A Divide and Conquer Method to Compute Binomial Ideals

Deepanjan Kesh[1⋆] and Shashank K Mehta[2]

[1] Indian Institute of Technology Guwahati
Guwahati, India deepkesh@iitg.ernet.in
[2] Indian Institute of Technology Kanpur
Kanpur, India skmehta@iitk.ac.in

Abstract. A binomial is a polynomial with at most two terms. In this paper, we give a *divide-and-conquer* strategy to compute binomial ideals. This work is a generalization of the work done by the authors in [12, 13] and is motivated by the fact that any algorithm to compute binomial ideals spends a significant amount of time computing Gröbner basis and that Gröbner basis computation is very sensitive to the number of variables in the ring. The divide and conquer strategy breaks the problem into subproblems in rings of lesser number of variables than the original ring. We apply the framework on five problems – radical, saturation, cellular decomposition, minimal primes of binomial ideals, and computing a generating set of a toric ideal.

1 Introduction

Consider the polynomial ring $k[x_1, \ldots, x_n]$. A **binomial** in such a ring is a polynomial of the form $c \cdot \mathbf{x}^\alpha + d \cdot \mathbf{x}^\beta$, where $c, d \in k$ and $\alpha, \beta \in \mathbb{N}^n$. An ideal in the polynomial ring which has a generating set comprising only of binomials is called a **binomial ideal**. In this paper, we will be concerned with computing various binomial ideals.

Binomial ideals, unlike general polynomial ideals, possess rich combinatorial structure which can be exploited while computing various structures derived from them, for example Gröbner bases, primary decomposition, and associated primes [17, 10]. Pure difference binomials are binomials of the form $\mathbf{x}^\alpha - \mathbf{x}^\beta$. The varieties of pure difference prime binomial ideals are exactly the toric varieties. Hence, such ideals are also known as toric ideals [7, 6]. There is a large literature studying applications and computations of toric ideals [14, 1]. Moreover, quotients of polynomial rings by pure difference binomial ideals form commutative semigroup rings [9].

Apart from a purely academic interest in the subject of binomial ideals, their study is also motivated by the fact that they are often encountered in interesting problems in diverse fields. These include solving integer programs [11, 3, 18, 16], computing primitive partition identities [14, Chapters 6,7], and solving

⋆ Support from IMPECS is acknowledged.

scheduling problems [15]. In algebraic statistics, closures of discrete exponential families have been identified with nonnegative toric varieties [8].

The theory of binomial ideals was developed in a seminal paper by Eisenbud and Sturmfels [6]. Their paper not only showed various properties of binomial ideals – for example, the radicals and associated primes of binomial ideals are themselves binomial ideals – but they also show how to compute these structures.

In [12], we had dealt with the computation of toric ideals. In [13], we had extended our approach to compute the saturation of binomial ideals. In this paper, we present a general framework to compute several of such binomial ideals, namely radical, saturation, minimal primes and cellular decompositions. This work is motivated by two crucial observations –

1. Most of these computations involve computing a Gröbner basis of certain ideals, and
2. Buchberger's algorithm [2] to compute Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring.

In the light of these observations, we propose a *divide-and-conquer* technique to solve the aforementioned problems, with the hope that this strategy can also be applied to a host of other problems related to binomial ideals, like computing associated primes, primary decomposition, primary component, and so on. The essence of the strategy is the following. Consider the polynomial ring $k[x_1, \ldots, x_n]$, and a binomial ideal $I \subseteq k[x_1, \ldots, x_n]$. We compute the image of I under the natural homomorphism in the derived rings $k[x_2, \ldots, x_n]$ and $k[x_1^{\pm}, x_2, \ldots, x_n]$ and perform the same computation on these ideals(Intuitively, $\mathbf{x}_1^{\pm}$ implies that we allow both positive and negative integers as exponents for x_1). Then we "lift" the results in the original ring and combine them to compute a solution of the original problem. Both these rings are isomorphic to polynomial rings with one less variable [13], hence Gröbner basis (actually such basis does not exist in some of these new rings but we use a variant for the computations) can be computed more efficiently.

The paper has been arranged as follows. In the next section, we briefly present some background required for the paper and define some notations. Section 3 defines two maps from ideals of a Laurent ring to certain derived rings, and state some useful properties of these maps. These two maps form the basis of the reduction of the problem into the subproblems, discussed earlier. Section 4 contains the main contribution of the paper – discussion of the proposed divide-and-conquer framework. In Section 5, we use this framework to compute radical, saturation, cellular decomposition, minimal primes of binomial ideals, and a generating set of a toric ideal.

