeneous in $\mathbf{U}$ if either it contains no variable of $\mathbf{U}$, or each of its terms does. $\mathbf{U} \subseteq \mathbf{X}$ is called an H-set for $F \subseteq K[\mathbf{X}]$ if each $f \in F$ is pseudo-homogeneous in $\mathbf{U}$. We show that when computing an elimination ideal of some ideal (F) of $K[\mathbf{X}]$ w.r.t. $\mathbf{V} \subseteq \mathbf{X}$, one may disregard all those elements of F that contain variables from some H-set $\mathbf{U} \subseteq \mathbf{X} \setminus \mathbf{V}$ for F. For given F and $\mathbf{V}$, we compute a maximal H-set for F contained in $\mathbf{V}$. Furthermore, we discuss how one can compute gradings by weighted total degree that make a given finite $F \subseteq K[\mathbf{X}]$ homogeneous. For certain limited purposes, one can then compute truncated Gröbner bases instead of full ones.

The detection of superfluous critical pairs in the Buchberger algorithm has long been one of the main tools to control the combinatorical complexity of the algorithm. In this paper, we point out how in a few very special cases, one can actually delete polynomials from the input set. The scope of possible applications of these methods is very limited. If they do apply, however, the resulting savings can be impressive, and testing whether or not they are applicable is quite cheap.

1. Pseudo-homogenity and H-sets Throughout, K will be a field, $\mathbf{X}$ a finite set of indeterminates. We assume familarity with the theory of Gröbner bases; a good introduction and further references may be found in [2]. For $f \in K[\mathbf{X}]$, we denote by $V(f)$ the set of all variables occuring in f with a non-zero exponent, and by $T(f)$ the set of all terms occuring in f with a non-zero coefficient. Let $\mathbf{U} \subseteq \mathbf{X}$, $f \in K[\mathbf{X}]$. We call f *pseudo-homogeneous* in the indeterminates $\mathbf{U}$ if the following condition holds: either $V(f) \cap \mathbf{U} = \emptyset$, or $V(t) \cap \mathbf{U} \neq \emptyset$ for all $t \in T(f)$. Now let $F \subseteq K[\mathbf{X}]$. A subset $\mathbf{U}$ of $\mathbf{X}$ is called an *H-set* for F if every $f \in F$ is pseudo-homogeneous in $\mathbf{U}$.

Lemma 1.1 *(i) The union of two H-sets for F is again an H-set for F.*

(ii) Let $\mathbf{V} \subseteq \mathbf{X}$. Then there is a unique H-set $\mathbf{U} \subseteq \mathbf{V}$ for F which is maximal w.r.t. inclusion among all H-sets for F that are contained in $\mathbf{V}$.

Proof (i) is immediate from the definitions. For $\mathbf{U}$ as described in (ii), take the union of all H-sets for F that are contained in $\mathbf{V}$. □

Proposition 1.2 *Let $\mathbf{V} \subseteq \mathbf{X}$, and suppose $\mathbf{U}$ is an H-set for F with $\mathbf{U} \subseteq \mathbf{X} \setminus \mathbf{V}$. Set $P = \{ f \in F \mid V(f) \cap \mathbf{U} = \emptyset \}$. Then*

$$(F) \cap K[\mathbf{V}] = (P) \cap K[\mathbf{V}].$$

In particular, if we take $\mathbf{V} = \emptyset$ *and for* $\mathbf{U}$ *any H-set for* F*, e.g. the maximal one, then* (F) *is proper iff* (P) *is proper.*

Proof The inclusion "$\supseteq$" is trivial. Now let $g \in (F) \cap K[\mathbf{V}]$. Then we can write g as a sum of monomial multiples of elements of F, say

$$g = \sum_{i=1}^{k} m_i f_i \quad (f_i \in F),$$

where the f_i are not necessarily distinct. We may now group the summands into two parts:

$$g = \sum_{i=1}^{k'} m_i f_i + \sum_{i=k'+1}^{k} m_i f_i,$$

where $V(m_i) \cap \mathbf{U} = \emptyset$ and $V(f_i) \cap \mathbf{U} = \emptyset$ for $1 \leq i \leq k'$, and for $k'+1 \leq i \leq k$, $V(m_i) \cap \mathbf{U} \neq \emptyset$ or $V(f_i) \cap \mathbf{U} \neq \emptyset$. We see that $V(t) \cap \mathbf{U} = \emptyset$ for each term in the first sum, whereas $V(t) \cap \mathbf{U} \neq \emptyset$ for every term in the second sum since $\mathbf{U}$ is an H-set for F. Setting $X_i = 0$ for all $X_i \in \mathbf{U}$, we see that

