Algorithms for Computing Minimal Associated Primes of Polynomial Ideals

Aoyama, Toru

(Degree)
博士（理学）

(Date of Degree)
2018-03-25

(Date of Publication)
2019-03-01

(Resource Type)
doctoral thesis

(Report Number)
甲第7117号

(URL)
https://hdl.handle.net/20.500.14094/D1007117

博　士　論　文

Algorithms for Computing Minimal Associated Primes of Polynomial Ideals

(多項式イデアルの極小付属素イデアル計算アルゴリズム)

平成 30 年 1 月

神戸大学大学院理学研究科

Toru Aoyama

青山　暢

Contents

Abstract

This paper proposes two algorithms for computing minimal associated primes of ideals in polynomial rings over a field.

The first one is an algorithm designed for binomial ideals. It utilizes the cellular decomposition as an intermediate decomposition. It is defined by Eisenbud-Sturmfels and improved by Kahle. In addition, following some parts of the algorithm by Laplagne, a new algorithm for an intermediate decomposition is constructed. This algorithm decomposes an ideal into cellular ideals whose sets of minimal associated primes are disjoint. It needs neither extensions of the coefficient field nor reductions to the zero-dimensional case. Most of the computations are saturations. We observe by this intermediate decomposition, binomial ideals are decomposed into components whose radicals correspond to the minimal associated primes in many cases. This algorithm executes nilpotency checks, radical membership tests and computations of saturations many times. Therefore, we try to speed up the check of $I = I : f$ (f is a polynomial) which is necessary for above computations. As a result, we obtain efficient algorithms including heuristic and optional methods.

The second one applies Chinese Remainder Theorem (CRT) to Laplagne's algorithm which computes minimal associated primes without producing redundant components. CRT reconstructs an object in a ring from its modular images in the quotient rings modulo some ideals. In Laplagne's algorithm, ideals are decomposed over rational function fields over $\mathbb{Q}$ by regarding some variables as parameters. In our new algorithm, we compute the minimal associated primes of $\langle \phi(G) \rangle$ for a given ideal $I = \langle G \rangle$, where ϕ is a substitution map for a parameter. Then we construct candidates of the minimal associated primes of I by applying CRT for those of $\langle \phi(G) \rangle$'s. In order for this method to work correctly, the shape of each modular component must coincide with that of the corresponding component of the ideal. This is realized with a high probability because a multivariate irreducible polynomial over $\mathbb{Q}$ remains irreducible after a substitution of integers for variables with a high probability.

Chapter 1

Basic Facts

In this chapter, we recall several well-known facts which are bases of this paper. We just list definitions and facts that is concerned with this paper concisely. For more details and proofs, refer to [AM, Chapter 1], [GP, Chapter 1] and [GG, Chapter 21].

1.1 Polynomial Ring

Definition 1.1.1. Let R be a commutative ring and $x_1, \ldots, x_n$ variables.

1) A **monomial** is a power product of variables

$$x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}, (\alpha = (\alpha_1, \ldots, \alpha_n) \in {\mathbb{Z}_{\geq 0}}^n).$$

2) A **term** over R is a product of monomial and an element of R

$$rx^\alpha, (r \in R, \alpha \in \mathbb{Z}_{\geq 0}^n).$$

In this case, r is called the **coefficient** of a term rx^α.

3) A **polynomial** over R is a finite sum of terms

$$\sum_{i=1}^{m} r_i x^{\alpha_i}, (m \in \mathbb{Z}_{\geq 0}, r_i \in R, \alpha_i \in {\mathbb{Z}_{\geq 0}}^n).$$

4) The polynomial ring over R is the set of all polynomials over R

$$R[X] := R[x_1, \cdots x_n] := \left\{ \sum_{i=1}^{m} r_i x^{\alpha_i} \;\middle|\; m \in \mathbb{Z}_{\geq 0}, r_i \in R, \alpha_i \in {\mathbb{Z}_{\geq 0}}^n \right\}.$$

A polynomial ring is a commutative ring with the usual addition and multiplication. We list some notions and operations on ideals.

Definition 1.1.2. Let I, J be ideals in $R[X]$, f, g polynomials in $R[X]$ and a a member of $R[X]$.

- I is a **prime ideal** if $fg \in I$ implies $f \in I$ or $g \in I$.
- The **quotient ideal** (**colon ideal**) of I with respect to J

$$I : J := \{ f \in R[X] \mid fJ \subset I \}.$$

  In particular

$$I : a := I : \langle a \rangle = \{ f \in R[X] \mid fa \in I \}.$$

- The **saturation** of I with respect to J

$$I : J^{\infty} := \{ f \in R[X] \mid m \in \mathbb{N} \text{ exists s.t } fJ^m \subset I \}.$$

  In particular

$$I : a^{\infty} := I : \langle a \rangle^{\infty} = \{ f \in R[X] \mid m \in \mathbb{N} \text{ exists s.t } fa^m \in I \}.$$

- The **radical** of I

$$\sqrt{I} := \{ f \in R[X] \mid m \in \mathbb{N} \text{ exists s.t. } f^m \in I \}.$$

- I is a **radical ideal** if $\sqrt{I} = I$.
- I is a **primary ideal** if $fg \in I$ and $f \notin I$ imply $g \in \sqrt{I}$.

For ideals I, J in $R[X]$, $I : J$, $I : J^{\infty}$ and $\sqrt{I}$ are also ideals.

Definition 1.1.3. A ring R is called a **Noetherian** ring if every ideal in R is generated by a finite set.

We utilize properties of Noetherian rings to ensure the termination of algorithms implemented in polynomial rings.

Proposition 1.1.4. Let R be a commutative ring. The following are equivalent.

1) R is Noetherian.
2) (**ascending chain condition**) For every ascending chain of ideals in R

$$I_1 \subset I_2 \subset \cdots,$$

   there exists $s \in \mathbb{N}$ such that if $s \leq i$, then $I_s = I_i$.

Theorem 1.1.5. (**Hilbert's basis theorem**) If R is a Noetherian ring, then $R[X]$ is a Noetherian ring.

1.2 Monomial Orderings

It is important to determine the ordering on monomials when we compute over polynomial rings. Computing processes are determined following a fixed **monomial ordering**.

Definition 1.2.1. Let $<$ be a total ordering on monomials. $<$ is called monomial ordering when it satisfies the following properties.

1) $1 \leq m$ for any monomials m.

2) Let m_1, m_2, m_3 be monomials. If $m_1 < m_2$, then $m_1 m_3 < m_2 m_3$.

Unless otherwise noted, let $x_1 > \cdots > x_n$ for any monomial orderings.

Example 1.2.2. The following two orderings are monomial orderings.

- **Lexicographical ordering** $<_{lex}$
  $x^\alpha <_{lex} x^\beta \overset{\text{def}}{\iff}$ there exists i $(1 \leq i \leq n)$ such that $j < i \Longrightarrow \alpha_j = \beta_j$ and $\alpha_i < \beta_i$.

- **Graded reverse lexicographical ordering** $<_{grev}$
  $x^\alpha <_{grev} x^\beta \overset{\text{def}}{\iff} \deg(x^\alpha) < \deg(x^\beta)$ or $(\deg(x^\alpha) = \deg(x^\beta)$ and there exists i $(1 \leq i \leq n)$ such that $i < j \Longrightarrow \alpha_j = \beta_j$ and $\alpha_i < \beta_i)$.

Definition 1.2.3. Let $<$ be a monomial ordering , S a subset of $R[X]$ and $f = r_1 x^{\alpha_1} + \cdots + r_m x^{\alpha_m}$ a polynomial in $R[X]$ where $x^{\alpha_1} > \cdots > x^{\alpha_m}$ and $r_1, \ldots, r_m \neq 0$.

- The **leading monomial** of f $LM(f) := x^{\alpha_1}$.
- The **leading monomial set** of S $LM(S) := \{\, LM(f) \mid f \in S \,\}$.
- The **leading term** of f $LT(f) := r_1 x^{\alpha_1}$.
- The **leading coefficient** of f $LC(f) := r_1$.
- The **leading ideal** of S $L(S) := \langle \{\, LT(f) \mid f \in S \,\} \rangle$.

Utilizing these definitions, we can construct an algorithm for division on $\mathbb{K}[X]$ where $\mathbb{K}$ is a field.

Definition 1.2.4. Let $f, f_1, \ldots, f_m$ be polynomials in $\mathbb{K}[X]$ and $<$ a monomial order. $r \in \mathbb{K}[X]$ is called a **normal form** of f with respect to $\{f_1, \ldots, f_m\}$ if r satisfies the following.

1) $f = q_1 f_1 + \cdots + q_m f_m + r$, $(q_i \in \mathbb{K}[X])$.

2) $LM(q_i f_i) \leq LM(f)$.

3) No term in r is divisible by any $LT(f_i)$.

Note that the normal form of f is not unique in general. We show an example.

Example 1.2.5. Let $f = 4x + y + 2z, f_1 = 2x + y, f_2 = x + z$ in $\mathbb{Q}[x, y, z]$ and $<$ a monomial order with $x > y > z$.

$$\begin{aligned} f &= 2f_1 + 0 \cdot f_2 + (-y + 2z) \\ &= 0 \cdot f_1 + 4f_2 + (y - 2z) \\ &= f_1 + 2f_2 + 0 \end{aligned}$$

Therefore each of $(-y + 2z), (y - 2z)$ and 0 is a normal form of f with respect to $\{f_1, f_2\}$.

We can construct an algorithm for computing a normal form of a polynomial with respect to a set of polynomials and a monomial ordering (Algorithm 1).

Algorithm 1 NF

Input: $f \in \mathbb{K}[X]$, $S = \{s_1, \ldots, s_m\} \subset \mathbb{K}[X]$ and a monomial ordering $<$
Output: a normal form of f with respect to S and $<$

```
r ← 0, g ← f
while g ≠ 0 do
    if there exists s_i ∈ S s.t. LT(s_i)|LT(g) then
        g ← g − (LT(g)/LT(s_i)) f_i
    else
        r ← r + LT(g), g ← g − LT(g)
    end if
end while
return r
```

The output of this algorithm depends on the choice of i in the if block. We will obtain the uniqueness of normal forms by **Gröbner Bases** in Section 1.3.

1.3 Gröbner Basis

A Gröbner basis is a finite set of generators of an ideal. It has desirable properties and helpful to solve various algorithmic problems on polynomial rings.

Definition 1.3.1. Let $<$ be a monomial ordering and I an ideal in $R[X]$. A finite set $G \subset I$ is a Gröbner basis of I with respect to $<$ if $L(G) = L(I)$.

For an arbitrary ideal I in $\mathbb{K}[X]$, a Gröbner basis of I can be computed from generators of I by **Buchberger's algorithm** (Algorithm 2).

Definition 1.3.2. Let f, g be polynomials in $\mathbb{K}[X]$, $<$ a monomial ordering and m the least common multiple with respect to $LT(f)$ and $LT(g)$. The **S-polynomial** with respect to f, g is defined as

$$Spoly(f, g) := \frac{m}{LT(f)} f - \frac{m}{LT(g)} g.$$

Algorithm 2 Buchberger's algorithm

Input: $f_1, \ldots, f_2 \in \mathbb{K}[X]$ and a monomial order $<$
Output: a Gröbner basis of $I = \langle f_1, \ldots, f_m \rangle$ w.r.t. $<$
  $G \leftarrow \{f_1, \ldots, f_m\}$, $P \leftarrow \{ (f_i, f_j) \mid 1 \leq i < j \leq m \}$
  while $P \neq \emptyset$ **do**
    choose $(p, q) \in P$, $P \leftarrow P \setminus \{ (p, q) \}$
    $r \leftarrow \mathrm{NF}(Spoly(p, q), G, <)$
    if $r \neq 0$ **then**
      $P \leftarrow P \cup \{ (g, r) \mid g \in G \}$, $G \leftarrow G \cup \{r\}$
    end if
  end while
  return G

We can solve the following problems by utilizing Gröbner Bases. Let $f \in \mathbb{K}[X]$, I, J ideals in $\mathbb{K}[X]$, $<$ a monomial ordering, G a Gröbner basis of I with respect to $<$ and $t \notin X$ a new variable.

- $\mathrm{NF}(f, G, <)$ is determined uniquely depending on its arguments. We call f is G**-reduced** with respect to $<$ if $\mathrm{NF}(f, G, <) = f$.
- (**ideal membership problem**) $f \in I \iff \mathrm{NF}(f, G, <) = 0$
- (**elimination theorem**) Let Y be a set of variables, $X \cap Y = \emptyset$, K an ideal in $\mathbb{K}[X, Y]$, $<_{elim}$ a monomial order such that if $f \in \mathbb{K}[X, Y]$ and $LT(f) \in \mathbb{K}[Y]$, then $f \in \mathbb{K}[Y]$ and H a Gröbner basis of K with respect to $<_{elim}$. Then

  $$\{ h \in H \mid LT(h) \in \mathbb{K}[Y] \} \text{ is a Gröbner basis of } K \cap \mathbb{K}[Y].$$

  We call a monomial ordering $<_{elim}$ with $X >_{elim} Y$ a **elimination ordering**.
- (**intersection**) $I \cap J = \langle tI, (1 - t)J \rangle \cap \mathbb{K}[X]$.
- (**radical membership problem**) $f \in \sqrt{I} \iff \mathbb{K}[t]\langle I, (1 - tf) \rangle = \mathbb{K}[X, t]$.
- (**quotient**) If $I \cap \langle f \rangle = \langle f_1 f, \ldots, f_m f \rangle$, then $I : f = \langle f_1, \ldots, f_m \rangle$. Moreover, if $J = \langle g_1, \ldots, g_s \rangle$, then $I : J = \bigcap_{i=1}^{s} (I : g_i)$.
- (**saturation**) $I : g^\infty = \langle I, 1 - tg \rangle \cap \mathbb{K}[X]$. Moreover, if $J = \langle g_1, \ldots, g_s \rangle$, then $I : J^\infty = \bigcap_{i=1}^{s} (I : g_i^\infty)$.