2 Background

A detailed treatment of the background required for the paper, like the notions of localization, Laurent polynomial rings, or of various kinds of ideals like radical,

prime, saturation, etc., was not included here due to constraint of page limit, but the reader can refer to [4, 5].

We will just state a few notations used in the paper. For a ring R, if $r_1, \ldots, r_s \in R$, then $\langle\ r_1, \ldots, r_n\ \rangle$ will denote the ideal generated by $r_1, \ldots, r_n$. For an ideal $I \subseteq R$, $\sqrt{I} = \{\ r \mid r^m \in I,\ m \geq 0\ \}$ is the radical of I. $I : r^\infty = \{\ s \mid sr^j \in I,$ for some $j \geq 0\ \}$ will denote the saturation of I w.r.t. r.

For a field k, we will use the standard notation of $k[x_1, \ldots, x_n]$ to denote the polynomial ring in n variables. If $U \subseteq R$ is a multiplicatively closed set of R, then $R[U^{-1}]$ is the localization of R w.r.t. U. If the ring $k[x_1, \ldots, x_n]$ is localized w.r.t. $x_1, \ldots, x_i$, then the partial Laurent polynomial ring $k[x_1^{\pm}, \ldots, x_i^{\pm}, x_{i+1}, \ldots, x_n]$ will be denoted by the tuple (k, X, L), where k is the field, X is the set of variables, and L is the set of variables w.r.t. which $k[X]$ has been localized.

3 Two Ring Homomorphisms

3.1 Modulo Map

Let r be an element of a Noetherian ring R. Then $\theta : R \to R/\langle\ r\ \rangle$ denotes the natural homomorphism $\theta(a) = [a] = a + \langle\ r\ \rangle,\ \forall a \in R$. Here, $[a]$ or $a + \langle\ r\ \rangle$ denotes the coset of a in $R/\langle\ r\ \rangle$. This induces a map Θ from the ideals in R containing r and the ideals of $R/\langle r \rangle$ as follows –

$$\Theta(I) = \{\ [a] \mid a \in I\ \},$$

where $I \subseteq R$ is an ideal containing r. Similarly, we define a map Θ^{-1} from the ideals of $R/\langle\ r\ \rangle$ to the ideals of R containing r as follows –

$$\Theta^{-1}(J) = \{\ x \mid [x] \in J\ \},$$

where $J \subseteq R/\langle\ r\ \rangle$ is an ideal. We state, without proof, some basic properties of Θ.

Lemma 1. *The maps Θ and Θ^{-1} have the following properties –*

(i) *Θ and Θ^{-1} preserve set inclusion.*
(ii) *Θ is a bijection.*
(iii) *Θ and Θ^{-1} map primes to primes.*
(iv) *Θ and Θ^{-1} distribute over finite intersections.*
(v) *In a Noetherian ring, $\Theta(\sqrt{I}) = \sqrt{\Theta(I)}$*
(vi) *$\Theta^{-1}(\langle\ [f_1], \ldots, [f_n]\ \rangle) = \langle\ f_1, \ldots, f_n\ \rangle + \langle\ r\ \rangle$*

3.2 Localization map

Let r be a nonzero-divisor of a Noetherian ring R. Let U denote the set of all powers of r, $U = \{\ r^i \mid i \geq 0\ \}$. Since r is not nilpotent, U does not contain zero. U is also multiplicatively closed. Therefore $R[U^{-1}]$ is well defined.

Let $\phi : R \to R[U^{-1}]$ be the natural homomorphism given by $\phi(a) = a/1$, $\forall a \in R$. We define a map, Φ, induced by ϕ, from the ideals in R saturated w.r.t. r to the ideals of $R[U^{-1}]$ as follows –

$$\Phi(I) = \langle \{ a/1 \mid a \in I \} \rangle,$$

where $I \subseteq R$ is an ideal saturated w.r.t. r. Similarly, we will define a map, Φ^{-1}, from the ideals in $R[U^{-1}]$ to the ideals in R which are saturated with respect to r as follows –

$$\Phi^{-1}(J) = \{ a \mid a/r^k \in J,\ k \geq 0 \}.$$

We present some straight forward properties of Φ and Φ^{-1} without proof.