$$g = \sum_{i=1}^{k'} m_i f_i. \ \square$$

If F is a finite subset of $K[\mathbf{X}]$, and an elimination ideal of (F) is to be found w.r.t. $\mathbf{V} \subseteq \mathbf{X}$ by means of a Gröbner basis computation, then, by the above proposition, we may delete from F all polynomials that contain variables from the maximal H-set for F contained in $\mathbf{X} \setminus \mathbf{V}$. In particular, to test (F) for properness, we may delete all polynomials containing variables from the maximal H-set. To be able to use this in practice, we need of course an algorithm computing maximal H-sets.

Proposition 1.3 *Let* F *be a finite subset of* $K[\mathbf{X}]$, $\mathbf{V} \subseteq \mathbf{X}$*. Then the following algorithm computes the maximal H-set for* F *contained in* $\mathbf{V}$ *and the set* $P = \{ f \in F \mid V(f) \cap \mathbf{U} = \emptyset \}$.

Algorithm HSET
Specification: $(\mathbf{U}, P) \longleftarrow \mathrm{HSET}(\mathbf{V}, F)$
Given: a finite subset F of $K[\mathbf{X}]$, $\mathbf{V} \subseteq \mathbf{X}$
Find: $(\mathbf{U}, P)$ where $\mathbf{U}$ is the maximal H-set for F contained in $\mathbf{V}$,
and $P = \{ f \in F \mid V(f) \cap \mathbf{U} = \emptyset \}$

begin
$U \longleftarrow \mathbf{V}$; $P \longleftarrow \emptyset$; $Q1 \longleftarrow F$
repeat
$Q2 \longleftarrow \emptyset$; $U1 \longleftarrow U$
while $Q1 \neq \emptyset$ **do**
select f from $Q1$
$Q1 \longleftarrow Q1 \setminus \{f\}$
if $V(t) \cap U = \emptyset$ for some $t \in T(f)$ **then**

$U \longleftarrow U \setminus V(f)$
$P \longleftarrow P \cup \{f\}$
else $Q2 \longleftarrow Q2 \cup \{f\}$ end
end
$Q1 \longleftarrow Q2$
until $U1 = U$ end
return(U, P)
end HSET

Proof *Termination*: The **repeat**-loop is called again only if at least one element of $Q1$ has been placed in P and not in $Q2$; so the cardinality of $Q1$ at the very end of each call of this loop is less than that at its beginning, except for the last call. Termination of the **while**-loop is obvious.

Correctness: During a call of the **while**-loop, each element of $Q1$ is placed either in P or in $Q2$, so we have $F = P \cup Q2$ at the end of each call of the **repeat**-loop. Just before a polynomial is added to P, all its variables are removed from U. If we denote by $\mathbf{U}$ the final value of U, then obviously $\mathbf{U} \subseteq U$ for all values of U occuring. So we always have, for all $f \in P$, $V(t) \cap \mathbf{U} = \emptyset$ for all $t \in T(f)$; in particular, each $f \in P$ is pseudo-homogeneous in $\mathbf{U}$. An invariant of the **while**-loop is given by: for all $f \in Q2$, $V(t) \cap U1 \neq \emptyset$ for all $t \in T(f)$. After the last call of the **repeat**-loop, we have $U = U1 = \mathbf{U}$; so each $f \in Q2$ is pseudo-homogeneous in $\mathbf{U}$ and does not qualify to be in P. It is clear that $\mathbf{U} \subseteq \mathbf{V}$, and it remains to prove that $\mathbf{U}$ is actually the *maximal* H-set for F contained in $\mathbf{V}$. If $X \in \mathbf{V} \setminus \mathbf{U}$, then X must have been removed from U during a call of the **while**-loop. This means that there is $f \in F$ with $X \in V(f)$ and $V(t) \cap U = \emptyset$ for some $t \in T(f)$. (We are refering to the value of U just before X was removed from it.) Considering that $\mathbf{U} \subseteq U$, we see that $\mathbf{U} \cup \{X\}$ is not an H-set for f. □