In general, there are infinite Gröbner Bases for an ideal. Therefore we define the **reduced Gröbner basis** of an ideal to describe ideals uniquely.

Definition 1.3.3. Let G be a Gröbner basis of I with respect to $<$.

- G is **minimal** if for all $g \in G$

  1) g is monic,

  2) $LT(g) \notin L(G \setminus \{g\})$.

- A minimal Gröbner basis G is reduced if all $g \in G$ is $G \setminus \{g\}$-reduced with respect to $<$.

Theorem 1.3.4. Every ideal in $\mathbb{K}[X]$ has a unique reduced Gröbner basis with respect to a given monomial ordering. Furthermore it can be computed from any Gröbner basis of the ideal.

In the following of this paper, we omit to specify a monomial ordering unless it is necessary.

Chapter 2

Minimal Associated Primes

Our goal is to construct algorithms for representing a radical of a given ideal as an intersection of prime ideals. First of all, we verify the existence of such decompositions.

Definition 2.0.1. Let I be an ideal in a Noetherian ring. A prime ideal P including I is called a minimal associated prime of I if a prime ideal P' satisfies $I \subset P' \subset P$, then $P' = P$. $\mathrm{minAss}(I)$ denotes the set of all minimal associated primes of I.

Proposition 2.0.2. Let I be an ideal in a Noetherian ring. $\mathrm{minAss}(I)$ is finite and if $\mathrm{minAss}(I) = \{P_1, \ldots, P_m\}$, then

$$\sqrt{I} = P_1 \cap \cdots \cap P_m.$$

We call it the **prime decomposition** of $\sqrt{I}$

2.1 Zero-dimensional Decomposition

There is an algorithm for computing minimal associated primes of zero-dimensional ideals in $\mathbb{K}[X]$ with $char(\mathbb{K}) = 0$ by using the notion of **general position**.

Definition 2.1.1. ([GP, Definition 4.2.1])

1) A maximal ideal $M \subset \mathbb{K}[X]$ is called in general position with respect to $x_i \in X$, if there exist $g_1, \ldots, g_n \in \mathbb{K}[x_i]$ such that $\{x_1 + g_1(x_i), \ldots, x_{i-1} + g_{i-1}(x_i), x_{i+1} + g_{i+1}(x_i), \ldots, x_n + g_n(x_i), g_i(x_i)\}$ is the reduced Gröbner basis of M with respect to lexicographical ordering where x_i is smallest in X.

2) A zero-dimensional ideal $I \subset \mathbb{K}[X]$ is called in general position with respect to $x_i \in X$, if all associated primes $P_1, \ldots, P_m$ are in general position with respect to x_i and if $P_j \cap \mathbb{K}[x_i] \neq P_k \cap \mathbb{K}[x_i]$ for $j \neq k$.

For zero-dimensional ideals in general position, we obtain the next proposition.

Proposition 2.1.2. [GP, Proposition 4.2.3] Let I be a zero-dimensional ideal in $\mathbb{K}[X]$, $\langle g \rangle = I \cap \mathbb{K}[x_n]$ and $g = g_1^{m_1} \dots g_s^{m_s}$ the factorization of g. Then

$$I = \bigcap_{i=1}^{s} \langle I, g_i^{m_i} \rangle.$$

If I is in general position with respect to x_n, then $\langle I, g_i^{m_i} \rangle$ is a primary ideal for all i.

We can make a given zero-dimensional ideal in general position by coordinate changes.

Proposition 2.1.3. ([GP, Proposition 4.2.2]) Let $\mathbb{K}$ be a field of characteristic 0 and $I \subset \mathbb{K}[X]$ a zero-dimensional ideal. Then there exists a non-empty, Zariski open subset $Z \subset \mathbb{K}^{n-1}$ such that for all $\underline{a} = (a_1, \dots, a_{n-1}) \in Z$, the coordinate change $\varphi_{\underline{a}} : \mathbb{K}[X] \to \mathbb{K}[X]$ defined by $\varphi_{\underline{a}}(x_i) = x_i$ if $i < n$, and

$$\varphi_{\underline{a}}(x_n) = x_n + \sum_{i=1}^{n-1} a_i x_i$$

has the property that $\varphi_{\underline{a}}(I)$ is in general position with respect to x_n.

These coordinate changes are chosen randomly in the algorithm. We can decide whether an ideal is in general position or not by the following criterion.

Lemma 2.1.4. ([GP, Proposition 4.2.4]) Let I be an ideal in $\mathbb{K}[X]$. Then the following two conditions are equivalent.

1)
   - I is zero-dimensional.
   - I is in general position with respect to x_n.
   - I is a primary ideal.

2) Let S be the reduced Gröbner basis of I with respect to $<_{lex}$. Then there exist $g_1, \dots, g_n \in \mathbb{K}[x_n]$ and positive integer $m_1, \dots, m_n$ such that
   - $g_n^{m_n} \in S$ and g_n is irreducible.
   - $(x_j + g_j)^{m_j}$ is congruent to an element in $S \cap \mathbb{K}[x_j, \dots, x_n]$ modulo $\langle g_n, x_{n-1} + g_{n-1}, \dots, x_{j+1} + g_{j+1} \rangle \subset \mathbb{K}[X]$ for $i \leq j \leq n-1$.

Combining the above propositions, we can decide whether a zero-dimensional ideal is primary and in general position or not (Algorithm 3 [GP, Algorithm 4.2.5]).

Finally, we can construct an algorithm for computing minimal associated primes of zero-dimensional ideals (Algorithm 4). Algorithm 4 follows from [GP, Algorithm 4.2.7]. However our algorithm outputs only minAss(I).

Algorithm 3 PRIMARYTEST

Input: a zero-dimensional ideal $I = \langle f_1, \ldots, f_k \rangle \subset \mathbb{K}[X]$
Output: $\langle 0 \rangle$ if I is either not primary or not in general position,
or $\sqrt{I}$ if I is primary and in general position.

```
compute the reduced Gröbner basis G of I w.r.t <_lex
factorize g ∈ S, the element with smallest leading monomial
if g = g_n^{m_n} with g_n is irreducible then
    prim ← ⟨g_n⟩
else
    return ⟨0⟩
end if
i ← n
while i > 1 do
    i ← i − 1
    choose f ∈ S with LM(f) = x_i^t
    b ← the coefficient of x_i^{t−1} in f considered as polynomial in x_i
    q ← x_i + b/t
    if q^t ≡ f (mod prim) then
        prim ← prim + ⟨q⟩
    else
        return ⟨0⟩
    end if
end while
return prim
```

Algorithm 4 ZEROMINASS

Input: a zero-dimensional ideal $I = \langle f_1, \ldots, f_k \rangle \subset \mathbb{K}[X]$
Output: minAss(I)

```
result ← ∅
choose a random a ∈ K^{n−1} and I' ← φ_a(I) (cf. Proposition 2.1.3)
compute the reduced Gröbner basis G of I' w.r.t. <_lex
factorize g = g_1^{m_1} ... g_s^{m_s} ∈ G ∩ K[x_n]
for i = 1 to s do
    P'_i ← PRIMARYTEST(⟨I', g_i⟩)
    if P'_i ≠ ⟨0⟩ then
        P_i ← φ_a^{−1}(P'_i)
        result ← result ∪{P_i}
    else
        result ← result ∪ ZEROMINASS(⟨I, φ_a^{−1}(g_i)⟩)
    end if
end for
return result
```

2.2 Laplagne's Algorithm

We give a concise introduction of an algorithm for computing minimal associated primes by Laplagne in [L2]. We call it **Laplagne's algorithm** and each of our two algorithms is based on it. It makes a given ideal zero-dimensional and decomposes without producing redundant components.

Laplagne's algorithm is based on the following well known property.

Lemma 2.2.1. Let I be an ideal in $\mathbb{K}[X]$ and $\sqrt{I} = \bigcap_{i=1}^{m} P_i$ the prime decomposition. Then a polynomial $g \in \mathbb{K}[X]$ gives the prime decomposition $\sqrt{I : g^\infty} = \bigcap_{g \notin P_i} P_i$.

Proof. This is derived from [AM, Exercise 1.12 iv)]. □

In order to reduce to the zero-dimensional case, we utilize a **maximal independent set** of given ideals.

Definition 2.2.2. Let I be an ideal in $\mathbb{K}[X]$. $U \subset X$ is called an independent set of I if $I \cap \mathbb{K}[U] = \{0\}$. We say that an independent set U is maximal when $\#U = \dim(I)$.

For a set of variables Y, $\mathbb{K}(Y)$ denotes the set $\left\{ \frac{f}{g} \;\middle|\; f, g \in \mathbb{K}[Y], g \neq 0 \right\}$. If U is a maximal independent set of I, then $I\mathbb{K}(U)[X \setminus U]$ is zero-dimensional in $\mathbb{K}(U)[X \setminus U]$.

Lemma 2.2.1 implies the next proposition which is the core of Laplagne's algorithm.

Proposition 2.2.3. ([L1, Proposition 4])
Let I be an ideal in $\mathbb{K}[X]$, MA $\subset$ minAss(I) and Int $= \bigcap_{P \in MA} P$ (if MA $= \emptyset$, we define Int $= \langle 1 \rangle$). Suppose Int $\neq \sqrt{I}$, $g \in$ Int $\setminus \sqrt{I}$, $\sqrt{I : g^\infty} = \bigcap_{i=1}^{m} P_i$ is the prime decomposition and U is a maximal independent set of $I : g^\infty$. Then prime components such that $P_i \cap \mathbb{K}[U] = \{0\}$ satisfy $P_i \in$ minAss(I) and $P_i \notin$ MA.

We can compute prime components satisfying the above condition by the reduction to the zero-dimensional case.

Proposition 2.2.4. Let I be an ideal in $\mathbb{K}[X]$, U a maximal independent set of $I : g^\infty$ and the prime decomposition $\sqrt{I} = \bigcap_{i=1}^{m} P_i$ in the condition

$$P_i \cap \mathbb{K}[U] = \{0\} \;\; (1 \leq i \leq l), \quad P_i \cap \mathbb{K}[U] \neq \{0\} \;\; (l+1 \leq i \leq m).$$

Then we have the prime decomposition

$$\sqrt{I\mathbb{K}(U)[X \setminus U]} \cap \mathbb{K}[X] = \bigcap_{i=1}^{l} P_i \ .$$

Proof. See [GP] Exercise 4.3.3 and Proposition 4.3.1 (2). □

Laplagne's algorithm is constructed as follows (Algorithm 5).

Algorithm 5 LMinAss

Input: an ideal $I \subset \mathbb{K}[X]$
Output: minAss(I)

```
Int ← ⟨1⟩, MA ← ∅
while Int \√I ≠ ∅ do
    choose g ∈ Int \√I
    J ← I : g^∞
    U ← a maximal independent set of J
    J ← JK(U)[X \ U]
    {P_1, ..., P_m} ← zeroMinAss(J)
    PJ ← {P_1 ∩ K[X], ..., P_m ∩ K[X]}
    MA ← MA ∪PJ, Int ← Int ∩ ⋂_{P∈PJ} P
end while
return MA
```

Chapter 3

Prime Decompositions for Binomial Ideals

In this chapter, we propose a new algorithm for computing minimal associated primes of binomial ideals over polynomial rings. We have all of arguments over a polynomial ring $\mathbb{K}[x] = \mathbb{K}[x_1, \ldots, x_n]$ (over an arbitrary field $\mathbb{K}$). **Binomials** mean polynomials with at most two terms, namely, $am_1 + bm_2 (a, b \in \mathbb{K}, m_1, m_2$ are monomials in $\mathbb{K}[x])$. And we define a **binomial ideal** as an ideal generated by binomials.

The motivation of our research is to speed up the algorithm for primary decomposition by Kawazoe-Noro [KN]. It efficiently decomposes binomial ideals which have many embedded components. However it leaves place for improvement at the part computing minimal associated primes where ideals are made zero-dimensional.

Eisenbud-Sturmfels [ES] and Kahle [K] propose algorithms for computing minimal associated primes of binomial ideals and they are implemented in the computer algebra system Macaulay2 [M2]. The feature of these algorithms is that only binomials appear through the computing process. However, we have to extend the coefficient field in certain cases. On the other hand, Laplagne [L2] also proposes an algorithm for computing minimal associated primes of ideals (not limited to binomial ideals). It can decompose ideals without producing redundant components but needs the reduction to the zero-dimensional case. We combine both of their advantages.

Algorithms in [ES] and [K] represent a given ideal as an intersection of cellular ideals. It is called a cellular decomposition. It does not require reductions to zero-dimensional and most of the computations are saturations. We utilize and improve it as an intermediate decomposition.

In Section 3.1, we review the notion of cellular ideal and algorithms concerning cellular decomposition.

Our main results are in Section 3.2 and 3.3. We propose a new algorithm for an intermediate decomposition following a part of the process of Laplagne's

algorithm. And we show improvements of the new algorithm. It contains a subroutine transforming input, strategies choosing a polynomial for a saturation, techniques for saturations, an algorithm for homogeneous ideals and so on. We show the timing data of computing minimal associated primes of binomial ideals by algorithms in [K], [L2] and ours.

3.1 Cellular Decomposition

First we define cellular ideals.

Definition 3.1.1. An ideal I is called **cellular** if every $x_i (1 \leq i \leq n)$ is nilpotent or a non-zerodivisor modulo I.

