Computation of the Minimal Associated Primes

Santiago Laplagne

Departamento de Matemática, Universidad de Buenos Aires
Buenos Aires, Argentina
slaplagn@dm.uba.ar

Abstract. We propose a new algorithm for the computation of the minimal associated primes of an ideal. In [1] we have introduced modifications to an algorithm for the computation of the radical by Krick and Logar ([2]) (based on ideas by Gianni, Trager and Zacharias ([3])), that made it more efficient. In this work, we show how these same modifications can be applied to the algorithm for the computation of the minimal associated primes proposed in [3]. We explain the algorithm, our modifications and show some benchmarks that confirm that the new algorithm is more efficient than the original one.

Keywords. Minimal associated primes, polynomial ideal, Gröbner bases, primary decomposition, radical

1 Introduction

Solving systems of polynomial equations is a main task in Computer Algebra, although the precise meaning of what is an acceptable solution depends on the context. In this work, we interpret it as finding the minimal associated primes of the ideal generated by the polynomials. Geometrically, this is equivalent to decompose the set of solutions into its irreducible components.

A common technique is to reduce the problem to the zero dimensional case. In a paper by Gianni, Trager and Zacharias they use this technique, combined with the splitting tool $I = (I : h^\infty) \cap \langle I, h^m \rangle$ for some specific polynomial h and integer m. This splitting introduces a number of redundant components that are not part of the original ideal.

Their ideas can be used to compute the primary decomposition, the radical and the minimal associated primes. In [2], the authors use these ideas to compute the radical of an ideal. In [1] we propose some modifications to that algorithm. We use the reduction to the zero dimensional case, but we avoid working with the ideal $\langle I, h^m \rangle$ using instead saturations with respect to appropriate polynomials. As a result, when the ideal has components of different dimensions, our algorithm is usually more efficient.

In the present work we show how the same modifications can be applied to the computation of the minimal associated primes of an ideal. We make a brief description of GTZ algorithm, we introduce our modifications and we show some time comparisons using an implementation in Singular [4].

Dagstuhl Seminar Proceedings 06271
Challenges in Symbolic Computation Software
http://drops.dagstuhl.de/opus/volltexte/2006/774

2 Preliminaries

We note $\mathbf{V}_k(I)$ for the vanishing set of I in k^n and $\bar{k}$ for the algebraic closure of k. An ideal is called zero dimensional if $\mathbf{V}_{\bar{k}}(I)$ has only a finite number of points. This is equivalent to saying that it contains polynomials pure in each variable. If I is zero-dimensional, we say that a variable x_i separate the points of I if the results of evaluating the polynomial x_i in each of the points of $\mathbf{V}_{\bar{k}}(I)$ are all different.

Following the ideas in [3], the computation of the minimal associated primes of a general ideal can be reduced to the zero dimensional case. For the computation of the minimal associated primes of a zero dimensional ideals, the following algorithm, also based in an algorithm proposed in [3], can be used.

Proposition 1 *Let $\langle g \rangle = I \cap k[x_n]$ and $g = g_1^{m_1} \dots g_t^{m_t}$, the factorization. Then*

$$I = \cap_{i=1}^{t} \langle I, g_i^{m_i} \rangle.$$

If x_n separate points, then

- *$\langle I, g_i^{m_i} \rangle$ is primary*
- *$\sqrt{\langle I, g_i^{m_i} \rangle} = \langle I, g_i \rangle$, and these are the minimal associated primes.*

In [5] [Criterion 4.2.4], an algorithm is given for checking if x_n separates variables, by looking at the shape that the ideals $\langle I, g_i^{m_i} \rangle$ must have in that case.

If x_n does not separate variables, a random coordinate change must be performed. If k is infinite a suitable coordinate change always exist.

In [3], the authors use the splitting tool $I = (I : h^\infty) \cap \langle I, h \rangle$ (for h such that $I : h = I : h^2$). They find h such that the minimal associated primes of $I : h$ can be obtained by reduction to the zero-dimensional case and the ones corresponding to $\langle I, h \rangle$ can be obtained by induction.