If we assume that the terms of our polynomials are given by exponent vectors, then testing the **if**-condition $\mathbf{V}(t) \cap U = \emptyset$ requires at most $|\mathbf{X}|$ many comparisons. The number of terms that need to be looked at decreases by at least one during each call of the **repeat**-loop, so the total number of tests of the **if**-condition is at most quadratic in the total number of terms. We see that the complexity of HSET is extremely benign and certainly negligible in comparison to that of the Gröbner basis computation for (F). There are of course many situations where one knows in advance that the maximal H-set that one is looking for is empty: due to the presence of the polynomial $T_1 + \cdots + T_m - 1$, this is e.g. the case if one eliminates Kronecker variables $T_1, \ldots, T_m$. If the set of polynomials F arises in a somewhat random manner, however, the savings from dropping superflous polynomials can be considerable at virtually no cost (see Section "Examples" below). So whenever an elimination ideal of (F) w.r.t. $\mathbf{V} \subseteq \mathbf{X}$ is to be computed, one should routinely replace F by the subset P which HSET$(\mathbf{X} \setminus \mathbf{V}, F)$ outputs.

2. Homogeneous Gröbner bases As before, we let K be a field, $\mathbf{X} = \{X_1, \ldots, X_n\}$ a set of indeterminates. We denote by $T(\mathbf{X})$ the set of all terms in $\mathbf{X}$. By a *grading* of $K[\mathbf{X}]$, we mean a monoid homomorphism

$$\Gamma : (T(\mathbf{X}), \cdot, 1) \longrightarrow (\mathbb{N}, +, 0).$$

If Γ is a grading on $K[\mathbf{X}]$, we refer to the value $a_i = \Gamma(X_i)$ as the (Γ-)*weight* of X_i

$(1 \leq i \leq n)$; the weights determine Γ because of the formula

$$\Gamma(X_1^{\nu_1} \cdot \ldots \cdot X_n^{\nu_n}) = \sum_{i=1}^{n} \nu_i a_i.$$

In fact, every $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ determines a grading $\Gamma_{\mathbf{a}}$ where a_i is the $\Gamma_{\mathbf{a}}$-weight of X_i. $0 \neq f \in K[\mathbf{X}]$ is called Γ-*homogeneous* if $\Gamma(s) = \Gamma(t)$ for all $s, t \in T(f)$. This common value is then called the Γ-*degree* of f and is denoted by $\Gamma(f)$. The following theorem is well-known ([10], [8]).

Theorem 2.1 *Let Γ be a grading of $K[\mathbf{X}]$, F a finite subset of $K[\mathbf{X}]$ consisting of Γ-homogeneous polynomials. Then the output of any Buchberger algorithm with input F consists again of Γ-homogeneous polynomials. Let $d \in \mathbb{N}$, G_d the output of a Buchberger algorithm with input F which treats only those critical pairs $\{g_1, g_2\}$ that satisfy $\Gamma(\mathrm{lcm}(\mathrm{HT}(g_1), \mathrm{HT}(g_2))) \leq d$. Then G_d consists of precisely those elements of the output of the corresponding full Buchberger algorithm that have Γ-degree $\leq d$. Moreover, $f \xrightarrow[G_d]{*} 0$ for all $f \in (F)$ that satisfy $\max\{\, \Gamma(t) \mid t \in T(f) \,\} \leq d$.* □

A formal proof of the above theorem is of course somewhat tedious; however, an informal proof can easily be given as follows. It is easy to see that $0 \neq f, f', p \in K[\mathbf{X}]$ with f and p Γ-homogeneous and $f \xrightarrow[p]{} f'$ implies $\Gamma(p) \leq \Gamma(f)$ and f' Γ-homogeneous with $\Gamma(f) = \Gamma(f')$. Furthermore, if $g_1, g_2 \in K[\mathbf{X}]$ are Γ-homogeneous, then so is $\mathrm{SPol}(g_1, g_2)$ (unless equal to zero), and the Γ-degree of the latter is greater than or equal to the maximum of $\Gamma(g_1)$ and $\Gamma(g_2)$. We see that in the situation of the theorem all non-zero polynomials occurring in the Buchberger algorithm are Γ-homogeneous; moreover, if $\Gamma(\mathrm{lcm}(\mathrm{HT}(g_1), \mathrm{HT}(g_2))) > d$, then the normal form of $\mathrm{SPol}(g_1, g_2)$ (unless equal to zero) will have Γ-degree $> d$, and the same is true for all its "descendants" in the course of the algorithm. It follows that the truncated algorithm as described in the theorem produces all elements of the full Gröbner basis whose Γ-degree is $\leq d$, and it is again easy to see that this is enough to perform the indicated reductions.