Example 3.1.2. $I = \langle x^2, y-1 \rangle (\subset \mathbb{K}[x,y])$ is a cellular ideal. Actually x is nilpotent modulo I, y is a non-zerodivisor modulo I.
$I = \langle x^2, xy \rangle (\subset \mathbb{K}[x,y])$ is not a cellular ideal. Because y is a zerodivisor modulo I but not nilpotent modulo I.

We can decide whether an ideal is cellular or not by Algorithm 6 which is explained in [K, Algorithm 1 Step 1].

Algorithm 6 CELLCHECK

Input: an ideal $I \subset \mathbb{K}[X]$
Output: If I is cellular, then 1, otherwise 0.

```
X ← 1
for i = 1 to n do
    if I : x_i^∞ ≠ ⟨1⟩ then
        X ← X · x_i
    end if
end for
if I = I : X then
    return 1
else
    return 0
end if
```

Remark 3.1.3. In [K, Algorithm 1 Step 1], we check whether $I = I : X^\infty$ or not. However, it is equivalent to checking whether $I = I : X$ or not in Algorithm 6.

In this paper, we call a representation of an ideal as an intersection of cellular ideals a **cellular decomposition**.

By Algorithm 7, every ideal can be decomposed into cellular ideals. In the algorithm, the splitting tool is applied.

Proposition 3.1.4. (splitting tool)[GP, Lemma 3.3.6]
Let I be a ideal. If a polynomial a satisfies $I : a = I : a^2$, then

$$I = I : a \cap \langle I, a \rangle.$$

Algorithm 7 CELLDECOMP

Input: an ideal $I \subset \mathbb{K}[X]$
Output: a set of cellular ideals whose intersection is I
  if CELLCHECK$(I) = 1$ **then**
    return I
  end if
  choose x_0 : a zerodivisor modulo I among non-nilpotent variables modulo I
  $X \leftarrow {x_0}^m$(an integer m is chosen s.t. $I : X = I : X^2$)
  $C_1 \leftarrow I : X$
  $C_2 \leftarrow \langle I, X \rangle$
  return CELLDECOMP$(C_1) \cup$ CELLDECOMP(C_2)

Kahle proposes an algorithm for computing minimal associated primes in [K, Algorithm 4] . The algorithm decomposes cellular ideals into minimal associated primes after cellular decomposition. Here we give its brief outline. The proofs and details are in [K, Section 1].

For a set of indices of variables $\varepsilon \subset \{1, \ldots, n\}$ and a vector of natural numbers$d = (d_i)_{i \notin \varepsilon}$, we define

$$M(\varepsilon) := \langle x_i | i \notin \varepsilon \rangle, M(\varepsilon)^d := \langle {x_i}^{d_i} | i \notin \varepsilon \rangle.$$

Lemma 3.1.5. A binomial ideal I is cellular if and only if there exist a set $\varepsilon \subset \{1, \ldots, n\}$ and a vector of natural numbers $d = (d_i)_{i \notin \varepsilon}$ such that

$$I = \langle I, M(\varepsilon)^d \rangle : (\prod_{i \notin \varepsilon} x_i)^\infty.$$

Definition 3.1.6. For a set $\varepsilon \subset \{1, \ldots, n\}$, a pair (L, σ) where $L \subset \mathbb{Z}^\varepsilon$ is an integer lattice and $\sigma : L \to \mathbb{K}^*$ is homomorphism, is called a *partial character*. A partial character induces a *lattice ideal* in $\mathbb{K}[(x_i)_{i \in \varepsilon}]$

$$Lat(\sigma) := \langle x^{m_+} - \sigma(m) x^{m_-} | m \in L \rangle$$

where m is decomposed into the positive part m_+ and the negative part m_-, so that $m = m_+ - m_-$.

Lemma 3.1.7. The radical of a binomial cellular ideal I is represented with $\varepsilon \subset \{1, \ldots, n\}$ and a partial character (L, σ) such that

$$\sqrt{I} = \langle M(\varepsilon), Lat(\sigma) \rangle.$$

Definition 3.1.8. For a set $\varepsilon \subset \{1, \ldots, n\}$ and an integer lattice $L \subset \mathbb{Z}^{\varepsilon}$, we define the *saturation* of L

$$Sat(L) := \{m \in \mathbb{Z}^{\varepsilon} | dm \in L \text{ for some } d \in \mathbb{Z}\}.$$

A partial character (L', σ') is called a saturation of (L, σ) if

$$L' = Sat(L), \sigma'(l) = \sigma(l)(l \in L).$$

Theorem 3.1.9. For a cellular ideal I, if its radical is represented with a set $\varepsilon \subset \{1, \ldots, n\}$ and a partial character (L, σ) such that

$$\sqrt{I} = \langle M(\varepsilon), Lat(\sigma) \rangle,$$

then its minimal associated primes are given by

$$P_{\sigma'} = \langle M(\varepsilon), Lat(\sigma') \rangle$$

where σ' runs through all saturations of σ.

Note that the splitting tool generates redundant components in general. Therefore we will consider an algorithm without producing redundant components.

3.2 A New Algorithm for Minimal Associated Primes

We describe a new algorithm for computing minimal associated primes without producing redundant components. It is based on Laplagne's algorithm and utilizes cellular decomposition as an intermediate decomposition.

First of all, we show an algorithm which outputs a cellular ideal including a given ideal (Algorithm 8).

Theorem 3.2.1. Algorithm 8 works correctly.

Proof. A variable x_0 which is not nilpotent modulo J is non-zerodivisor modulo $J : x_0^{\infty}$ because polynomials f with $fx_0 \in J : x_0^{\infty}$ are in $J : x_0^{\infty}$. And it is easy to show if x_i is nilpotent(respectively non-zerodivisor) modulo J, then it is nilpotent(respectively non-zerodivisor) modulo $J : x_0^{\infty}$. Therefore C is a cellular ideal and $C = J : X^{\infty}$ with a monomial X. It implies $C \supset J$. This algorithm terminates after n loops.

□

This cellular ideal C from Algorithm 8 has the following property.

Lemma 3.2.2. $\sqrt{C}$ is an intersection of some components of minAss(J).

Algorithm 8 CELLULARIZE

Input: an ideal $J \subset \mathbb{K}[X]$
Output: a cellular ideal including J
  $V \leftarrow \{x_1, \ldots, x_n\}$
  $C \leftarrow J$
  while $V \neq \emptyset$ **do**
    choose $x_0 \in V$
    $V \leftarrow V \setminus \{x_0\}$
    if x_0 is not nilpotent (mod C) **then**
      $C \leftarrow C : x_0^\infty$
    end if
  end while
  return C

Proof. Let X be the product of all variables which are not nilpotent modulo C. X satisfies the condition of the splitting tool, hence

$$C = J : X^\infty.$$

Then C is an intersection of primary components of J because of Lemma 2.2.1. Computing radicals of both sides and removing redundant components, $\sqrt{C}$ is an intersection of some components of minAss(J).

□

Then we propose a new algorithm for the intermediate decomposition without producing redundant components (Algorithm 9).

Algorithm 9 INTERMEDIATECELLDECOMP

Input: an ideal $I \subset \mathbb{K}[X]$
Output: an intermediate decomposition of I s.t. $\sqrt{\cap\{C | C \in ID\}} = \sqrt{I}$
  $C_1, C_2 \in ID$ and $C_1 \neq C_2$ then $\mathrm{minAss}(C_1) \cap \mathrm{minAss}(C_2) = \emptyset$
  $Int \leftarrow \langle 1 \rangle$
  $ID \leftarrow \emptyset$
  while $Int \supsetneq \sqrt{I}$ **do**
    choose $g \in Int \setminus \sqrt{I}$
    $J \leftarrow I : g^\infty$
    $C \leftarrow$ CELLULARIZE(J)
    $Int \leftarrow Int \cap C$
    $ID \leftarrow ID \cup \{C\}$
  end while
  return ID

Theorem 3.2.3. Algorithm 9 works correctly without producing redundant components.

Proof. Let $\sqrt{I} = \bigcap_i \sqrt{Q_i}$ be the minimal prime decomposition. Lemma 2.2.1 implies

$$\sqrt{I : g^\infty} = \bigcap_{g \notin \sqrt{Q_i}} \sqrt{Q_i}.$$

From Lemma 3.2.2, C is decomposed minimally into a subset of these components. Let this decomposition be

$$\sqrt{C} = \bigcap_j \sqrt{Q_j}.$$

Since $g \in Int$, $\sqrt{Q_j}$ differs from components of $minAss(C')$ where $C' \in ID$. On the other hand, since $Int \supsetneq Int \cap \sqrt{C} \supset \sqrt{I}$, we obtain an expected output. In addition, this algorithm terminates in finite steps since the number of minAss(I) is finite and C is an intersection of new minimal associated primes.

□

Finally, a new algorithm for computing minimal associated primes has been completed. It decomposes a given ideal by Algorithm 9 then decomposes each component by Laplagne's algorithm. Through the algorithm, already-known minimal associated primes never appear again (Algorithm 10).

Algorithm 10 minAssβ

Input: an ideal $I \subset \mathbb{K}[X]$
Output: minAss(I)
  $MA \leftarrow \emptyset$
  $ID \leftarrow$ intermediateCellDecomp(I)
  while $ID \neq \emptyset$ **do**
    choose $C \in ID$
    $ID \leftarrow ID \setminus C$
    $MA \leftarrow$ LMinAss(C)
  end while
  return MA

We measure the time for computing minimal associated primes by Algorithm 10 and Laplagne's algorithm(Table 3.1). Laplagne's algorithm is implemented as a function `minAssGTZ` in Singular [DGPS]. With its option `minAssGTZ`(I, 1), it is more similar to Algorithm 5 than the default. For comparison, we also measure the default algorithm `minAssGTZ`(I). It utilizes the factorized Gröbner basis algorithm as an intermediate decomposition.

In this paper, the unit of timings is a second and all results have been rounded to no more than three significant figures. All of our algorithms were implemented in Singular [DGPS] and measured on a 64-bit Linux machine with Intel Xeon E5-2650 v2, 2.60GHz and 256GB memory. Definitions and examples of decomposed ideals are in Appendix. The library file of algorithms will be available from the URL [A16].

Table 3.1: Timing data of computing minimal associated primes

	$A(2,14)$	$A(3,9)$	$A(4,6)$	$P(2,13)$	$P(3,7)$
Algorithm 10	234	904	5990	185	126
`minAssGTZ`$(I,1)$	421	806	5810	190	119
`minAssGTZ`(I)	429	98	38	5	2

	$P(3,8)$	$P(4,6)$	$P(5,5)$	$I^{(1,4)}$	$I^{(2,2)}$
Algorithm 10	581	545	891	144	132
`minAssGTZ`$(I,1)$	537	438	679	107	87
`minAssGTZ`(I)	7	7	9	112	24

Algorithm 10 is much slower than `minAssGTZ`(I). However, it is as fast as `minAssGTZ`$(I,1)$. To improve it we measure the runtimes of its components.

Table 3.2: Details of Algorithm 10

	$A(2,14)$	$A(3,9)$	$A(4,6)$	$P(2,13)$	$P(3,7)$
Total	234	904	5990	185	126
radical membership	3.3	796	5780	159	116
saturation	0.3	43	188	13	4.2
cellularize	12	6.3	10	1.2	0.7
intersection	195	50	7.7	8.6	2.4
Laplagne algorithm	24	8.0	3.5	3.1	1.9

	$P(3,8)$	$P(4,6)$	$P(5,5)$	$I^{(1,4)}$	$I^{(2,2)}$
Total	581	545	891	144	132
radical membership	554	515	847	89	79
saturation	14	17	27	1.4	1.2
cellularize	1.2	1.2	1.6	3.0	2.6
intersection	8.4	8.3	12	46	44
Laplagne algorithm	3.1	3.3	3.9	4.7	4.9

Table 3.2 shows that radical membership tests are bottle-necks of this algorithm. In this implementation, we use a general-purpose function for radical membership tests. We will improve it in the next section. On the other hand, the data of Laplagne's algorithm (the bottom line) shows that Algorithm 9 is useful as an intermediate decomposition. Actually, all of cellular components are already prime in these examples.

Remark 3.2.4. Our algorithm works correctly not only for binomial ideals but also for general ideals. However, our algorithm dose not always decompose general ideals efficiently. By observation, it seems to relate with the number of variables which are zerodivisors modulo the given ideal. In general, binomials tend to have some variables as their factors. Conversely polynomials with many terms do not because every term must have a common variable. These variables transform the given ideal by saturations in Algorithm 8. Therefore, Algorithm

8 sometimes affects general ideals little or nothing. It means that most parts of the decomposition are performed by Laplagne's algorithm. Hence we restrict our targets to binomial ideals in this paper.

3.3 Improvements

We observe the behavior of Algorithm 10 and improve by various methods including heuristic approaches. Let I be an ideal in $\mathbb{K}[x]$ and for any polynomial f, let $\sqrt{f}$ denote the square free part of f.

3.3.1 Simplification of the Ideal

The goal of Algorithm 10 is to decompose $\sqrt{I}$, not I. Therefore we can transform I into an ideal whose radical is equal to $\sqrt{I}$. By Algorithm 11, I is enlarged without changing its radical.

Algorithm 11 SQUAREFREE

Input: an ideal $I = \langle f_1, \ldots, f_m \rangle \subset \mathbb{K}[X]$
Output: generators of an ideal J s.t. $\sqrt{J} = \sqrt{I}$
  return $\{\sqrt{f_1}, \ldots, \sqrt{f_m}\}$

Correctness of Algorithm 11 is clear by $\sqrt{I} = \sqrt{\langle \sqrt{f_1}, \ldots, \sqrt{f_m} \rangle}$. (See [AM, Chapter1 Exercise 1.13 v)].) Note that $\deg(\sqrt{f_i}) \leq \deg(f_i)$. With this algorithm, we obtain an algorithm for computing a Gröbner basis $S = \{s_1, \ldots, s_l\}$ s.t. $\sqrt{\langle S \rangle} = \sqrt{I}$.

