USING KAPUR-SUN-WANG ALGORITHM FOR THE GRÖBNER COVER

ANTONIO MONTES

Abstract. Kapur-Sun-Wang have recently developed a very efficient algorithm for computing Comprehensive Gröbner Systems that has moreover the required essential properties for being used as first step of the Gröbner Cover algorithm. We have implemented and adapted it inside the Singular grobcov library for computing the Gröbner Cover and there are evidences that it makes the canonical algorithm much more effective. In this note we discuss the performance of GC with KSW on a collection of examples.

Introduction

The Gröbner Cover (GC), described in [9], is a canonical description of the discussion of a parametric polynomial ideal $\mathcal{I} \subset K[\overline{a}][\overline{x}]$ over a computable infinite field K (whose algebraic closure is $\overline{K}$), $\overline{a} = a_1, \dots, a_m$ being the parameters and $\overline{x} = x_1, \dots, x_n$ the variables. It consists of a set of pairs of segment and basis $GC = \{(S_i, B_i) : 1 \leq i \leq s\}$, where the segments $S_i \subseteq \overline{K}^m$ form a partition of the parameter space $\overline{K}^m$ and are locally closed subsets (i.e. difference of varieties), that can be described in canonical form $S_i = \bigcup_j (\mathbb{V}(\mathfrak{p}_{ij}) \setminus \cup_k \mathbb{V}(\mathfrak{q}_{ijk}))$, where the $\mathfrak{p}$'s and $\mathfrak{q}$'s are prime ideals. The basis B_i specializes to the reduced Gröbner basis of the ideal on every point $\overline{a}_0 \in S_i$, and have constant set of leading power products (lpp) on the whole segment.

Its canonical character is proved in [16]. In [9] we present an algorithm to compute it, implemented in Singular in the library grobcov.lib, that can be downloaded from the Singular web [12].

The algorithm has the following steps:

(1) Homogenize the ideal;
(2) Compute a disjoint CGS (Comprehensive Gröbner System) with constant lpp's;
(3) Group the segments by lpp (leading power products) of the reduced Gröbner basis;
(4) Dehomogenize and reduce the bases;
(5) Compute the homogenized-lpp-segments grouped in 3) by adding together all the segments with the same lpp;
(6) Compute the generic (and the extended) basis for the lpp segments.

Step 2) computes a CGS, which needs to contain disjoint segments with constant lpp's, and must be described in a canonical form in order to be able to be added together with the other segments having the same lpp. By Wibmer's Theorem ([16]) its union is locally closed and allows a unique basis $B_i \subset \mathcal{O}_{S_i}[X]$[1]. In several papers [6, 5], and more precisely in [9], we present an algorithm (Buildtree) for this purpose.

Other people have proposed algorithms to compute a CGS. We mention [14, 15], who initiated the research field. Then [13] and [10] gave more efficient algorithms. But these algorithms do not have the required properties to be used in the canonical description.

Recently [3] have proposed a very efficient algorithm that moreover produces disjoint segments with constant lpp's. Thus it is easy to adapt it to be used as the first step of the Gröbner Cover algorithm. We only need to transform the description of the segments into canonical form. We have implemented this procedure into a new version of the grobcov library, that is not yet published, increasing considerably the efficiency of the GC implementation.

We tested many examples, and give here some results for problems of medium-high difficulty. Time is given in seconds using an Intel(R)Core(TM)2 Duo CPU / T7500 @ 2.20 GHz / 2.19 GHz, 2.00 GB Ram computer.

Performance of the Gröbner Cover algorithm using KSW / Buildtree inside

	Using Buildree		Using KSW		GrobCov	Time ratio
System	Time	Segments	Time	Segments	Segments	KSW/BT
S10	2.64	6	0.45	4	3	0.17
S11	4.79	6	5.86	23	6	1.22
S15	89.54	93	3.45	31	22	0.04
S16	2.64	36	0.72	16	5	0.27
S47	21.21	18	0.72	9	7	0.03
S53	2.62	9	0.55	7	5	0.21
S54	5.43	23	0.86	2	2	0.16
S58	10.14	88	2.48	31	12	0.24
S59	4.98	66	2.24	25	12	0.45

The examples are taken from real applications or from test problems with special difficulties. S_{10} discusses the inverse kinematic problem of a simple robot arm [6]. S_{11} comes also from robotics [1], as well as S_{15} [2]. S_{16} studies a tensegrity problem [11]. S_{47} is the automatic proof of the nine points circle theorem. S_{53} is the problem of automatically discover the conditions for having an isosceles orthic triangle. S_{54} is also an automatic discovering theorem problem, proposed in the pastimes section of the French journal "Le Monde" on the printed edition of Jan. 8, 2007 as a problem of two skaters. Both problems S_{53} and S_{54} are discussed in [7]. S_{58} and S_{59} are problems where bases containing I-regular functions described by multiple polynomials are expected and was proposed by the author.