Theorem 2.1 has been used to obtain complexity bounds on the Buchberger algorithm: one homogenizes an arbitrary input set w.r.t. the standard grading $\Gamma_{(1,\ldots,1)}$ by means of an additional variable, proves that the complexity of the homogenized algorithm is essentially the same as that of the regular one, and then uses complexity results on homogeneous Gröbner bases. Here, we propose the following way to exploit the theorem. Given an arbitrary finite ideal basis F, one may try to compute a non-trivial weight vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ such that each $f \in F$ becomes $\Gamma_{\mathbf{a}}$-homogeneous. The truncation at $d \in \mathbb{N}$ of a Gröbner basis of (F) is easier to compute in general than the full one (see Section "Examples" below), and it is still good enough to test for membership in (F) any $f \in K[\mathbf{X}]$ with $\max\{\, \Gamma_{\mathbf{a}}(t) \mid t \in T(f) \,\} \leq d$.

From now on, F will be a finite subset of $K[\mathbf{X}]$. To facilitate the computation of the desired weights, we first note the following connection with H-sets.

Lemma 2.2 *Let $\mathbf{U}$ be the maximal H-set for F, Γ a grading such that every $f \in F$ is Γ-homogeneous. Then $a_i = 0$ for all i such that $X_i \notin \mathbf{U}$.*

Proof Let $\mathbf{V} = \mathbf{U} \cup \{\, X_i \in \mathbf{X} \mid \Gamma(X_i) \neq 0 \,\}$, and assume for a contradiction that $\mathbf{V} \neq \mathbf{U}$. Then $\mathbf{V}$ is a proper superset of $\mathbf{U}$ and thus not an H-set for F. Hence there exists $f \in F$

and $s, t \in T(f)$ such that $V(s) \cap \mathbf{V} = \emptyset$ and $V(t) \cap \mathbf{V} \neq \emptyset$. It follows that $V(s) \cap \mathbf{U} = \emptyset$ and thus $V(t) \cap \mathbf{U} = \emptyset$ since $\mathbf{U}$ is an H-set for F. We see that $V(t) \cap (\mathbf{V} \setminus \mathbf{U}) \neq \emptyset$, so we must have $\Gamma(t) \neq 0$. On the other hand, $V(s) \cap \mathbf{V} = \emptyset$ implies $\Gamma(s) = 0$, and so f is not Γ-homogeneous, a contradiction. □

Let us assume w.l.o.g. that $\mathbf{U} = \{X_1, \ldots, X_k\}$ is the maximal H-set for F. It is clear from the above lemma that for any grading Γ, all $f \in F$ are Γ-homogeneous if and only if all $f(X_1, \ldots, X_k, 1, \ldots, 1)$ $(f \in F)$ are Γ-homogeneous. For the computation of the desired weights, we may thus set $X_{k+1} = \ldots = X_n = 1$, which amounts to assuming w.l.o.g. that $\mathbf{X}$ is the maximal H-set for F. Now we let $a_1, \ldots, a_n$ be unknowns. For $t = X_1^{\nu_1} \cdot \ldots \cdot X_n^{\nu_n} \in T(\mathbf{X})$, $\Gamma_{\mathbf{a}}(t)$ has the obvious meaning:

$$\Gamma_{\mathbf{a}}(t) = \sum_{i=1}^{n} a_i \nu_i.$$

Next, we fix some term $t_f \in T(f)$ for each $f \in F$. It is clear that the weight vectors whose corresponding gradings make all elements of F homogeneous are precisely the non-negative integer solutions of the system

$$\Gamma_{\mathbf{a}}(t_f) - \Gamma_{\mathbf{a}}(s) = 0 \quad (f \in F,\ s \in (T(f) \setminus \{t_f\})) \qquad (1)$$

of linear equations.