When taking $\langle I, h \rangle$ there appear redundant components (that is, components that were not part of the original ideal) that slow down the algorithm performance.

In the algorithm that we proposed in [1] for computing the radical of an ideal, we avoided using $\langle I, h \rangle$ and instead we used repeatedly the saturation $I : h^\infty$ for appropriate h. This leaded in some cases to a more efficient algorithm.

The same ideas can be used for computing the minimal associated primes of an ideal, obtaining the following algorithm

Algorithm 2 minAssPrimes*(I)*
Input: $I \subset k[\boldsymbol{x}]$
Output: $P_1, \dots, P_t$, *the minimal assocciated primes of* I.

1. $\tilde{P} \leftarrow \langle 1 \rangle$ *($\tilde{P}$ will be the intersection of the minimal associated primes already obtained).*
2. *Repeat*

(a) *Look for* $g \in \tilde{P} \setminus \sqrt{I}$. *To find it, search over the generators of* $\tilde{P}$ *and check if they are in* $\sqrt{I}$.
(b) *If there does not exist such* g, *it means that* $\tilde{P} \subset \sqrt{I}$. *Since we always have* $\sqrt{I} \subset \tilde{P}$, *we conclude that* $\tilde{P} = \sqrt{I}$. *Exit the cycle.*
(c) *If there exists* $g \in \tilde{P} \setminus \sqrt{I}$, *this means that there exists at least one minimal prime* P *associated to* I *such that* $g \notin P$.
$J \leftarrow I : g^{\infty}$.
(d) *Reduction to the zero-dimensional case:*
Take a maximal independent set $\boldsymbol{u}$ *with respect to* J *and compute* $P'_1, \ldots, P'_s$, *the minimal associated primes of the zero-dimensional ideal* $Jk(\boldsymbol{u})[\boldsymbol{x} \setminus \boldsymbol{u}]$.
(e) *Contract the ideals* $P'_i \subset k(\boldsymbol{u})[\boldsymbol{x} \setminus \boldsymbol{u}]$ *to* $P_i \subset k[\boldsymbol{x}]$, $1 \leq i \leq s$.
(f) $\tilde{P} \leftarrow \tilde{P} \cap P_1 \cap \cdots \cap P_s$.
(g) $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_1, \ldots, P_s\}$.

3. *output* $= \mathcal{P}$, *the minimal associated primes of* I.

The correctness and termination of the algorithm can be proven in exactly the same way as in [1].

Remark 1. As in [1], in this algorithm there is no redundancy. All the ideals that we add to $\mathcal{P}$ are minimal prime ideals associated to I.

As an example, we apply the algorithm to the ideal

$$I = \langle y + z, xz^2w, x^2z^2 \rangle \subset \mathbb{Q}[x, y, z, w].$$

In the first iteration, we take $g := 1$ and $J := I : 1^{\infty} = I$. We find that $\boldsymbol{u} = \{x, w\}$ is a maximal independent set with respect to J. Making the reduction step, we obtain that the only minimal associated primes of $J(\boldsymbol{u})[\boldsymbol{x} \setminus \boldsymbol{u}]$ is $\langle y, z \rangle$, which contracted to $k[\boldsymbol{x}]$ is $P_1 = \langle y, z \rangle$. We take $\tilde{P} := P_1$ and $\mathcal{P} := \{P_1\}$.

In the second iteration, we look for $g \in \tilde{P}$ such that $g \notin \sqrt{I}$. We obtain that $z \notin \sqrt{I}$ and compute $J = I : z^{\infty} = \langle y + z, xw, x^2 \rangle$. Now $\boldsymbol{u} = \{z, w\}$ is a maximal independent set with respect to J. The only minimal associated prime of $Jk(\boldsymbol{u})[\boldsymbol{x} \setminus \boldsymbol{u}]$ is $\langle y + z, x \rangle$, which contracted to $k[\boldsymbol{x}]$ gives $P_1 = \langle y + z, x \rangle$. We take $\tilde{P} := \langle y, z \rangle \cap \langle y + z, x \rangle = \langle y + z, xz \rangle$ and $\mathcal{P} = \{\langle y, z \rangle, \langle y + z, x \rangle\}$.