Algorithm 12 SIMPLIFICATION

Input: an ideal $I \subset \mathbb{K}[X]$
Output: a Gröbner basis of J s.t. $\sqrt{J} = \sqrt{I}$
  $S \leftarrow$ a Gröbner basis of I
  $SF \leftarrow$ SQUAREFREE($\langle S \rangle$)
  while $\langle S \rangle \neq \langle SF \rangle$ **do**
    $S \leftarrow$ the reduced Gröbner basis of SF
    $SF \leftarrow$ SQUAREFREE($\langle S \rangle$)
  end while
  return S

Proposition 3.3.1. Algorithm 12 works correctly. In particular, $\langle S \rangle$ can be used instead of I in Algorithm 10.

Proof. Correctness is clear. The series of $\langle S \rangle$ is an ascending chain with proper inclusions. The properties of Noetherian ring ensure the termination.

□

By this simplification, we expect lower degrees in the Gröbner basis of J. They are used for executing radical membership tests and computations of saturations. And they are executed frequently through the algorithm. Therefore we expect that it saves the total computing time.

3.3.2 Choice of Polynomials for Radical Membership Tests

Algorithm 9 searches a polynomial $g \in Int \setminus \sqrt{I}$. The choice of g affects greatly the subsequent computation. Let $I = \bigcap_i Q_i$ be the minimal primary decomposition. If the number of $\sqrt{Q_i}$ containing g becomes large, then the number of minimal primary components of $I : g^\infty$ becomes small (See Lemma 2.2.1.). If $I : g^\infty$ is an intersection of small number of components, we expect its number of generators is small and they are low-degree.

To search for a polynomial which belongs to as many $\sqrt{Q_i}$ as possible, we propose the following strategy.

Strategy 3.3.2. Choose a polynomial g which has as many variables as possible.

Example 3.3.3. Let I be in $\mathbb{Q}[x_1, \dots x_{10}]$, $g_1 = x_1 - x_2$ and $g_2 = x_1x_3x_5x_7x_9 - x_2x_4x_6x_8x_{10}$. Consider which of the two has more chance to belong to minimal associated primes of I.

Let the leading monomial of g_1 be x_1 and the one of g_2 be $x_1x_3x_5x_7x_9$. Ideals which contain g_1 must have at least one generator whose leading monomial is x_1. In the case of g_2, the essential generator can have $2^5 = 32$ kinds of leading monomials. Even just limited to monomial ideals, g_1 belongs to ideals which contain both of x_1 and x_2. On the other hand, g_2 can be in ideals which contain at least one pair (x_s, x_t) where s is odd and t is even.

3.3.3 Saturations of Homogeneous Ideals with Respect to a Variable

In Algorithm 8, saturations of I with respect to a variable are performed many times. For homogeneous ideals, the following proposition is helpful.

Proposition 3.3.4. ([S96, Lemma 12.1])
Let J be a homogeneous ideal and $G = \{g_1, \dots, g_m\}$ the reduced Gröbner basis of J with respect to graded reverse lexicographic order with $x_1 > \cdots > x_n$. Then a Gröbner basis of $J : x_n^\infty$ with respect to the same order is

$$\{g_1/x_n^{l_1}, \dots, g_m/x_n^{l_m}\} \text{ where } l_i = \max\{l \in \mathbb{N} | x_i^l \text{ divides } g_i\}.$$

3.3.4 Computing Several Cellular Ideals at One Iteration

Algorithm 10 computes one cellular ideal at one iteration. The following proposition ensures computing several cellular ideals without loss of irredundancy.

Proposition 3.3.5. Let S be a Gröbner basis of J computed by Algorithm 12 and $g = g_1 \cdots g_s$ be a member of S.
Then, sets of minimal associated primes of $J : (g/g_i)^\infty$ are disjoint for $1 \leq i \leq s$.

Proof. For $i \neq j$, g_i and g_j do not have common factors because g is square free. Since $g_i \in J : (g/g_i)^\infty$, all minimal associated primes of $J : (g/g_i)^\infty$ have g_i. On the other hand, $J : (g/g_j)^\infty = (J : (g/g_i g_j)^\infty) : g_i^\infty$. From Lemma 2.2.1, all minimal associated primes of $J : (g/g_j)^\infty$ do not contain g_i.

□

With this proposition, we can compute several cellular ideals including a given ideal in a particular case (Algorithm 13).

Algorithm 13 SEVERALCELLS

Input: an ideal $J \subset \mathbb{K}[X]$ whose generators are square free,
$m = m_1 \cdots m_s$(the factorization of a monomial generator of J)
Output: cellular ideals including J
$ID \leftarrow \emptyset$
for $i = 1$ to s **do**
$C \leftarrow J : (m/m_i)^\infty$
$C \leftarrow$ CELLULARIZE(C)
$ID \leftarrow ID \cap \{C\}$
end for
return ID

Remark 3.3.6. From Proposition 3.3.5, Algorithm 13 also works correctly when m_i are square free polynomials. In this case, we compute saturations with respect to polynomials (not variables). In general, saturations with respect to polynomials are relatively slower than ones with respect to variables. Moreover, Proposition 3.3.4 is helpful to compute saturations with respect to variables. Therefore, we restrict m_i to variables in our algorithm.

If the given ideal has another monomial generator, we can try to compute other cellular ideals. However, the new ones are not always different from ones which we have already computed. Now, we can check whether a monomial generator produces new cellular ideals or not. For that, we record non-nilpotent variables with respect to cellular ideals.

Proposition 3.3.7. Let J be an ideal whose generators are square free, $C_1, \ldots, C_t$ cellular ideals including J and X_i a product of all non-nilpotent variables with respect to C_i. If there exist a monomial generator m of J and a variable factor x_j of m such that m/x_j does not divide any of $X_1, \ldots, X_t$, then the output of CELLULARIZE$(J : (m/x_j)^\infty)$ differs from $C_1, \ldots, C_t$.

Proof. If m/x_j does not divide X_i, then variables in m/x_j is not a subset of non-nilpotent variables with respect to C_i. Let x_0 be a variable which is in m/x_j

and not in X_i. If x_0 is not nilpotent with respect to a cellular ideal $C \subset J$, then C differs from $C_1, \ldots, C_t$ because x_0 is nilpotent with respect to $C_1, \ldots, C_t$. We show that x_0 can never become nilpotent in Algorithm 13. In the former part of Algorithm 13, x_0 becomes non-zerodivisor with respect to $J' := J : (m/x_j)^\infty$, namely, $J' : x_0^\infty = J' \neq \langle 1 \rangle$. If x_0 becomes nilpotent while CELLULARIZE(J'), there is a product of variables V such that $(J' : V^\infty) : x_0^\infty = \langle 1 \rangle$. Exchanging the two saturations, it means $(J' : V^\infty) = \langle 1 \rangle$ and such saturations are avoided in CELLULARIZE(J'). Therefore x_0 can never become nilpotent and the output cellular ideal differs from $C_1, \ldots, C_t$. □

Combining Algorithm 13 and Proposition 3.3.7, we obtain a recursive algorithm for computing cellular ideals including a given ideal I (Algorithm 14).

Algorithm 14 DISTINCTCELLS

Input: an ideal $I \subset \mathbb{K}[X]$ whose generators are square free,
$NonZD$: an argument for recursion (= 1 for the first time)
$NonNil$: an argument for recursion (= $\emptyset$ for the first time)
Output: cellular ideals (including I) whose minimal associated primes are distinct from each other

```
ID ← ∅
S ← SIMPLIFICATION(I)
Monom ← {M_1, ..., M_m} : monomial generators in S, deg(M_i) > 1
if Monom = ∅ then
    C ← CELLULARIZE(I)
    NonNil ← NonNil∪{the product of non-nilpotent variables modulo C}
    ID ← ID ∪ {C}
else
    for i = 1 to m do
        factorize M_i = v_1 ··· v_t
        for j = 1 to t do
            NewNonZD ← NonZD · (M_i/v_j)
            if NewNonZD does not divide any member of NonNil then
                J ← I : (M_i/v_j)^∞
                (Cell, NonNil) ← DISTINCTCELLS(J, NewNonZD, NonNil)
                ID ← ID ∪ Cell
            end if
        end for
    end for
end if
return (ID, NonNil)
```

3.3.5 Experiments

We show the timing data of the final version of our algorithm utilizing all the above improvements (MINASSC(I) ; Algorithm 15), Laplagne's algorithm(`minAssGTZ`(I))

and Kahle's algorithm(`binomialMinimalPrimes`(I)). Since Kahle's is implemented in Macaulay2 [M2], it is just a reference. About ideals in Table 3.1, our algorithm is faster than `minAssGTZ`(I) or finishes decomposing in a few seconds. Therefore we omit some examples in Table 3.1 and show timing data for more complicated ideals.

Algorithm 15 MINASSC

Input: an ideal $I \subset \mathbb{K}[X]$
Output: minAss(I)
  $MA \leftarrow \emptyset$, $ID \leftarrow \emptyset$, $Int \leftarrow \langle 1 \rangle$
  $S \leftarrow$ SIMPLIFICATION$(()I)$
  while $Int \setminus \sqrt{S} \neq \emptyset$ **do**
    choose $g \in Int \setminus \sqrt{S}$ following Strategy 3.3.2
    $J \leftarrow S : g^{\infty}$
    $J \leftarrow$ SIMPLIFICATION$(()J)$
    $(Cell, NonNil) \leftarrow$ DISTINCTCELLS$(J, 1, \emptyset)$
    $ID \leftarrow ID \cup Cell$
    $Int \leftarrow Int \cap \bigcap_{C \in ID} C$
  end while
  while $ID \neq \emptyset$ **do**
    choose $C \in ID$
    $MA \leftarrow MA \cup$ LMINASS(C)
  end while
  return MA

Table 3.3: Timing data of computing minimal associated primes

	MINASSC(I)	`minAssGTZ`(I)	`binomialMinimalPrimes`
$A(2,14)$	43	429	90
$A(3,10)$	165	642	248
$A(4,7)$	166	485	1430
$P(2,18)$	34	54	94
$P(3,10)$	40	48	61
$P(4,9)$	144	292	194
$I^{(1,5)}$	79	11100	> 40000
$I^{(1,6)}$	1120	> 50000	
$I^{(2,2)}$	12	24	36
$I^{(2,3)}$	4420	22900	> 50000

3.3.6 Discussion

Table 3.3 and Table 3.4 show that the above improvements are helpful and our new algorithm works well.

Table 3.4: Details of minAssC(I)

	$A(2,14)$	$A(3,10)$	$A(4,7)$	$P(2,18)$	$P(3,10)$
Total	43	165	166	34	40
SIMPLIFICATION	0	3.1	86	0.9	0.7
radical membership &saturation	1.8	33	16	1.7	0.4
CELLULARIZE	15	11	31	7.1	4.0
intersection	2.3	99	21	13	27
LMinAss	24	19	11	12	7.7

	$P(4,9)$	$I^{(1,5)}$	$I^{(1,6)}$	$I^{(2,2)}$	$I^{(2,3)}$
Total	144	79	1220	12	4420
SIMPLIFICATION	2.9	0.07	0.9	0.02	1.0
radical membership &saturation	0.1	3.9	58	0.37	33
CELLULARIZE	10	22	98	5.3	126
intersection	112	15	377	1.0	2870
LMinAss	18	37	684	5.5	1390

In many cases, the extra time for Algorithm 12 is not long and it shortens the time of radical membership tests and computations of saturations.

Strategy 3.3.2 is suitable for our algorithm. It makes significant contributions to radical membership tests and computations of saturations.

Proposition 3.3.4 works very efficiently. Owing to speeding up computations of saturations with respect to a variable, we can compute cellularizations very fast. Therefore, computing several cellular ideals from one ideal becomes a valid strategy.

With Algorithm 13 and Proposition 3.3.7, we can reduce the number of iterations. It means we can reduce the frequency of radical membership tests, computations of saturations and computations of intersections.

There is room for improvement in computations concerning intersections.

Chapter 4

A Modular Algorithm for Laplagne's Algorithm

In this chapter, we propose a modular algorithm for computing minimal associated primes of ideals in $\mathbb{Q}[X]$. Modular algorithms avoid the swell of coefficients which makes ideal computations slow-down. For computational targets in a ring R, modular algorithms choose projection maps R to R', take projected images of targets and compute in R' with the images to avoid the swell of coefficients. Then they reconstruct the real computed results in R from the computed results in R'. For reconstructions, the projection images need to maintain information of the original targets. We call a projection is lucky if its images are 'useful' for reconstructions. Luckiness depends on what computations we perform and in general, we can not decide whether a projection is lucky or not before computations. It means that the computation is probabilistic and that in many cases the computed results of modular algorithms are only candidates of the expected results and we should verify the correctness in some way. Therefore, it is important for modular algorithms to detect unlucky projections quickly and to guarantee the correctness of the computed results by efficient methods.

There are several researches about modular algorithms for ideal computations. Arnold [Ar] and Pauer [P] propose modular algorithms for computing Gröbner basis. Idrees-Pfister-Steidel [IPS] apply a modular algorithm for radical computations and computing minimal associated primes of zero-dimensional ideals. Noro-Yokoyama [NY] summarize them, describe the relation among several notions of luckiness and illustrate applications of modular algorithms for saturation, intersection, radical computation and primary decomposition.