In order to compute the a_i, one would thus have to apply integer linear programming (ILP) (see e.g. [12]) to the equations (1) with addtional inequalities requiring

$$a_i \geq 0 \quad \text{for} \quad 1 \leq i \leq n. \qquad (2)$$

Furthermore, one would want to add the condition

$$\text{minimize } a_1 + \ldots + a_n \qquad (3)$$

to obtain low bounds and avoid solutions with $\gcd(a_1, \ldots, a_n) > 1$. Now this alone would of course produce the trivial solution $a_1 = \ldots = a_n = 0$, so an additional condition is needed to avoid that solution. The most general way to achieve this is to require

$$\sum_{i=1}^{n} a_i \geq 1. \qquad (4)$$

Then every non-trivial solution is admissible and so ILP will find one if one exists at all. On the other hand, it could be that there is a solution with many non-zero a_i (which is desirable in view of the intended truncation of the Buchberger algorithm), and another one with fewer non-zero a_i. To prefer the former over the latter, one has to require explicitely that many or all a_i be positive; but then again this may of course make the system unsolvable since some of the a_i may have to be zero. The only way around this would be to start with

$$a_i \geq 1 \quad (1 \leq i \leq n) \qquad (5)$$

and then systematically delete some of these until solvability is reached.

ILP is NP-complete and thus a complex algorithm ([4],[9]). However, experience seems to suggest that the attempt to compute a grading making all $f \in F$ homogeneous

still takes only a small fraction of the time required to compute a Gröbner basis of (F) (see Section "Examples" below). Moreover, the truncated Gröbner basis G_d can easily be extended to a truncation $G_{d'}$ at some bound $d' > d$: the same reasoning that we used to informally prove Theorem 2.1 shows that $G_{d'}$ can be obtained by performing on $G_d \cup F$ a Buchberger algorithm which treats only those critical pairs $\{g_1, g_2\}$ that satisfy $d < \Gamma(\mathrm{lcm}(\mathrm{HT}(g_1), \mathrm{HT}(g_2))) \leq d'$. The only difference between such a stepwise computation and the regular Buchberger algorithm is that one may be deviating from the principle of preferring those critical pairs $\{g_1, g_2\}$ where $\mathrm{lcm}(\mathrm{HT}(g_1), \mathrm{HT}(g_2))$ is minimal w.r.t. the term order in question. Under all implementations that we know of, the detection of unneccessary S-polynomials remains fully intact.

We can now summarize the results of this section as follows. Suppose we whish to test $f \in K[\mathbf{X}]$ for membership in (F). We first compute the maximal H-set $\mathbf{U}$ for F. If $\mathbf{U} = \emptyset$, then there is nothing that gradings can do for us. Otherwise, set $X_i = 1$ for all $X_i \notin \mathbf{U}$ and try to solve the ILP problem (1), (2), (3), (4). (If a solution is found with few non-zero a_i, one might try to solve (1), (2), (3), (5) and systematically delete conditions from (5) to obtain a solution with as many non-zero a_i as possible.) If this procedure yields a solution $\mathbf{b}$, then define $\mathbf{a}$ by setting $a_i = 0$ for those i with $X_i \notin \mathbf{U}$ and taking $\mathbf{b}$ for the remaining a_i with the obvious indexing. Compute $d = \max\{\, \Gamma_{\mathbf{a}}(t) \mid t \in T(f)\,\}$, and let G_d be the result of performing on F a Buchberger algorithm deleting all critical pairs $\{g_1, g_2\}$ with $\Gamma(\mathrm{lcm}(\mathrm{HT}(g_1), \mathrm{HT}(g_2))) > d$. Then $f \xrightarrow[G_d]{*} 0$ iff $f \in (F)$.