Lemma 2. *The maps Φ and Φ^{-1} have the following properties –*

(i) Φ and Φ^{-1} preserve set inclusion.
(ii) Φ is a bijection.
(iii) Φ and Φ^{-1} map primes to primes.
(iv) Φ and Φ^{-1} distribute over finite intersections.
(v) For $x \in R, \Phi(I : x^\infty) = \Phi(I) : x^\infty$.
(vi) In a Noetherian ring $\Phi(\sqrt{I}) = \sqrt{\Phi(I)}$
(vii) $\Phi^{-1}(\langle f_1/r^{a_1}, \ldots, f_n/r^{a_n} \rangle) = \langle f_1, \ldots, f_n \rangle : r^\infty$.

4 A Divide-and-Conquer Method

In this section, we focus on the main objective of this paper. We present a general algorithm (Algorithm 1) based on *divide-and-conquer* technique which is useful in computing several binomial ideals associated with a given binomial ideal. The algorithm takes as input the following 3 objects (i) A ring (k, X, L), (ii) A set of binomials, S, generating an ideal I, and (iii) A set of variables $V \subseteq X \setminus L$ called *forbidden* set. The objective of the algorithm is to compute $\mathsf{A}(\langle S \rangle)$, where A is some object associated with the binomial ideal I. In Section 5 we demonstrate how Algorithm 1 computes (i) Radical of I, (ii) Saturation of I w.r.t. all the variables in the ring, (iii) Generating basis of a toric ideal from I, (iv) Minimal Primes of I, and (v) Cellular decomposition of I.

We restate, from the introduction, the two crucial observations behind this algorithm –

1. Most computations involving binomial ideals compute Gröbner basis of certain ideals, and
2. Buchberger's algorithm [2] to compute Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring.

The motivation behind the algorithm is to divide the problem suitably into smaller subproblems, solve these subproblems in rings with less variables than the original ring, and combine these results to solve the original problem.

Let $x \in (X \setminus L) \setminus V$, and consider the maps (i) $\Theta : (k, X, L) \to (k, X \setminus \{x\}, L)$, (ii) $\Phi : (k, X, L) \to (k, X, L \cup \{x\})$, and (iii) $f : (k, X, L) \to (k, X, L)$

Algorithm 1: A framework for computing binomials ideals – A

Data: A ring (k, X, L), where k is algebraically closed, and $\mathsf{char}(k) = 0$; forbidden set $V \subseteq X \setminus L$; a binomial generating set S of an ideal in the ring.

Result: $\mathsf{A}(\langle\ S\ \rangle)$

1 **if** $X = \phi$ **then** `// The ring is a field`
2 | Nothing to do ;
3 **else if** $X = L$ **then** `// Laurent polynomial ring`
4 | Compute $\mathsf{A}(\langle\ S\ \rangle)$ and **return** ;
5 **else if** $V = X \setminus L$ **then** `// No more reductions`
6 | Compute $\mathsf{A}(\langle\ S\ \rangle)$ and **return** ;
7 **end**
8 Let $x \in (X \setminus L) \setminus V$;
`/* computing` $\mathsf{A}(\Theta(\langle\ S\ \rangle + \langle\ x\ \rangle))$ `and lift */`
9 Call A with ideal $\Theta(\langle\ S\ \rangle + \langle\ x\ \rangle)$, ring $(k, X \setminus \{x\}, L)$ and forbidden set V ;
10 Compute $\Theta^{-1}(\mathsf{A}(\Theta(\langle\ S\ \rangle + \langle\ x\ \rangle)))$;
`/* computing` $\mathsf{A}(\Phi(\langle\ S\ \rangle : x^{\infty}))$ `and lift */`
11 Call A with ideal $\Phi(\langle\ S\ \rangle : x^{\infty})$, ring $(k, X, L \cup \{x\})$ and forbidden set V ;
12 Compute $\Phi^{-1}(\mathsf{A}(\Phi(\langle\ S\ \rangle : x^{\infty})))$;
`/* computing` $\mathsf{A}(f(\langle\ S\ \rangle : x^{\infty}))$ `*/`
13 Call A with ideal $f(\langle\ S\ \rangle)$, ring (k, X, L) and forbidden set $V \cup \{x\}$;
`/* Computing` $\mathsf{A}(\langle\ S\ \rangle)$ `*/`
14 Combine $\Theta^{-1}(\mathsf{A}(\Theta(\langle\ S\ \rangle + \langle\ x\ \rangle)))$, $\Phi^{-1}(\mathsf{A}(\Phi(\langle\ S\ \rangle : x^{\infty})))$ and $\mathsf{A}(f(\langle\ S\ \rangle))$ to get $\mathsf{A}(\langle\ S\ \rangle)$;
`/* Return */`
15 **return** $\mathsf{A}(\langle\ S\ \rangle)$;