In this chapter, we apply a modular algorithm for Laplagne's algorithm (Algorithm 5). It deals with a rational function field $\mathbb{K}(U)$ as a coefficient field, for the sake of reductions to zero-dimensional case. This tends to produce huge coefficients at intermediate computations. Therefore we apply modular algorithms which suppress the swell of coefficients. On the other hand, A modular algorithm for computing minimal associated primes of polynomial ideals has

been proposed in [IPS]. The most significant difference between our algorithm and the algorithm in [IPS] is the setting of projection maps. The algorithm in [IPS] utilizes projections $\mathbb{Q}$ to $\mathbb{F}_p$ where p is a prime number, while our algorithm utilizes projections $\mathbb{Q}(u)$ to $\mathbb{Q}$ (u is a parameter). Our projections reduce the number of parameters and keep the characteristic of coefficient fields 0.

In Section 4.1, we introduce some well-known tools on which our algorithm is based. Chinese Remainder Theorem guarantees the existence of an inverse image for given projected images. And the Lagrange's interpolation is an arithmetic method to compute a result whose existence is guaranteed by Chinese Remainder Theorem. Then we give definitions of luckiness for computing minimal associated primes. Our definitions are based on the luckiness for computing Gröbner basis defined in [NY].

Our main results are in Section 4.2. We construct a modular algorithm for computing a subset of minimal associated primes of zero-dimensional ideals in $\mathbb{Q}(U)[X]$. Then we apply it for Laplagne's algorithm. We show the correctness of our algorithm. We also show that the number of lucky moduli is sufficiently large so that we can obtain the correct result with a high probability. Then we show the results of our implementation of the new algorithm. We measure the time for computing minimal associated primes of some ideals. We see that our algorithm is efficient for ideals which take long time to compute minimal associated primes by the Laplagne's original algorithm.

4.1 Fundamental Tools and Definitions

In this section, we review well-known tools and define luckiness of ideals for constructing our new algorithms.

4.1.1 Chinese Remainder Theorem

Let R be a commutative ring. When we perform a computation of an object from an input $F \subset R$ utilizing Chinese Remainder Theorem, we choose some ideals $I_i \subset R$ and compute a modular image of the object from $F \bmod I_i$ on R/I_i. Interpolating these computed results we try to reconstruct the true object. Chinese Remainder Theorem is formulated as follows.

Theorem 4.1.1. (Chinese Remainder Theorem; CRT) Let R be a commutative ring and $I_1, \ldots, I_s$ pairwise comaximal ideals in R. For $r_1, \ldots, r_s \in R$, there exists $y \in R$ satisfying

$$\begin{aligned} y &\equiv r_1 \bmod I_1 \\ &\vdots \\ y &\equiv r_s \bmod I_s. \end{aligned}$$

y is unique modulo $\cap_{i=1}^{s} I_i$.

CRT can be applied in two typical situations: $R = \mathbb{Z}$ or $R = \mathbb{K}[u]$ where $\mathbb{K}$ is a field. In each case we illustrate Lagrange's interpolation which is one of concrete methods to construct y.

Lemma 4.1.2. (Lagrange's Interpolation in $\mathbb{Z}$) Let $p_1, \ldots, p_s$ be distinct prime numbers from each other, $p = p_1 \cdots p_s$ and $I_1 = \langle p_1 \rangle, \ldots, I_s = \langle p_s \rangle$. Then for $1 \leq i \leq s$, $a_i, b_i \in \mathbb{Z}$ such that

$$a_i(p/p_i) + b_i p_i = 1$$

can be computed by the extended Euclidean algorithm. For any $r_1, \ldots, r_s \in \mathbb{Z}$, the unique y satisfying conditions in Theorem 4.1.1 is given by

$$y = r_1 L_1 + \cdots + r_s L_s (\text{where } L_i = a_i(p/p_i)).$$

Lemma 4.1.3. (Lagrange's Interpolation in $\mathbb{K}[u]$) Let $k_1, \ldots, k_s \in \mathbb{K}$ be distinct elements from each other, $I_1 = \langle u - k_1 \rangle, \ldots, I_s = \langle u - k_s \rangle$ and

$$L_i = \frac{(u-k_1)\cdots(u-k_{i-1})(u-k_{i+1})\cdots(u-k_s)}{(k_i-k_1)\cdots(k_i-k_{i-1})(k_i-k_{i+1})\cdots(k_i-k_s)}.$$

Then the unique y satisfying conditions in Theorem 4.1.1 is given by

$$y = r_1 L_1 + \cdots + r_s L_s.$$

Definition 4.1.4. Let $r_1, r_2 \in \mathbb{K}[u]$ and I_1, I_2 comaximal ideals $\subset \mathbb{K}[u]$. We name the interpolation r_1 modulo I_1 and r_2 modulo I_2 $\mathrm{CRT}(r_1, r_2, I_1, I_2)$. For $f = \sum_\alpha c_\alpha x^\alpha, g = \sum_\alpha d_\alpha x^\alpha \in \mathbb{K}[u][X]$, we define $\mathrm{CRT}(f, g, I_1, I_2) = \sum_\alpha CRT(c_\alpha, d_\alpha, I_1, I_2) x^\alpha$. For $F = \{ f_1, \ldots, f_s \}$, $G = \{ g_1, \ldots, g_s \} \subset \mathbb{K}[u][X]$ where $LM(f_i)$'s and $LM(g_i)$'s are distinct respectively and $LM(f_i) = LM(g_i)$, we define $\mathrm{CRT}(F, G, I_1, I_2) = \{ \mathrm{CRT}(f_i, g_i, I_1, I_2) \mid 1 \leq i \leq s \}$. Let I, J be ideals in $\mathbb{K}[u]$, G_I, G_J the reduced Gröbner bases of I, J, respectively and $\mathrm{CRT}(G_I, G_J, I_1, I_2)$ is defined. We difine $\mathrm{CRT}(I, J, I_1, I_2) = \langle \mathrm{CRT}(G_I, G_J, I_1, I_2) \rangle$. Moreover, for $\mathcal{F} = \{ F_1, \ldots, F_t \}$ and $\mathcal{G} = \{ G_1, \ldots, G_t \}$ where $\mathrm{CRT}(F_i, G_i, I_1, I_2)$'s are defined, we define $\mathrm{CRT}(\mathcal{F}, \mathcal{G}, I_1, I_2) = \{ \mathrm{CRT}(F_i, G_i, I_1, I_2) \mid 1 \leq i \leq t \}$. When we compute CRT of indexed sets, we reset indices of members implicitly in order to complete the computation unless there are two or more candidates of indices which are suitable for the computation.

4.1.2 Rational function reconstruction

Our main target in this section is the reduced Gröbner basis G of a minimal associated prime of an ideal I over a rational function field $\mathbb{K}(u)$. If we apply CRT for the modular images computed over $\mathbb{K}$, what we obtain is an object G' over $\mathbb{K}[u]$. If a coefficient $c(u)$ appearing in G is not a polynomial we have to recover $c(u)$ from the corresponding polynomial coefficient in G'. This procedure is as follows. Suppose that we try reconstructing a rational function $\frac{g(u)}{h(u)} \in \mathbb{K}(u)$. Let $k_i \in \mathbb{K}$ such that $h(u) \notin \langle u - k_i \rangle$ and $\langle M \rangle = \cap_i \langle u - k_i \rangle$. Utilizing CRT, we obtain a polynomial $f(u) \in \mathbb{K}[u]$ such that $f(u) \equiv \frac{g(u)}{h(u)} \pmod{M}$. Then g, h can be recovered by the following theorem and algorithm (Algorithm 16).

Theorem 4.1.5. ([GG, Theorem5.16]) Let $f, M \in \mathbb{K}[x]$, $\deg(f) < \deg(M) = n > 0$ and $r_i, s_i, t_i \in \mathbb{K}[x]$ be the j-th row in extended Euclidean Algorithm for M, f ,where j is minimal such that $\deg(r_j) < k$. There exist polynomials $r, t \in \mathbb{K}[x]$ satisfying

$$r \equiv tf \pmod{M},\ \deg(r) < k,\ \deg(t) \leq n - k,$$

namely $r = r_j$, $t = t_j$. If in addition $\gcd(r_j, t_j) = 1$, then r, t also satisfy

$$\gcd(t, M) = 1, rt^{-1} \equiv f \pmod{M},\ \deg(r) < k,\ \deg(t) \leq n - k$$

Algorithm 16 RFR

Input: polynomials $f, M \in \mathbb{K}[x]$
Output: $g, h \in \mathbb{K}[x]$ s.t. $f \equiv g/h \pmod{M}$, h is monic and $\gcd(g, h) = 1$

```
r_0 ← M, r_1 ← f
t_0 ← 0, t_1 ← 1
i ← 1
while 2 deg(r_i) > deg(M) do
    R_i ← NF(r_{i-1}, {r_i})
    Q ← (r_{i-1} − R_i)/r_i
    r_{i+1} ← R_i, t_{i+1} ← t_{i-1} − Q t_i
    i ← i + 1
end while
return (r_i, t_i)
```

We also utilize the algorithm RFR for reconstructing coefficients of polynomials, ideals and a set of ideals.

Definition 4.1.6. Let $\langle M \rangle = \cap_i \langle u - k_i \rangle (k_i \in \mathbb{K}) \subset \mathbb{K}[u]$. For a polynomial $f = \sum_\alpha c_\alpha x^\alpha \in \mathbb{K}[u][X]$, we denote $\mathrm{RFR}(f, M) = \sum_\alpha \mathrm{RFR}(c_\alpha, M) x^\alpha$. For a subset $F \subset \mathbb{K}[u][X]$, we define $\mathrm{RFR}(F, M) = \{\, \mathrm{RFR}(f, M) \mid f \in F \,\}$. For an ideal $I \subset \mathbb{K}[u][X]$, we define $\mathrm{RFR}(I, M) = \langle \mathrm{RFR}(G, M) \rangle$ where G is the reduced Gröbner basis of I. Moreover, for $\mathcal{F} = \{\, F_1, \ldots, F_s \,\}$ where $\mathrm{RFR}(F_i, M)$'s are defined, we define $\mathrm{RFR}(\mathcal{F}, M) = \{\, \mathrm{RFR}(F, M) \mid F \in \mathcal{F} \,\}$.

Remark 4.1.7. According to Theorem 4.1.5, when we reconstruct $\frac{g(u)}{h(u)} \in \mathbb{K}(u)$ $(\gcd(g, h) = 1)$ from $f(u) \in \mathbb{K}[u]$ by RFR, we need more than $\deg(g) + \deg(h)$ ideals $\langle u - k_i \rangle$ ($k_i \in \mathbb{K}$ and $h(k_i) \neq 0$). With a shortage of ideals, RFR can return a rational function which is different from $\frac{g(u)}{h(u)}$. We say that the output of RFR is **stable** if we have more than $\deg(g) + \deg(h)$ ideals. However, we can not decide $\deg(g) + \deg(h)$ before computation in general. Therefore we say that the output is **pseudo stable** if $RFR(f(u), M) = RFR(f(u), M')$,where $\langle M \rangle = \cap_{i=1}^{r} \langle u - k_i \rangle$, $\langle M' \rangle = \cap_{i=1}^{s} \langle u - k_i \rangle$ $(r < s)$. When the output becomes pseudo stable, we regard the output as a candidate of the unique rational function.

4.1.3 Luckiness

For constructing modular algorithms, we have to define several notions of luckiness of moduli. The following definitions are extensions of [NY, Definition 2.1].

Definition 4.1.8. Let $u \notin X$ be a variable, F a subset of $\mathbb{K}(u)[X]$, G the reduced Gröbner basis of $\langle F \rangle$ and $k \in \mathbb{K}$, then $\langle u-k \rangle$ is a prime ideal in $\mathbb{K}[u]$.

1) $\mathbb{K}[u]_{(u-k)} := \{\frac{f}{g} | f, g \in \mathbb{K}[u], g(k) \neq 0\}$.

2) $\phi_{(u-k)} : \mathbb{K}(u) \to \mathbb{K}; f \mapsto f(k)$. We denote projection maps $\mathbb{K}[u]_{(u-k)} \to \mathbb{K}$ and $\mathbb{K}[u]_{(u-k)}[X] \to \mathbb{K}[X]$ by the same symbol $\phi_{(u-k)}$ such that $\frac{f}{g} \mapsto \frac{f(k)}{g(k)}$ and $\sum_\alpha c_\alpha x^\alpha \mapsto \sum_\alpha \phi_{(u-k)}(c_\alpha) x^\alpha$ (c_α is the coefficient of $c_\alpha x^\alpha$).

3) $I_{(u-k)}(F) := \langle \phi_{(u-k)}(f) | f \in F \rangle$, $I^0_{(u-k)}(F) := \phi_{(u-k)}(\langle F \rangle \cap \mathbb{K}[u, X])$, G_{u-k} denotes a Gröbner basis of $I_{(u-k)}(F)$.

4) $\langle u-k \rangle$ is said to be **weak permissible** for F if $F \subset \mathbb{K}[u]_{(u-k)}$. $\langle u-k \rangle$ is said to be **permissible** for F if $\langle u-k \rangle$ is weak permissible for F and $\phi_{(u-k)}(LC(f)) \neq 0$ for all $f \in F$.

5) $\langle u-k \rangle$ is said to be **compatible** for F if $\langle u-k \rangle$ is weak permissible for F and $I^0_{(u-k)}(F) = I_{(u-k)}(F)$. $\langle u-k \rangle$ is said to be **strong compatible** for F if $\langle u-k \rangle$ is weak compatible for F and $\phi_{(u-k)}(L(\langle F \rangle) \cap \mathbb{K}[u]_{(u-k)}[X]) = L(I_{(u-k)}(F))$.

6) $\langle u-k \rangle$ is said to be **lucky** for F if $\langle u-k \rangle$ is weak permissible for F and $LM(G) = LM(G_{u-k})$.