3. Abstract context The two situations described in Sections 1 and 2 can actually be placed in a common mathematical context. Let $S = K[\mathbf{X}]$, and suppose Γ is a grading on S. If we define

$$S_d = \{\, f \in S \mid \Gamma(t) = d \text{ for all } t \in T(f)\,\}$$

for all $n \in \mathbb{N}$, then we get $S = \oplus_{d \in \mathbb{N}} S_d$, and $f \in S$ is Γ-homogeneous with $\Gamma(f) = d$ iff $f \in S_d$. For any ideal I of S and all $d \in \mathbb{N}$ we set, accordingly, $I_d = I \cap S_d$. Such an ideal is then called Γ-**homogeneous** if $I = \oplus_{d \in \mathbb{N}} I_d$. The following proposition is now easy to prove.

Proposition 3.1 *The following are equivalent for any ideal I of S.*

(i) I is Γ-homogeneous.

(ii) I has a finite basis consisting of Γ-homogeneous polynomials. □

To describe the situation of Section 1 in a similar manner, let $\mathbf{X} = \mathbf{U} \cup \mathbf{Y}$, and denote by $(\mathbf{U})$ the ideal generated by $\mathbf{U}$ in S. Then clearly $S = K[\mathbf{Y}] \oplus (\mathbf{U})$. This could be called a **pseudograduation**, since it is like a graduation with only the two degrees 0 and 1. An element $f \in S$ is then pseudohomogeneous in the indeterminates $\mathbf{U}$ if either $f \in K[\mathbf{Y}]$ or $f \in (\mathbf{U})$. For an ideal I of $K[\mathbf{X}]$ we define

$$I_0 = I \cap K[\mathbf{Y}] \quad \text{and} \quad I_1 = I \cap (\mathbf{U}),$$

and we say that I is **pseudohomogeneous** w.r.t. $\mathbf{U}$ if $I = I_0 \oplus I_1$. One can now easily prove the following two propositions.

Proposition 3.2 *For any ideal of $K[\mathbf{X}]$, the following are equivalent.*

(i) I is pseudohomogeneous w.r.t. $\mathbf{U}$.

(ii) I has a finite basis con

Proposition 3.3 *Let I be a*
$I \cap K[\mathbf{Y}] = I_0 \cap K[\mathbf{Y}]$. □

This last proposition bri
Sections 1 and 2. If one no
include pseudograduations,
both cases simultaneously.

4. Examples One can of c
from dropping superfluous
one has to do is take a harm
with unpleasant coefficients
wreak havoc in a Buchberg
that the superfluous polyn
example, we used the Gröb

Example 4.1 Let F be th
$\mathbb{Q}[X_1, \ldots, X_5]$.

```
f1:= X1**3 * X3**2 - 5
f2:= X1**3 * X4 - X1**
f3:= X1 * X3**3 * X4 -
f4:= X1**2 * X2 * X4 +
f5:= X1 * X2**2 * X3 *
     + 3 * X2 * X5$
f6:= X1 * X4 * X5 - 2
```

To find the maximal H-se
assumption that $\mathbf{U} = \{$X1,
of f3 to find that the rema
HSET is $P = \{$f1, f2, f3$\}$.
or any subset thereof eq
$\{$X1, X3, X4$\}$ can be read
to find the elimination i
with high degrees should
term order with X4 $\gg$ X
X4 $\gg$ X3 $\gg$ X1 when work
to compute the two Gröb
to be reduced (♯ SPOL) a

Both computations foun

```
X1**5 -
```

(ii) I has a finite basis consisting of polynomials that are pseudohomogeneous in $\mathbf{U}$. □

Proposition 3.3 *Let I be an ideal of $K[\mathbf{X}]$ which is pseudohomogeneous w.r.t.* $\mathbf{U}$. *Then* $I \cap K[\mathbf{Y}] = I_0 \cap K[\mathbf{Y}]$. □

This last proposition brings out a strong analogy between the deletion principles of Sections 1 and 2. If one now generalizes the theory of graded structures of [13] so as to include pseudograduations, then one has found a uniform abstract framework that covers both cases simultaneously.

4. Examples One can of course make up arbitrarily impressive examples for the savings from dropping superfluous polynomials from the computation of an elimination ideal: all one has to do is take a harmless set of polynomials and then add a few unnecessary ones with unpleasant coefficients. (Just a few small prime numbers as leading coefficients can wreak havoc in a Buchberger algorithm.) The examples that we have chosen are such that the superfluous polynomials look similar to the necessary ones. For the following example, we used the Gröbner basis package of Reduce (see [5]) under VM/CMS.