If we search for $g \in \tilde{P}$ such that $g \notin \sqrt{I}$, we obtain that $y + z$ and xz are both in $\sqrt{I}$. Therefore, the algorithm terminates. We obtain that the minimal associated primes of I are $\langle y, z \rangle$ and $\langle y + z, xz \rangle$.

We now apply GTZ algorithm ([3]) to the same ideal, to compare it with ours. We start with $I = \langle y + z, x\,z^2w, x^2z^2 \rangle$. The first step is the same, we obtain $P_1 = \langle y, z \rangle$ $\tilde{P} := P_1$ and $\mathcal{P} = \{P_1\}$.

The next step is different. We look for h such that $I = (I(\boldsymbol{u})[\boldsymbol{x} \setminus \boldsymbol{u}] \cap k[\boldsymbol{x}]) \cap \langle I, h \rangle$. We can take $h = xz$. Now, $\sqrt{I} = \langle y, z \rangle \cap \sqrt{\langle I, xz \rangle}$. So it remains to compute the minimal associated primes of $\langle I, xz \rangle$. Carrying on the algorithm, we get that they are $\langle y + z, x \rangle$ and $\langle w, y, z \rangle$.

The last prime is not a minimal associated prime of I (not even an associated prime of I). It is a new component that appeared when we added xz to I.

This is a situation that repeats often in the examples. The polynomials that the algorithm adds to I make it more and more complex. The polynomials added are usually large, since they are the product of coefficients of polynomials in a Gröbner basis and the size of the Gröbner basis of the new ideal can increase drastically.

This does not happen in our proposed algorithm. We compute instead the saturation with respect to polynomials that are usually simple, and this saturation does not increase the complexity of the ideal since it only takes some components away from it. No new components can appear.

3 Performance evaluation

In this section, we apply the proposed algorithm to several examples given in [6], [7] and other ideals and evaluate its performance. (We only consider those ideals that are not zero dimensional.) We implemented the algorithm in Singular ([4]). Our routine uses the subroutine for the reduction to the zero dimensional case that is already implemented in the library `primdec` [8] for the computation of the Minimal Associated primes by Gianni-Trager-Zacharias algorithm. We compare the times obtained by our algorithm with the algorithms implemented in `primdec`: Gianni-Trager-Zacharias ([3]) and via Characteristic Sets (`proc minAssChar`).

We created some new examples where the differences are more significant, which we detail below.

$p_1 = a+c+d+e+f+g+h+j-1, p_2 = -b+c+e+g+j, q_1 = 59ad+59ah+59dh-705d-1199h, q_2 = -54acf-54adf+a+d, q_3 = adfg+a+d$
$I_1 = \langle p_1, p_2\rangle \cap \langle q_1, q_2, q_3\rangle$ (polynomials taken from DGP25 and DGP28)
$p_1 = x^2+y^2+z^2-t^2, p_2 = xy+z^2-1, q_1 = w^2xy+w^2xz+w^2z^2, q_2 = tx^2y+x^2yz+x^2z^2, q_3 = twy^2+ty^2z+y^2z^2, q_4 = t^2wx+t^2wz+t^2z^2$
$I_2 = \langle p_1, p_2\rangle \cap \langle q_2, q_3, q_4\rangle, I_3 = \langle p_1, p_2\rangle \cap \langle q_1, q_3, q_4\rangle$ (polynomials taken from DGP31 and DGP32)

The results are shown in Table 1. All the computations are done over $\mathbb{Q}$. The ordering of the monomials is always the degree reverse lexicographical ordering with the underlying ordering of the alphabet.