7) $\langle u-k \rangle$ is said to be **effectively lucky** for F if $\langle u-k \rangle$ is weak permissible for F, $\langle u-k \rangle$ is permissible for G and $\phi_{(u-k)}(G)$ is a Gröbner basis of $I_{(u-k)}(F)$.

8) Let $\sqrt{\langle G \rangle} = \cap_{i=1}^m P_i$ be the prime decomposition and G_i the reduced Gröbner basis of P_i. $\langle u-k \rangle$ is said to be **effectively minass lucky** for G if $\langle u-k \rangle$ is permissible for G and G_i ($i = 1, \ldots, m$), $\sqrt{I_{(u-k)}(G)} = \cap_{i=1}^m Q_i$ is the prime decomposition and $\phi_{(u-k)}(G_i)$ is the reduced Gröbner basis of Q_i.

Note that Definition 4.1.8 1) to 7) are defined for computing Gröbner basis by Noro-Yokoyama [NY]. Now, our goal is computing minimal associated primes. The computation of minimal associated primes includes not only computations of Gröbner basis but also computations of decomposition. Therefore we define Definition 4.1.8 8) as luckiness for computing minimal associated primes. The following lemma is fundamental.

Lemma 4.1.9. Let G be a Gröbner basis (respectively the reduced Gröbner basis) of $I \subset \mathbb{K}(u)[X]$ ($u \notin X$). If an ideal $\langle u-k \rangle$ is permissible for G, then $\phi_{(u-k)}(G)$ is a Gröbner basis (respectively the reduced Gröbner basis) of $I_{(u-k)}(G)$.

Proof. For $\overline{h} \in I_{(u-k)}(G)$, $\overline{h}$ is written as $\overline{h} = \sum_{g \in G} c_g \phi_{(u-k)}(g)$ where $c_g \in \mathbb{K}[X]$. Then $h = \sum_{g \in G} c_g g \in I \cap \mathbb{K}[u]_{(u-k)}[X]$ and $\phi_{(u-k)}(h) = \overline{h}$. Let $h_0 \in I \cap \mathbb{K}[u]_{(u-k)}[X]$ such that $\phi_{(u-k)}(h_0) = \overline{h}$ and $LM(h_0)$ is minimal. Since $h_0 \in I$, there exists $g \in G$ such that $LM(g) \mid LM(h_0)$. Since $\langle u-k \rangle$ is permissible for G, $\phi_{(u-k)}(LC(g)) \neq 0$. Set $h' = h_0 - \frac{LT(h_0)}{LT(g)} g$. Then $LM(h') < LM(h_0)$. If $\phi_{(u-k)}(LC(h_0)) = 0$ then $\phi_{(u-k)}(h') = \phi_{(u-k)}(h_0)$ and it contradicts the construction of h_0. Thus $\phi_{(u-k)}(LC(h_0)) \neq 0$ and $LM(h_0) = LM(\overline{h})$. Therefore $LM(\phi_{(u-k)}(g)) \mid LM(\overline{h})$ and $\phi_{(u-k)}(G)$ is a Gröbner basis of $I_{(u-k)}(G)$. If G is the reduced Gröbner basis of I, then $\phi_{(u-k)}(G)$ is a Gröbner basis of $I_{(u-k)}(G)$ consisting of monic polynomials. The permissibility implies $LM(G) = LM(\phi_{(u-k)}(G))$ and it is clear that $\phi_{(u-k)}(G)$ is the reduced Gröbner basis. □

4.2 New Algorithm

Algorithm 4 contains factorizations of polynomials and it may cause a problem which does not occur in the case of Gröbner basis computation : a problem caused by extraneous factors. For example, if we try to apply the modular algorithm over $\mathbb{Q}$, in many cases, a factorization over $\mathbb{F}_p$ produces more factors than over $\mathbb{Q}$ and it is hard to reconstruct the correct result from the results of modular computations. [IPS, Algorithm 3] is a modular algorithm for computing minimal associated primes which contains factorizations of polynomials however it performs reconstructions before factorizations and avoids factorizations over $\mathbb{F}_p$.

Now, Algorithm 5 regards some variables $U \subset X$ as parameters. Therefore we propose to apply Chinese Remainder Theorem over $\mathbb{Q}(U)$ for Algorithm 5. We fix some $u \in U$ and we reconstruct the result over $\mathbb{Q}(U)$ from the results of modular computation over $\mathbb{Q}(U \setminus \{u\})$ by using CRT and RFR. Namely, for an ideal $I = \langle G \rangle \subset \mathbb{Q}(U)[X \setminus U]$ where G is the reduced Gröbner basis of I, we find $\langle u - z \rangle$ which is permissible for G and compute the minimal associated primes of $\langle \phi_{(u-k)}(G) \rangle$. We gather these results for sufficiently many moduli for reconstructing the results over $\mathbb{Q}(U)$. Applying $\phi_{(u-k)}$ is equivalent to substituting k for u. Thus we can reduce one parameter. we repeat this procedure recursively and finally we compute minimal associated primes in $\mathbb{Q}[X \setminus U]$. Then we reconstruct parameters one by one recursively and obtain some members of the minimal associated primes (Algorithm 17).

We describe how to apply modular algorithms for Algorithm 4 concretely. Algorithm 17 is to compute a subset of $\mathrm{minAss}(\langle G \rangle)$. For showing the termination and correctness of Algorithm 17 we give several propositions. In the following let $U = \{u_1, \ldots, u_l\}$ be a set of parameters and $\mathbb{K} = \mathbb{Q}(U)$ a rational function field over $\mathbb{Q}$.

Proposition 4.2.1. Let $u \notin X$ be a parameter, $I \subset \mathbb{K}(u)[X]$ an ideal and $G = \{g_1, \ldots, g_m\}$ the reduced Gröbner basis of I. If $k \in \mathbb{K}$, $\langle u-k \rangle$ is permissible

Algorithm 17 modZeroMinAss

Input: G is a Gröbner basis of a zero-dimensional ideal in $\mathbb{Q}(U)[X]$,
U a set of parameters
Output: a subset P of minAss$(\langle G\rangle) = \{P_1, \ldots, P_m\}$ such that
$P = \{\ P_i \mid j \neq i \Rightarrow LM(P_j) \neq LM(P_i)\ \}$

```
if U = ∅ then
    MA ← zeroMinAss(⟨G⟩)
    MA ← { the reduced Gröbner basis of I | I ∈ MA }
    MA ← MA \ { G_j ∈ MA | G_k(k ≠ j) exists s.t. LM(G_j) = LM(G_k) }
    return P = {⟨G_i⟩|G_i ∈ MA}
end if
M ← 1
Z ← ∅
P ← ∅, Q ← ∅
u ← an element of U
while do
    z ← an integer not in Z s.t. ⟨u − z⟩ is effectively minass lucky for G
    Z ← Z ∪ {z}
    m ← u − z
    P' ← modZeroMinAss(φ_(u−z)(G), U \ {u})
    if P' = ∅ then
        return ∅
    end if
    if P ≠ ∅ then
        P' ← CRT(P, P', ⟨M⟩, ⟨m⟩)
    end if
    Q' ← RFR(P', mM)
    if Q = Q' then
        if for all G_i ∈ Q, ⟨G_i⟩ ⊃ ⟨G⟩ then
            return P = {⟨G_i⟩|G_i ∈ Q}
        end if
    end if
    M ← mM
    P ← P', Q ← Q'
end while
```

for G and $\overline{I} = I_{(u-k)}(G)$ is a prime ideal in $\mathbb{K}[X]$, then I is a prime ideal in $\mathbb{K}(u)[X]$.

Proof. From Lemma 4.1.9, $\phi_{(u-k)}(G)$ is the reduced Gröbner basis of $\overline{I}$. For $f, g \in \mathbb{K}(u)[X] \setminus I$, assume that $fg \in I$. We can regard f, g as G-reduced and $(u-k) \nmid f, g$ without loss of generality. Then $\phi_{(u-k)}(f) \neq 0$ and $\phi_{(u-k)}(g) \neq 0$. On the other hand, $\phi_{(u-k)}(fg) = \phi_{(u-k)}(f)\phi_{(u-k)}(g) \in \overline{I}$. Since $\overline{I}$ is a prime ideal, $\phi_{(u-k)}(f) \in \overline{I}$ or $\phi_{(u-k)}(g) \in \overline{I}$. Since $\phi_{(u-k)}(f), \phi_{(u-k)}(g)$ are $\phi_{(u-k)}(G)$-reduced, $\phi_{(u-k)}(f) = 0$ or $\phi_{(u-k)}(g) = 0$.

□

Proposition 4.2.2. Let P, Q be ideals in $\mathbb{K}(u)[X]$, $G = \{g_1, \ldots, g_s\}$ the reduced Gröbner basis of P, $H = \{h_1, \ldots, h_r\}$ the reduced Gröbner basis of Q. If $k \in \mathbb{K}$ and $\langle u-k \rangle$ is permissible for G, H and $\langle \phi_{(u-k)}(G) \rangle \not\subset \langle \phi_{(u-k)}(H) \rangle$, then $P \not\subset Q$.

Proof. Take a polynomial $\overline{f} \in \langle \phi_{(u-k)}(G) \rangle \setminus \langle \phi_{(u-k)}(H) \rangle$. $\overline{f}$ can be written as $\overline{f} = \sum_{i=1}^{s} c_i \phi_{(u-k)}(g_i)$ $(c_i \in \mathbb{K})$. Set $f = \sum_{i=1}^{s} c_i g_i \in P$. If $P \subset Q$, then f can be written as $f = \sum_{i=1}^{r} d_i h_i$ $(d_i \in \mathbb{K}[u]_{(u-k)})$ because H is the reduced Gröbner basis of Q and $\langle u-k \rangle$ is permissible for H. Then $\overline{f} = \phi_{(u-k)}(f) = \sum_{i=1}^{r} \phi_{(u-k)}(d_i)\phi_{(u-k)}(h_i) \in \langle \phi_{(u-k)}(H) \rangle$. It is a contradiction. □

Theorem 4.2.3. Algorithm 17 terminates and outputs a subset of $\mathrm{minAss}(\langle G \rangle)$.

Proof. If $U = \emptyset$ then the algorithm simply calls zeroMinAss and the output is correct. We assume that the algorithm terminates and outputs a correct result in the case $\#U = s$. Suppose $\#U = s + 1$. Let $G_1, \ldots, G_m$ be the reduced Gröbner bases of the minimal associated primes of $\langle G \rangle$. Set

$$\{ G_{i_1}, \ldots, G_{i_k} \} = \{ G_i \mid j \neq i \Rightarrow LM(G_j) \neq LM(G_i) \}.$$

Since $\langle u - z \rangle$ is effectively minass lucky,

$$\mathrm{minAss}(\langle \phi_{(u-z)}(G) \rangle) = \{ \langle \phi_{(u-z)}(G_1) \rangle, \ldots, \langle \phi_{(u-z)}(G_m) \rangle \}$$

and $LM(G_i) = LM(\phi_{(u-z)}(G_i))$ $(i = 1, \ldots, m)$. From the assumption on $\#U = s$,

$$P' = \{ \phi_{(u-z)}(G_{i_1}), \ldots, \phi_{(u-z)}(G_{i_k}) \}$$

and for each $H \in P'$ there exists the unique element G_i such that $LM(H) = LM(G_i)$. Thus we can combine the correct modular images by CRT and Q will be eventually the set $\{ G_{i_1}, \ldots, G_{i_k} \}$ after sufficient interpolations. In this case Q satisfies the termination condition and the termination of the algorithm is guaranteed.
When the algorithm terminates, from Proposition 4.2.1, every $P_i \in P$ is a prime ideal in $\mathbb{Q}(U)[X]$ and $P_i \supset \langle G \rangle$. Then we have $\sqrt{P_i} = P_i \supset \sqrt{\langle G \rangle} = \cap_{i=1}^{m} \langle G_i \rangle$, which implies that $P_i \supset \langle G_j \rangle$ for some j. Since $\langle G \rangle$ is zero-dimensional $\langle G_j \rangle$ is maximal and we have $P_i = \langle G_j \rangle$. Thus every $P_i \in P$ is a member of $\mathrm{minAss}(\langle G \rangle)$ and the result is correct in the case $\#U = s + 1$.

□

Remark 4.2.4. In Algorithm 17, the recursive application of Proposition 4.2.2 implies that the output P has no redundant components.

Remark 4.2.5. In Algorithm 17, depending on the input, some prime components of P' can have the same leading monomial set. In such a case, we can not determine which pair of ideals we should interpolate. Therefore we do not perform interpolations for such components. If all of the components do not have unique leading monomial set unfortunately, we utilize Algorithm 4 for computing the minimal associated primes of the zero-dimensional ideal.

Remark 4.2.6. In Algorithm 17, we cannot decide whether a modulus $\langle u - z \rangle$ is effectively minass lucky during the computation. If we choose a modulus which is permissible for G, we can obtain a subset of $\mathrm{minAss}(\langle \phi_{(u-z)}(G) \rangle)$ by calling Algorithm 17 but the result may not be $\{\phi_{(u-z)}(G_{i_1}), \ldots, \phi_{(u-z)}(G_{i_k})\}$. In this case the result is a noise for our modular algorithm and we have to add some additional criteria or preprocessing to avoid bad moduli as much as possible. However, even if we do not assume the effective minass luckiness of moduli, if the algorithm terminates then the result is a subset of $\mathrm{minAss}(\langle G \rangle)$. This is ensured by the last part of the proof of Theorem 4.2.3.

Utilizing Algorithm 17 instead of Algorithm 4 in Algorithm 5, we can compute $\mathrm{minAss}(I)$ for $I \subset \mathbb{Q}[X]$ (Algorithm 18).