Example 4.1 Let F be the set consisting of the following six polynomials in the ring $\mathbb{Q}[X_1, \ldots, X_5]$.

```
f1:= X1**3 * X3**2 - 5 * X1**2 * X3**3 + 3 * X1**2 * X3**3 $
f2:= X1**3 * X4 - X1**2 + 1$
f3:= X1 * X3**3 * X4 - X1 * X4$
f4:= X1**2 * X2 * X4 + X1 * X2**3 * X3 - 3 * X2 * X4$
f5:= X1 * X2**2 * X3 * X4 * X5 - X1 * X2 * X3**2 * X5**2
     + 3 * X2 * X5$
f6:= X1 * X4 * X5 - 2 * X2**2 * X3 - X2 * X4 * X5**2$
```

To find the maximal H-set $\mathbf{U}$ for F, we follow the algorithm HSET: starting with the assumption that $\mathbf{U} = \{\mathtt{X1}, \ldots, \mathtt{X5}\}$, we remove `X1` and `X4` because of `f2`, then `X3` because of `f3` to find that the remaining set $\{\mathtt{X2}, \mathtt{X5}\}$ is the maximal H-set. The second output of HSET is $P = \{\mathtt{f1}, \mathtt{f2}, \mathtt{f3}\}$. This means that the elimination ideal of (F) w.r.t. $\{\mathtt{X1}, \mathtt{X3}, \mathtt{X4}\}$ or any subset thereof equals that of (P). In particular, the elimination ideal w.r.t. $\{\mathtt{X1}, \mathtt{X3}, \mathtt{X4}\}$ can be read of immediately: it is generated by P. Now suppose we wish to find the elimination ideal w.r.t. $\{\mathtt{X1}\}$. Following the rule that variables occurring with high degrees should be placed lexicographically low, we used the lexicographical term order with $\mathtt{X4} \gg \mathtt{X5} \gg \mathtt{X2} \gg \mathtt{X3} \gg \mathtt{X1}$ when working with F and the one with $\mathtt{X4} \gg \mathtt{X3} \gg \mathtt{X1}$ when working with P. The table below gives the times that were needed to compute the two Gröbner bases; we also give the number of S-polynomials that had to be reduced (♯ SPOL) and the number of elements in the Gröbner basis (♯ GB).

	♯ GB	♯ SPOL	Time (sec)
F	10	76	94
P	4	13	.2

Both computations found the elimination ideal to be generated by

```
X1**5 - X1**3 - 8 * X1**2 + 8.
```

It is perhaps noteworthy that there cannot be a non-trivial grading that makes all elements of F homogeneous: the only elements of the maximal H-set are `X2` and `X5`, so the only non-zero weights can be a_2 and a_5. But these must satisfy $a_5 - 2a_2 = 0$ and $a_5 - (a_2 + 2a_5) = 0$ because of `f6`, and this implies that they must both be zero too.

Next, we give an example of a set F of polynomials that allows a non-trivial grading making every element of F homogeneous, so that one can compute truncated Gröbner bases as explained in Section 2. To this end, the source code of Reduce's Gröbner basis algorithm was modified by the author. The resulting program had to be run through the interpreter, which causes a considerable distortion of the computing times. We therefore give the number of S-polynomials (♯ SPOL) and the number of output polynomials (♯ GB) as measures of the expense of the computation.

Example 4.2 Let F consist of the following four elements of $\mathbb{Q}[X_1, \ldots, X_6]$.

```
f1:= x1**3 * x3**2 +  x1**2 * x3**3 $
f2:= x1 * x3**3 *x4**2 - 2 * x3 * x4 + 1$
f3:= x1 * x2 * x5**2  - 3 * x3 * x2**2 + x4 * x5 * x6**3$
f4:= x1 * x2**2 * x6 - x2 * x5**2 * x6$
```

Here, the H-set equals $\{X_2, X_5, X_6\}$, and if we set $\mathbf{a} = (0, 2, 0, 0, 1, 1)$, then `f1`-`f4` are $\Gamma_{\mathbf{a}}$-homogeneous with $\Gamma_{\mathbf{a}}$-degrees 0, 0, 4, and 5, respectively. We used the lexicographically refined total degree order with $X_1 > \ldots > X_6$. The full Gröbner basis computation shows a maximal $\Gamma_{\mathbf{a}}$-degree of 20, so we list the partial ones obtained by truncation at $d = 1, \ldots, 20$.