[1] $\mathcal{O}_{S_i}[X] : S_i \to \overline{K}[x]$ is a set of I-regular functions over S_i, and can be represented in some way by polynomials of $K[\overline{a}][\overline{x}]$.

The parametric ideals are the following:

$$
\begin{array}{rcl}
S_{10} & = & s_1s_2l - c_1c_2l - c_1 + (r), -s_1c_2l - s_1 - c_1s_2l + (z), s_1^2 + c_1^2 - 1, \\
& & s_2^2 + c_2^2 - 1; \\
S_{11} & = & (rd_3d_4 - r + Z - r_2^2d_3d_4 + r_2^2 - d_3^3d_4 + d_3^2d_4^2 - d_3d_4^3 + d_3d_4)t^4 \\
& & +(-2rr_2d_4 + 2r_2^3d_4 + 2r_2d_3^2d_4 - 4r_2d_3d_4^2 + 2r_2d_4^3 + 2r_2d_4)t^3 \\
& & +(-2r + 2Z + 4r_2^2d_4^2 + 2r_2^2 - 2d_3^2d_4^2 + 4d_4^2)t^2 \\
& & +(-2rr_2d_4 + 2r_2^3d_4 + 2r_2d_3^2d_4 + 4r_2d_3d_4^2 + 2r_2d_4^3 + 2r_2d_4)t \\
& & +(-rd_3d_4 - r + Z + r_2^2d_3d_4 + r_2^2 + d_3^3d_4 + d_3^2d_4^2 + d_3d_4^3 - d_3d_4); \\
S_{15} & = & (d)s_1 + (a), (-d)c_1 + (b), (l_3)c_3 + (l_2)c_2 + (-d), (l_3)s_3 + (l_2)s_2 + (-c), \\
& & c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, c_3^2 + s_3^2 - 1; \\
S_{16} & = & (x - y - z)w_5, (z)w_4 + (-z)w_5, (x + y - 1)w_4 + (-z + 1)w_5, \\
& & (y - 1)w_3 + (y)w_4 + (-2z + 1)w_5, (x - z)w_3 + (-y)w_4, \\
& & (z - 1)w_2 + (z)w_3 + (2z - 1)w_5, (y - 1)w_2 + (y + 2z - 1)w_5, \\
& & (x - 1)w_2 + (z)w_3 + (y + 2z - 1)w_5; \\
S_{47} & = & x_0^2 + y_0^2 - r_2, x_0^2 + y_0^2 + (-2a - 2)x_0 + (-2b)y_0 - r_2 + (a^2 + 2a + b^2 + 1), \\
& & x_0^2 + y_0^2 + (-2a + 2)x_0 + (-2b)y_0 - r_2 + (a^2 - 2a + b^2 + 1), \\
& & x_0^2 + y_0^2 + (-4a)x_0 - r_2 + (4a^2), \\
& & x_1^2 + y_1^2 - 2x_1x_0 + x_0^2 - 2y_1y_0 + y_0^2 + (2a)x_1 + (2b)y_1 + (-2a)x_0 \\
& & +(-2b)y_0 - r_2 + (a^2 + b^2), x_1 + (-a), (a + 1)x_1 + (b)y_1 + (-a - 1), \\
& & x_1^2 + y_1^2 - 2x_1x_0 + x_0^2 - 2y_1y_0 + y_0^2 + 2x_1 - 2x_0 - r_2 + 1, \\
& & x_1^2 + y_1^2 - 2x_1x_0 + x_0^2 - 2y_1y_0 + y_0^2 - 2x_1 + 2x_0 - r_2 + 1, \\
& & (b)x_2 + (-a - 1)y_2 + (2b), (a + 1)x_2 + (b)y_2 + (-2a{-}_2), \\
& & x_2^2 + y_2^2 - 2x_2x_0 + x_0^2 - 2y_2y_0 + y_0^2 - r_2, \\
& & (b)x_3 + (-a + 1)y_3 + (-2b), (a - 1)x_3 + (b)y_3 + (2a - 2), \\
& & x_3^2 + y_3^2 - 2x_3x_0 + x_0^2 - 2y_3y_0 + y_0^2 - r2; \\
S_{53} & = & (-b)x_2 + (a - 1)y_2 + (b), (a - 1)x_2 + (b)y_2 + (a - 1), \\
& & (b)x_3 + (-a - 1)y_3 + (b), (a + 1)x_3 + (b)y_3 + (-a - 1), \\
& & -x_2^2 + x_3^2 - y_2^2 + y_3^2 + (2a)x_2 + (-2a)x_3; \\
S_{54} & = & x_1^2 + y_1^2 + (-2a)x_1 - 2y_1, x_2^2 + y_2^2 + (2b)x_2 - 2y_2, \\
& & (a)x_1 + y_1 + (a^2c_v - a^2 + c_v - 1), (-b)x_2 + y_2 + (b^2c_w - b^2 + c_w - 1), \\
& & -x_1 + (a)y_1 + (a^2s_v + s_v), -x_2 + (-b)y_2 + (b^2s_w + s_w), \\
& & -y_1x_2 + x_1y_2 - 2x_1 + 2x_2;\ a^2 + 1 \neq 0, b^2 + 1 \neq 0, a + b \neq 0; \\
S_{58} & = & (a_0)x^2 + (b_0)y + (c_0), (a_1)x^2 + (b_1)y + (c_1); \\
S_{59} & = & (a_0)x^2 + (b_0)xy + (c_0)y^2, (a_1)x^2 + (b_1)xy + (c_1)y^2, (a_2)x + (b_2)y;
\end{array}
$$