Algorithm 18 modLMinAss

Input: an ideal $I \subset \mathbb{Q}[X]$
Output: $\mathrm{minAss}(I)$
  Int $\leftarrow \langle 1 \rangle$, MA $\leftarrow \emptyset$
  while Int $\setminus \sqrt{I} \neq \emptyset$ **do**
    choose $g \in$ Int $\setminus \sqrt{I}$
    $U \leftarrow$ a maximal independent set of $I : g^\infty$
    $G \leftarrow$ a Gröbner basis of $I : g^\infty$ in $\mathbb{Q}(U)[X \setminus U]$
    $P \leftarrow$ modZeroMinAss(G, U)
    if $P = \emptyset$ **then**
      $P \leftarrow$ zeroMinAss$(\langle G \rangle)$
    end if
    $PG \leftarrow \{P_i \cap \mathbb{Q}[X] \mid P_i \in P\}$
    MA $\leftarrow$ MA$\cup PG$, Int $\leftarrow$ Int $\cap \bigcap_{P \in PG} P$
  end while
  return MA

Theorem 4.2.7. Algorithm 18 works correctly.

Proof. (correctness) Since Algorithm 17 outputs a subset of $\mathrm{minAss}(I : g^\infty)$ in $\mathbb{Q}(U)[X \setminus U]$, PG is a subset of $\mathrm{minAss}(I : g^\infty)$ in $\mathbb{Q}[X]$ by Proposition 2.2.4. Therefore MA is always a subset of $\mathrm{minAss}(I)$ by Lemma 2.2.1 and Int $\supset \sqrt{I}$.

When the termination condition is satisfied, Int $= \sqrt{I}$ and $MA = \mathrm{minAss}(I)$. (termination) Let MA $\subset \mathrm{minAss}(I)$ and Int $= \bigcap_{P \in MA} P$ (if MA $= \emptyset$, we define Int $= \langle 1 \rangle$). Suppose Int $\neq \sqrt{I}$, $g \in$ Int $\setminus \sqrt{I}$, $\sqrt{I : g^\infty} = \bigcap_{i=1}^{m} P_i$ is the prime decomposition and PG is a subset of $\mathrm{minAss}(I : g^\infty)$. For all $P_i \in PG$, $g \notin P_i$ and $P_i \in \mathrm{minAss}(I)$ by Lemma 2.2.1. On the other hand, since $g \in$ Int, for all $Q_i \in$ MA, $g \in Q_i$. Therefore for all $P_i \in PG$, $P_i \notin$ MA. In other words, Algorithm 18 obtains at least one new components in every loop. Since the number of components of $\mathrm{minAss}(I)$ is finite, Algorithm 18 terminates in finite steps. □

4.2.1 Existence of minass lucky moduli

In Algorithm 17 and Algorithm 18, we suppose all $\langle u - z \rangle$ are effectively minass lucky. However, effective minass luckiness is defined depending on the minimal associated primes of the given ideal. In general, we can not decide whether an ideal is effectively minass lucky or not while the computation. Therefore we show that there are sufficiently many effectively minass lucky ideals and we can obtain them with a high probability by random choice. Let G be the reduced Gröbner basis of a zero-dimensional ideal $I \subset \mathbb{K}(u)[X]$, $\sqrt{\langle G \rangle} = \cap_{i=1}^{m} P_i$ the prime decomposition, G_i the reduced Gröbner basis of P_i. If $\langle u - k \rangle$ is permissible for G and G_i's then $\phi_{(u-k)}(G)$ and $\phi_{(u-k)}(G_i)$'s are Gröbner basis of $\langle \phi_{(u-k)}(G) \rangle$ and $\langle \phi_{(u-k)}(G_i) \rangle$'s respectively, and $LM(G) = LM(\phi_{(u-k)}(G))$ and $LM(G_i) = LM(\phi_{(u-k)}(G_i))$ imply $\langle \phi_{(u-k)}(G) \rangle$ and $\langle \phi_{(u-k)}(G_i) \rangle$'s are zero-dimensional. For simplicity, we assume that I is in general position with respect to x_n. This implies that each $\langle G_i \rangle$ is in general position with respect to x_n. If the monomial order is the lexicographical order, then G_i is a shape base, i.e. $G_i = \langle x_1 - c_1(u), \ldots, x_{n-1} - c_{n-1}(u), g_i(x_n) \rangle$, $c_1(u), \ldots, c_{n-1}(u), g_i(x_n) \in \mathbb{K}(u)[x_n]$ and $g_i(x_n)$ is irreducible over $\mathbb{K}(u)$ because $\langle G_i \rangle$ is zero-dimensional and prime. Set

$$NP = \{k \in \mathbb{K} \mid \langle u - k \rangle \text{ is not permissible for } G \text{ or some } G_i\}.$$

Then NP is a finite set. A modulus $\langle u - k \rangle$ is effectively minass lucky for G if the following four conditions hold.

1) $k \notin NP$.
2) $\sqrt{I_{(u-k)}(G)} = I_{(u-k)}(G_1) \cap \cdots \cap I_{(u-k)}(G_m)$.
3) If $i \neq j$, then $I_{(u-k)}(G_i) \neq I_{(u-k)}(G_j)$.
4) Each $I_{(u-k)}(G_i)$ is prime.

First of all we consider the condition 2).

Lemma 4.2.8. Let $G \subset \mathbb{K}(u)[X]$ be the reduced Gröbner basis of a zero-dimensional ideal $\langle G \rangle$ and $H \subset \mathbb{K}(u)[X]$ the reduced Gröbner basis of $\sqrt{\langle G \rangle}$. Except for a finite number of $k \in \mathbb{K} \setminus NP$, $\sqrt{I_{(u-k)}(G)} = I_{(u-k)}(H)$.

Proof. If $\langle u-k\rangle$ is permissible for G, H, then $G \subset \langle H\rangle$ implies $\phi_{(u-k)}(G) \subset I_{(u-k)}(H)$ and $H \subset \sqrt{G}$ implies $\phi_{(u-k)}(H) \subset \sqrt{I_{(u-k)}(G)}$. Thus we have $\sqrt{I_{(u-k)}(H)} = \sqrt{I_{(u-k)}(G)}$. Since $\langle H\rangle$ is zero-dimensional and radical, for each $x_i \in X$ there exists a univariate square-free polynomial $f_i(x_i) \in \langle H\rangle$. Then $r_i(u) = \mathrm{resultant}_{x_i}(f_i, f_i') \neq 0$. If $\langle u-k\rangle$ is permissible for $f_i(x_i)$ and $r_i(k) \neq 0$ for all i, then $\phi_{(u-k)}(f_i) \in I_{(u-k)}(H)$ is square-free. Then $I_{(u-k)}(H)$ is radical and in this case $\sqrt{I_{(u-k)}(H)} = I_{(u-k)}(H) = \sqrt{I_{(u-k)}(G)}$. Since the number of $k \notin NP$ such that k is not permissible for $f_i(x_i)$'s, or $r_i(k) = 0$ for some i is finite, the assertion is proved. □

Proposition 4.2.9. Except for a finite number of $k \in \mathbb{K} \setminus NP$, $\sqrt{I_{(u-k)}(G)} = I_{(u-k)}(G_1) \cap \cdots \cap I_{(u-k)}(G_m)$.

Proof. Set

$$\tilde{I} = \langle 1-(t_1+\cdots+t_m), t_1G_1, \ldots, t_mG_m\rangle \subset \mathbb{K}(u)[t_1,\ldots,t_m,X].$$

If $\tilde{H}$ is the reduced Gröbner basis of $\tilde{I}$ with respect to an elimination ordering such that $\{t_1,\ldots,t_m\} >> X$, then $H = \tilde{H} \cap \mathbb{K}(u)[X]$ is the reduced Gröbner basis of $\sqrt{\langle G\rangle} = \langle G_1\rangle \cap \cdots \cap \langle G_m\rangle$. If $\langle u-k\rangle$ is permissible for all intermediate polynomials appearing during the execution of Buchberger's algorithm for computing $\tilde{H}$, then the remainder computations in the execution can be mapped by $\phi_{(u-k)}$. This implies that the reduced Gröbner basis of

$$\langle 1-(t_1+\cdots+t_m), t_1\phi_{(u-k)}(G_1), \ldots, t_m\phi_{(u-k)}(G_m)\rangle$$

with respect to the same elimination ordering is $\phi_{(u-k)}(\tilde{H})$. Since $\langle u-k\rangle$ is permissible for $\tilde{H}$, $\phi_{(u-k)}(\tilde{H}) \cap \mathbb{K}[X] = \phi_{(u-k)}(H)$ and $\phi_{(u-k)}(H)$ is the reduced Gröbner basis of $I_{(u-k)}(G_1) \cap \cdots \cap I_{(u-k)}(G_m)$. By Lemma 4.2.8 $\sqrt{I_{(u-k)}(G)} = I_{(u-k)}(H)$ except for a finite number of $k \in \mathbb{K}$. Therefore $\sqrt{I_{(u-k)}(G)} = I_{(u-k)}(H) = I_{(u-k)}(G_1) \cap \cdots \cap I_{(u-k)}(G_m)$ except for a finite number of $k \in \mathbb{K} \setminus NP$. □

Next we consider the condition 3).

Proposition 4.2.10. Except for a finite number of $k \in \mathbb{K} \setminus NP$, $I_{(u-k)}(G_i)$'s are distinct.

Proof. If $i \neq j$ then $\langle G_i\rangle$ and $\langle G_j\rangle$ are comaximal and $1 \in \langle G_i\rangle + \langle G_j\rangle$. Thus if $\langle u-k\rangle$ is permissible for G_i, G_j and all the coefficients in the generating relation of 1 by G_i and G_j, then $1 \in I_{(u-k)}(G_i) + I_{(u-k)}(G_j)$, which implies $I_{(u-k)}(G_i) \neq I_{(u-k)}(G_j)$. Thus $I_{(u-k)}(G_i)$'s are distinct except for a finite number of $k \in \mathbb{K} \setminus NP$. □

Finally we consider the condition 4).

Proposition 4.2.11. ([Z, Proposition 132, Proposition 133]) Let $F(X_1, \ldots, X_n, Y_1, \ldots, Y_m)$ be an irreducible polynomial over $\mathbb{Q}$ and let $R(N)$ denote the number of integer x_i with $|x_i| < N$ such that $F(x_1, \ldots, x_n, Y_1, \ldots, Y_m)$ is reducible. Then

$$R(N) < cN^{n-1/2} \log N$$

where c depends only on the degree of F.

Proposition 4.2.12. Set

$$N_i = \{k \in \mathbb{Z} \mid |k| < N, k \notin NP, I_{(u-k)}(G_i) \text{ is not prime}\}.$$

Then $\#N_i \leq cN^{1/2} \log N$ for a constant c.

Proof. If $\langle u - k \rangle$ is permissible for G_i, then $\phi_{(u-k)}(G_i)$ is the reduced Gröbner basis of $I_{(u-k)}(G_i)$. $I_{(u-k)}(G_i)$ is prime if and only if $\phi_{(u-k)}(g_i(x_n))$ is irreducible. $g_i(x_n)$ can be written as $g_i(x_n) = \tilde{g}(u, u_1, \ldots, u_l, x_n)/d(u, u_1, \ldots, u_l)$ with $\tilde{g} \in \mathbb{Q}[u, u_1, \ldots, u_l, x_n]$, with $d \in \mathbb{Q}[u, u_1, \ldots, u_l]$ and $\tilde{g}$ is irreducible over $\mathbb{Q}$. Then the irreducibility of $\phi_{(u-k)}(g_i(x_n))$ is equivalent to that of $\tilde{g}(k, u_1, \ldots, u_l, x_n)$ and Proposition 4.2.11 with the case $n = 1$ implies $\#N_i \leq cN^{1/2} \log N$ for a constant c. □

Theorem 4.2.13. Set

$$NEML = \{k \in \mathbb{Z} \mid |k| < N, k \notin NP, \langle u - k \rangle \text{ is not effectively minass lucky for G}\}.$$

Then there exist constants c_1, c_2 such that $\#(NP \cup NEML) \leq c_1 + c_2 N^{1/2} \log N$.

Proof. Set

$$BAD_2 = \{k \in \mathbb{K} \setminus NP \mid \sqrt{I_{(u-k)}(G)} \neq I_{(u-k)}(G_1) \cap \cdots \cap I_{(u-k)}(G_m)\},$$

$$BAD_3 = \{k \in \mathbb{K} \setminus NP \mid I_{(u-k)}(G_i) = I_{(u-k)}(G_j) \text{ for some } i, j (i \neq j)\}.$$

Then BAD_2 and BAD_3 are finite sets by Proposition 4.2.9 and Proposition 4.2.10 respectively. Then Proposition 4.2.12 implies $\#(NP \cup NEML) \leq (\#NP + \#BAD_2 + \#BAD_3) + (mc)N^{1/2} \log N$. □

Corollary 4.2.14. If k is randomly chosen from $\{k \in \mathbb{Z} \mid |k| < N\}$, then the probability that $\langle u - k \rangle$ is effectively minass lucky tends to 1 as $N \to \infty$.

4.3 Experiments and Timing data

We measure the timings for computing minimal associated primes by our algorithm and Laplagne's algorithm. Laplagne's algorithm is implemented as a function `minAssGTZ` in SINGULAR [DGPS]. In this paper, the unit of timing is a second and all results have been rounded to no more than three significant digits. All of our algorithms were implemented in SINGULAR [DGPS] and measured

on a 64-bit Linux machine with Intel Xeon E5-2650 v2, 2.60GHz and 256GB memory.