d	♯ SPOL	♯ GB
0–3	10	3
4	21	7
5	24	9
6	24	9
7	55	17
8	79	24
9	96	26
10	123	31
11	132	32

d	♯ SPOL	♯ GB
12	136	33
13	143	34
14	144	34
15	144	34
16	153	35
17	155	35
18	155	35
19	158	36
20	159	36

Example 4.3 A non-artificial example, originating in physics, to which the methods of section Section 2 apply is given by Example 1 of Section *Some special examples* in [1]. (The example is due to Gerdt, Shavachka, and Zharkov, see also [6].) Here, we are looking at 13 polynomials in 7 variables with no constant terms occurring at all. The maximal H-set thus consists of all seven variables. The system (1) of Section 2 stating homogenity consists of 93 equations (not all of which are pairwise different). Upon input of all 93 equations together with the conditions (2), (3), and (4) of Section 2, the ILP package of SAS produced the weight vector $\mathbf{a} = (1, 1, 1, 2, 2, 2, 3)$ as a solution. Computing time in full seconds was given as 0. Using a lexicographical term order with optimized variable ordering as in [1], we found that the full Gröbner basis G consisted of 20 polynomials of maximal $\Gamma_{\mathbf{a}}$-degree 7. 151 S-polynomials were treated during the computation. Again we used our modified source code in the interpreter mode of Reduce

to compute the truncations of G. Since this does not only take longer, but also uses more space than running compiled source code, we were not able to obtain all of G in this way. Computing G_d as described in Theorem 2.1, we got up to $d = 5$ (17 S-polynomials treated, $\sharp G_5 = 13$). Starting with G_3 and then extending to G_4, G_5, G_6 (with complete reduction after each step) took 2,3,11,13 S-polynomials, respectively, and produced the 19 elements of G whose $\Gamma_{\mathbf{a}}$-degree is less than 7. The attempt to extend to $G_7 = G$ then produced a stack overflow.

Acknowledgements. The author is indebted to P. Kleinschmidt for his advice and assistance with integer linear programming, and to one of the referees for pointing out the abstract mathematical context explained in Section 3.

References

[1] W. Böge, R. Gebauer, and H. Kredel, Some examples for solving systems of algebraic equations by calculating Gröbner Bases, J. Symbolic Computation 1, 83-98 (1985)

[2] B. Buchberger, Gröbner bases: an algorithmic method in polynomial ideal theory, in: N.K. Bose (ed.), Progress, Directions, and Open Problems in Systems Theory, 184–232, D. Riedel Publ. Comp. (Dordrecht 1985)

[3] M. Caboara, Dynamic evaluation of Gr-bner bases, Communication at the CAMASA '91 Meeting, Cagliari 1991

[4] E.G. Coffman Jr. (ed.), Computer and Job-Shop Scheduling Theory, John Wiley & Sons, New York (1976)

[5] R. Gebauer, H.M. Möller, On an installation of Buchberger's algorithm, JSC 6 2/3, 185–91 (1988)

[6] V.P. Gerdt, A.B. Shavachka, A. Zharkov, Computer algebra application for classification of integral non-linear evolution equation, JSC 1, 101–107 (1985)

[7] P. Gritzman and B. Sturmfels, Minkowski addition of polytopes: computaional complexity and applications to Groebner bases, Preprint, Cornell University (1990)

[8] D. Lazard, Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations, Proc. EUROCAL '83, Springer LNCS 162, 146–156 (1983)

[9] C.E. Miller, A.W. Tucker, and R.A. Zemlin, Integer programming formulation and traveling salesman problems, J. ACM 7 (1960)

[10] H.M. Moeller and T. Mora, Upper and lower bounds for the degree of Gröbner bases, Proc. EUROSAM '84, Springer LNCS 174, 172–183 (1984)

[11] T. Mora and L. Robbiano, The Groebner fan of an ideal, JSC 6 183-208 (1988)

[12] C.H. Papadimitrou and K. Steglitz, Combinatorical Optimization, Prentice Hall, Eaglewood Cliffs (1982)

[13] L. Robbiano, On the theory of graded structures, JSC 2, 139-170 (1986)