The codes for the examples in the firsts columns are the ones given in [6] and [7]. "Dim" indicates the dimension of the ideal; "Prim. comps.", the total number of primary components; "Min. ass.", the number of minimal associated primes; "Emb. comps.", the number of embedded components and "Equidim?" if the ideal is equidimensional. The last three columns show the timings. GTZ is the algorithm of Gianni, Trager and Zacharias ([3]) and Char is an algorithm using characteristic sets implemented in Singular. Timing is measured in hundredths of seconds. The entry * means that after one day of computations, the algorithm did not terminate.

In the implementation of GTZ in Singular, the original ideal is first decomposed using factorizing Gröbner bases algorithm and then the minimal associated

Table 1. Timing results

Source	Code	Dim	Prim. comps	Min. ass.	Emb. comps	Equidim?	this paper	GTZ	Char
DGP	1	3	4	4	0	Yes	39	37	1037
DGP	2	3	16	15	1	No	57	40	86
DGP	3	2	11	4	7	No	6	4	2
DGP	4	6	4	3	1	No	18	17	14
DGP	7	3	6	6	0	Yes	26	20	76
DGP	14	1	8	2	6	No	9	7	5
DGP	20	4	2	1	1	No	15	14	3185
DGP	21	9	9	1	8	No	3	2	1
DGP	22	2	9	7	2	No	33	25	370
DGP	23	2	18	12	6	No	91	71	22750
DGP	24	8	6	5	1	No	14	9	12
DGP	25	5	7	5	2	No	101	81	1615
DGP	27	4	3	3	0	Yes	13	9	11
DGP	28	7	2	2	0	Yes	30	27	18
DGP	29	2	12	1	11	No	4	2	9
DGP	30	1	14	14	0	Yes	283	259	12145
DGP	31	1	1	1	0	Yes	10	10	3
DGP	32	2	17	8	9	No	21	15	34
DGP	33	2	3	3	0	No	10	8	5
CCT	M	5	3	3	0	No	58	48	2268
CCT	83	5	3	3	0	No	133	603	98
CCT	O	2	5	5	0	Yes	26	209	3
New	1	9	4	4	0	No	281	*	2383
New	2	3	11	8	3	No	120	*	32065
New	3	3	11	8	3	No	69	*	27088

primes of each component are computed. We do the same decomposition in our algorithm.

We see that for time consuming computations, our proposed algorithm is always faster than GTZ algorithm.

4 Acknowledgements

The author thanks Teresa Krick for her guidance in this work, and Gert-Martin Greuel, Gerhard Pfister and Hans Schönemann for all their help during my stay at the University of Kaiserslautern.

References

1. Laplagne, S.: An algorithm for the computation of the radical of an ideal. In: ISSAC '06: Proceedings of the 2006 international symposium on Symbolic and algebraic computation, New York, NY, USA, ACM Press (2006) 191–195

2. Krick, T., Logar, A.: An algorithm for the computation of the radical of an ideal in the ring of polynomials. AAECC9, Springer LNCS (1991) 195–205
3. Gianni, P., Trager, B., Zacharias, G.: Bases and primary decomposition of ideals. J. Symbolic Computation (1988) 149–167
4. Greuel, G.M., Pfister, G., Schonemann, H.: SINGULAR 3.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern (2005) `http://www.singular.uni-kl.de`.
5. Greuel, G.M., Pfister, G.: A Singular Introduction to Commutative Algebra. Springer (2002)
6. Decker, W., Gruel, G.M., Pfister, G.: Primary decomposition: Algorithms and comparisons. Algorithmic algebra and number theory, Springer Verlag, Heidelberg (1998) 187–220
7. Caboara, M., Conti, P., Traverso, C.: Yet another algorithm for ideal decomposition. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (1997) 39–54
8. Decker, W., Pfister, G., Schoenemann, H.: `primdec.lib`. A SINGULAR 3.0 library for computing primary decomposition and radical of ideals (2005)