In Subsection 4.2.1, we have shown that there are sufficiently many effectively minass lucky moduli. However, when we choose an unlucky modulus there is a possibility that our algorithm does not terminate. Therefore we should discard unlucky modular images during computations. We adopt following three strategies to terminate Algorithm 17 with a high probability.

Strategy 4.3.1. 1) Choose $z \in \mathbb{Z} \setminus Z$ such that $\langle u - z \rangle$ is permissible for G.

2) For z_i's satisfying 1), compute modZeroMinAss$(I_{(u-z_i)}(G))$, classify them by leading monomial sets of their components and perform CRT for the class of largest cardinality.

3) If Q is pseudo stable and there are some components which do not include $\langle G \rangle$, then we discard them and return components which include $\langle G \rangle$.

As explained in Remark 4.2.6, when Algorithm 17 with Strategy 4.3.1 terminates, the output is a set of minimal associated primes of the input ideal $\langle G \rangle$ even if we choose some moduli which are not effectively minass lucky for G in the computational process. If Algorithm 17 with Strategy 4.3.1 terminates, then Algorithm 18 terminates and outputs the minimal associated primes of the input ideal.

Furthermore, there are three improvements for Algorithm 18 which are not written in the pseudo code for the sake of simplicity. The first one is recording the number of moduli for reconstructing. For a parameter u and an integer $z_1 \in \mathbb{Z}$, the number of moduli which makes modMinAss$(\phi_{(u-z_1)}(G))$ pseudo stable becomes a hint to decide how many moduli we should gather for modMinAss$(\phi_{(u-z_i)}(G))$'s where z_i's are distinct from z_1. The second one is utilizing modular computations for RFR. We do not need the exact value of RFR before the output become pseudo stable. Therefore we perform RFR over $\mathbb{F}_p$ where p is a prime number. We perform RFR over the original coefficient field only when we confirm pseudo stability of RFR over $\mathbb{F}_p$. The last one is the preprocessing Algorithm 12. We name Algorithm 18 with these improvements and Strategy 4.3.1 Algorithm 18'.

We construct examples of ideals from ideals given in [DGP, 3 Examples]. We classify ideals in [DGP, 3 Examples] by their number of variables. For $I_i \in \mathbb{Q}[v_1, \ldots, v_n]$ and $I_j \in \mathbb{Q}[u_1, \ldots u_n]$, we set a map

$$\varphi_{u,v} : \mathbb{Q}[u_1, \ldots u_n] \to \mathbb{Q}[v_1, \ldots, v_n]; u_m \mapsto v_m (1 \leq m \leq n)$$

and denote that

$$I_{i \cap j} = I_i \cap \varphi_{u,v}(I_j)$$

In addition, we utilize the factorizing Gröbner basis algorithm for some examples in order to construct more examples. Note that the factorizing Gröbner basis algorithm is an intermediate decomposition of `minAssGTZ` and returns a list of ideals and the radical of the intersection of them coincides with the radical of the

given ideal. For examples which can not be decomposed in four hours by both of algorithms, we decompose them by the factorizing Gröbner basis algorithm and treat each component as a new example.

For these examples, we measure the timings for computing minimal associated primes by Algorithm 18' and Laplagne's algorithm (Table 4.1). In addition, we show runtimes of its components (Table 4.2). We omit examples which are decomposed in a few seconds by Laplagne's algorithm or do not terminate in four hours by both of algorithms and zero-dimensional. For zero-dimensional ideals, Algorithm 18' and Laplagne's algorithm simply call Algorithm 4.

Remark 4.3.2. As a characteristic of modular algorithms, if there are no swells of coefficients in the original computations, then computations of modular algorithms have little advantages over original one. Moreover, modular algorithms iterate modular computations for some moduli and reconstruct their results. They become extra times of computations.

Table 4.1: Timing data of computing minimal associated primes of examples

	$I_{3\cap 8}$	$I_{18\cap 31}$	$I_{18\cap 33}$	$I_{31\cap 33}$	$I_{7\cap 9}$	$I_{7\cap 12}$	$I_{5\cap 23}$	$F_{5\cap 23}[24]$
Variables	3	4			6		8	
Algorithm 18'	1.2	34.4	28.4	34.9	54.9	240	5880	15.3
Laplagne's	$> 4h$	$> 4h$	$> 4h$	$> 4h$	284	87.1	12100	11300

	$I_{1\cap 4}$	$F_{1\cap 4}[1]$	$F_{1\cap 4}[5]$	$F_{1\cap 4}[7]$	$F_{1\cap 4}[8]$
Variables	9				
Algorithm 18'	812	10.0	5140	11.3	7.4
Laplagne's	$> 4h$	737	$> 4h$	$> 4h$	14

$F_{i\cap j}[k]$ denotes the k-th component of `facstd`$(I_{i\cap j})$.

4.4 Concluding Remarks

Our algorithm is fast for some class of ideals and will be a choice when general-purpose algorithms can not decompose ideals in practical time. In addition, our algorithm is suitable for parallelizations. In this paper, we have not yet implemented the parallel version of our algorithm and we expect further speed-up with parallelizations. It is a future project.

In Subsection 4.2.1, we have shown that there are sufficiently many effectively minass lucky moduli. However there are infinite moduli which are not effectively minass lucky in general. On the other hand, in the case of computing Gröbner basis by modular algorithms over $\mathbb{Z}_p$, it is shown that the number of unlucky primes is finite [NY, Section 3] and we can construct algorithms which terminate in finite steps even if we choose some unlucky primes in the computational process [NY, Section 5]. In order to improve our algorithm, we need more researches of luckiness and moduli whose modular images of irreducible

Table 4.2: Details of Algorithm 18'

	$I_{3\cap 8}$	$I_{18\cap 31}$	$I_{18\cap 33}$	$I_{31\cap 33}$	$I_{7\cap 9}$	$I_{7\cap 12}$	$I_{5\cap 23}$	$F_{5\cap 23}[24]$
Variables	3	4			6		8	
Total	1.2	34.4	28.4	34.9	54.9	240	5880	15.3
SIMPLIFICATION	0.02	0.02	0.03	0.04	0.07	2.64	0.07	0.02
radical membership &saturation	0.82	33.6	27.4	33.7	40.4	179	5860	9.05
modular	0.16	0.39	0.77	0.79	10.6	42	7.85	4.05
ZEROMINASS	0.08	0.08	0.09	0	0.09	0.18	4.41	0.06
intersection	0.09	0.17	0.01	0.21	3.8	16.3	13.2	1.7

	$I_{1\cap 4}$	$F_{1\cap 4}[1]$	$F_{1\cap 4}[5]$	$F_{1\cap 4}[7]$	$F_{1\cap 4}[8]$
Variables	9				
Total	812	10.0	5141	11.3	7.41
SIMPLIFICATION	0.3	0.02	0.03	0.04	0.02
radical membership &saturation	640	0.2	11.5	30.1	1.87
modular	135	8.03	5.89	6.45	3.56
ZEROMINASS	0	0.02	0	0.01	0.01
intersection	35, 8	1.34	1.16	0.97	0.12

polynomials keeps their irreducibility. It will be closely related with Hilbert's irreducibility theorem and its applications.

Appendix

We set

$$V(s,t) := (x_{ij})_{\substack{1\leq i\leq s \\ 1\leq j\leq t}} .$$

Definition 1. (**adjacent minor ideal**)([S02, CHAPTER 5.3])
$s \times t$ two-degree *adjacent minor ideal* is defined by

$$A_{2,s,t} := A(s,t) := \langle \{\det B | B \text{ is an adjacent } 2\times 2\text{-submatrix of } V(s,t)\}\rangle.$$

Example 1. $A_{2,3,4} = A(3,4) = \langle -x_{12}x_{21}+x_{11}x_{22}, -x_{13}x_{22}+x_{12}x_{23}, -x_{14}x_{23}+x_{13}x_{24}, -x_{22}x_{31}+x_{21}x_{32}, -x_{23}x_{32}+x_{22}x_{33}, -x_{24}x_{33}+x_{23}x_{34}\rangle$.

Definition 2. (**permanental ideal**)([S02, CHAPTER 5.4])
The *permanent* of 2×2-matrix is the sum over its diagonal product i.e.

$$per \begin{pmatrix} a & b \\ c & d \end{pmatrix} := ad + bc.$$

Then $s \times t$ two-degree *permanental ideal* is defined by

$$Per_{2,s,t} := P(s,t) := \langle \{per(Q) | Q \text{ is a } 2\times 2\text{-submatrix of } V(s,t)\}\rangle$$

Example 2. $Per_{2,2,3} = P(2,3) = \langle x_{11}x_{21}+x_{12}x_{21}, x_{11}x_{23}+x_{13}x_{21}, x_{12}x_{23}+x_{13}x_{22}\rangle$.

Definition 3. (**birth-and-death ideal**)([ESE, Section 5])
For a pair of integers (s,t), consider a two-dimensional integer lattice E such that

$$E := \{0,\ldots,s-1\}\times\{0,\ldots,t-1\}.$$

Variables are

$$\{R_{i,j} | 0\leq i<s, 0\leq j\leq t\}\cup\{L_{i,j} | 0<i\leq s, 0\leq j\leq t\}\cup$$

$$\{D_{i,j} | 0\leq i\leq s, 0<j\leq t\}\cup\{U_{i,j} | 0\leq i\leq s, 0\leq j<t\}.$$

For an unit square G whose vertices are $\{(u,v),(u+1,v),(u,v+1),(u+1,v+1)\}$ induces an ideal

$$I^G := \langle U_{u,v}R_{u,v+1} - R_{u,v}U_{u+1,v}, D_{u,v+1}R_{u,v} - R_{u,v+1}D_{u+1,v+1},$$

$$L_{u+1,v+1}D_{u,v+1} - D_{u+1,v+1}L_{u+1,v}, L_{u+1,v}U_{u,v} - U_{u+1,v}L_{u+1,v+1}\rangle$$

generated by 4 binomials which mean the equivalence between two paths from each vertex to the opposite one. We define

$$I^E := I^{(s,t)} := \sum_{G\text{: unit square in } E} I^G.$$

Example 3.

$I^{(1,1)} := \langle U_{0,0}R_{0,1} - R_{0,0}U_{1,0}, D_{0,1}R_{0,0} - R_{0,1}D_{1,1}, L_{1,1}D_{1,0} - D_{1,1}L_{1,0}, L_{1,0}U_{1,1} - U_{1,0}L_{1,1}\rangle$.

$I^{(2,3)}$ is induced from a lattice which has 6 unit squares. Therefore $I^{(2,3)}$ is generated by 24 binomials.

Bibliography

[A16] Aoyama, T.; A SINGULAR package for computing minimal associated primes of binomial ideals. (2016). http://www.math.kobe-u.ac.jp/HOME/taoyama/singular

[A17] Aoyama, T.; An Algorithm for Computing Minimal Associated Primes of Binomial Ideals without Producing Redundant Components. Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. (2017). 21–27.

[AM] Atiyah, M, F., MacDonald, I, G.: Introduction to commutative algebra. Vol. 2. Reading. Addison-Wesley, (1969).

[Ar] Arnold, E, A; Modular algorithms for computing Gröbner bases. Journal of Symbolic Computation 35. (2000) 403–419.

[DGP] Decker, W., Greuel, G,-M., Pfister.; Primary decomposition: Algorithms and comparisons. Algorithmic algebra and number theory, Springer Berlin, Heidelberg, 187-220, (1999).

[DGPS] Decker, W., Greuel, G,-M., Pfister, G., Schönemann, H.: SINGULAR 4-1-0 — A computer algebra system for polynomial computations. **http://www.singular.uni-kl.de** (2016).

[ES] Eisenbud, D., Sturmfels, B.; Binomial ideals. Duke Mathematical Journal 84, 1 (1994) 1-45.

[ESE] Evans, S, N. Sturmfels, B. Euler, C.; Commuting birth-and-death process. the annals of Applied Probability 20,1 (2010), 238-266.

[GG] Gathen, J, V, Z,. Gerhard, J..: Modern Computer Algebra. Cambridge University Press New York, NY, USA. (2003).

[GP] Greuel, G,-M., Pfister, G.: A singular introduction to commutative algebra. (2008).

[IPS] Idrees, N., Pfister, G., Steidel, S.; Parallelization of modular algorithms. Journal of Symbolic Computation 46 (2011) 672-684

[K] Kahle, T.; Decompositions of binomial ideals. Annals of the institute of statistical mathematics 62, 4 (2010), 727-745.

[KN] Kawazoe, T., Noro, M.: Algorithms for computing a primary ideal decomposition without producing intermediate redundant components. Journal of Symbolic Computation 46.10 (2011) 1158–1172

[L1] Laplagne, S.: An algorithm for the computation of the radical of an ideal. Proceedings of the 2006 international symposium on Symbolic and algebraic computation. ACM, (2006)

[L2] Laplagne, S.: Computation of the minimal associated primes. Challenges in Symbolic Computation Software 06271 (2006)

[M2] Grayson,D,R,.Stillman,M,E.; Macaulay2 1.8.2, a software system for research in algebraic geometry. **http://www.math.uiuc.edu/Macaulay2/** (2014).

[NY] Noro, M., Yokoyama, K.; Usage of Modular Techniques for Efficient Computation of Ideal Operations. To appear in Mathematics in Computer Science, DOI 10.1007/s11786-017-0325-1.

[P] Pauer, F.; On lucky ideals for Gröbner basis computations. Journal of Symbolic Computation 14 (1992) 471–482

[S96] Sturmfels, B.; Gröbner bases and convex polytopes. American Mathematical Soc. (1996).

[S02] Sturmfels, B.; Solving systems of polynomial equations. American Mathematical Soc. (2002).

[Z] R, Zippel.: Effective Polynomial Computation. Springer US. (1993).