1. Conclusions

It can be observed that, in problems of medium-high difficulty, in general the performance of the Gröbner Cover algorithm using KSW is much better than with Buildtree. The speed increases up to 30 times in the best cases even if there are particular problems for which it does not gain anything. One of the reasons of the better performance lies not only in the speed of the KSW algorithm but also in the fact that, in general, KSW produces less segments than Buildtree, and this makes the remaining parts of the algorithm to be less expensive.

For problems of small difficulty there are no particular differences between both methods, and even Buildtree can be more efficient, but this is not significative. It can be observed that the efficiency increases considerably when the number of segments of the CGS is smaller for KSW than with Buildtree.

Another interesting conclusion we can derive from the analysis of the performance of both methods is about the canonicity of the Gröbner Cover algorithm. Even if the CGS computation produces very different results in both procedures, the output of the Gröbner Cover algorithm after steps 3), 4), 5), 6) becomes always the same (at least as far as we have verified).

References

[1] M.Coste. Classifying serial manipulators: Computer Algebra and Geometric Insight. Plenary Talk. Proceedings of EACA-2004, (2004), 323–323.
[2] M.J. González-López, T. Recio. The ROMIN inverse geometric model and the dynamic evaluation method. In: Computer Algebra in Industry, A.M. Cohen ed., John Wiley & Sons, (1993), 117–141.
[3] D. Kapur, Y. Sun, and D.K. Wang. A New Algorithm for Computing Comprehensive Gröbner Systems. Proceedings of ISSAC'2010, ACM Press, (2010), 29–36.
[4] D. Kapur, Y. Sun, and D.K. Wang. Computing Comprehensive Gröbner Systems and Comprehensive Gröbner Bases Simultaneously. Proceedings of ISSAC'2011, ACM Press, (2011), 193–200.
[5] M. Manubens, A. Montes, Minimal Canonical Comprehensive Gröbner Systems, *Jour. Symb. Comp.*, **44**:5, (2009), 463–478.
[6] A. Montes. New Algorithm for Discussing Gröbner Bases with Parameters. *Jour. Symb. Comp.* **33**:1-2 (2002), 183–208.
[7] A. Montes, T. Recio, Automatic discovery of geometry theorems using minimal canonical comprehensive Groebner systems. Proceedings of ADG 2006, L.N.A.I., Springer, **4869**, (2007), 113–138.
[8] http://www-ma2.upc.edu/∼montes/ download a beta version of the software grobcov.lib. (2011).
[9] A. Montes, M. Wibmer. Gröbner Bases for Polynomial Systems with Parameters. *Jour. Symb. Comp.*, **45**, (2010), 1391–1425.
[10] K. Nabeshima. A speed-up of the Algorithm for Computing Comprehensive Gröbner Systems. Proceedings of ISSAC'2007, ACM Press, (2007), 299–306.
[11] D. Orden, M. de Guzmán. Finding tensegrity structures: geometric and symbolic approaches. Proceedings of EACA-2004, (2004), 167–172.
[12] http://www.singular.uni-kl.de
[13] A. Suzuki and Y. Sato. A simple Algorithm to Compute Comprehensive Gröbner Bases using Gröbner Bases. Proceedings of ISSAC'2006, ACM Press, (2006), 326–331.
[14] V. Weispfenning. Comprehensive Gröbner Bases. *Jour. Symb. Comp.* **14**, (1992), 1–29.
[15] V. Weispfenning. Canonical Comprehensive Gröbner Bases. *Jour. Symb. Comp.* **36**, (2003), 669–683.
[16] M. Wibmer, Gröbner Bases for Families of Affine or Projective Schemes. *Jour. Symb. Comp.*, **42**:8 (2007), 803–834.

Universitat Politècnica de Catalunya
E-mail address: antonio.montes@upc.edu