which depends on the problem $\mathsf{A}()$. The reduction step involves solution of the subproblems (i) $\mathsf{A}(\Theta(I + \langle\ x\ \rangle))$, in ring $(k, X \setminus \{x\}, L)$ and forbidden set V(step 9), (ii) $\mathsf{A}(\Phi(I\ :\ x^{\infty}))$, in ring $(k, X, L \cup \{x\})$ and forbidden set V(step 11), (iii) $\mathsf{A}(f(I))$ in ring (k, X, L) and forbidden set $V \cup \{x\}$(step 13). The first subproblem is in a ring with one less variable compared to the original ring. In the case of the second subproblem, Gröbner bases are not defined in the context of partial Laurent polynomial rings (k, X, L). But pseudo-Gröbner bases [13], briefly discussed later in this section, can effectively substitute Gröbner bases for binomial ideal computations. The time complexity of the algorithm to compute pseudo Gröbner basis was shown in that paper to be dependent on the number of variables in $X \setminus L$. Hence, this subproblem is also justifiably "smaller".

The role of the forbidden set of variables is that reduction must not be done with respect to these variables. Thus, if $V = X \setminus L$, then the computation $A(I)$ must be easy to perform without further reduction. In addition, the third subproblem should be such that it does not require the computation of a Gröbner basis since in this case the ring is same as in the original problem and involves no reduction in ring size. Here is a motivating example to justify the use of forbidden set. Suppose we want to compute the saturation, $I : (x_1 \cdots x_n)^{\infty}$,

while I is already saturated w.r.t. x_1, x_2. Then reduction with these variables is futile. Hence we can put these variables in the forbidden set.

Next, the algorithm computes the inverse images of $\mathsf{A}(\Theta(I + \langle\ x\ \rangle))$ (step 10) and $\mathsf{A}(\Phi(I : x^\infty))$ (step 12) in the original ring (k, X, L). In the applications discussed in the next section, $\mathsf{A}(I)$ is either an ideal (as in the case of radical of I) or a set of ideals (as in the case of minimal primes of I). Hence these images are well defined. Abusing notation, we denote these inverse images respectively by $\Theta^{-1}(\mathsf{A}(\Theta(I + \langle\ x\ \rangle)))$ and $\Phi^{-1}(\mathsf{A}(\Phi(I : x^\infty))$.

Finally in step 14, $\mathsf{A}(I)$ is to be constructed from these images and $\mathsf{A}(f(I))$. One can easily observe that the algorithm terminates, as in each step either cardinality of X decreases, or that of L or V increases. This algorithm is a general method and can be tuned to a particular problem by specifying the following three steps in the context of that problem –

(steps 4, 6) $V = X \setminus L$: Give the method to compute $A(I)$ in these base cases.

(step 13) : Specify function f.

(step 14) : Show how to combine the results of the subproblems.

In the next few subsections we show how to compute Θ, Φ, and their inverses using a generating set of the input ideal.

4.1 Computing Modulo

Let $L = \{y_1, \ldots, y_k\}$ and $X = \{x_1, \ldots, x_l\} \cup \{z\} \cup L$. Maps θ and Θ from $(k, X, L) \to (k, X \setminus \{z\}, L)$ are computed as follows. Consider an arbitrary polynomial in (k, X, L), $f = \sum_i \mathbf{x}^{\alpha_i}\mathbf{y}^{\beta_i} + \sum_j \mathbf{x}^{\alpha_j}\mathbf{y}^{\beta_j}z^{c_j}$. Then, $\theta(f) = \sum_i \mathbf{x}^{\alpha_i}\mathbf{y}^{\beta_i}$. Further, suppose $S \subset (k, X, L)$ is a set of binomials. Then, $\Theta(\langle\ S\ \rangle) = \langle\ \theta(f) \mid f \in S\ \rangle$. Conversely, if $S' \subset (k, X \setminus \{z\}, L)$, then $\Theta^{-1}(\langle\ S'\ \rangle) = \langle\ S' \cup \{z\}\ \rangle$, from Lemma 1.

4.2 Computing Localization

Consider the ring (k, X, L) as defined in the previous subsection. If $g \in (k, X, L)$, then $\phi(g) = g/1$.

Computing Φ and Φ^{-1} is also easy. For any $S \subset (k, X, L)$, $\Phi(\langle\ S\ \rangle) = \langle\ \{\ g/1 \mid g \in S\ \}\ \rangle$. In the reverse direction, for any $S' \subset (k, X, L \cup \{z\})$, we define $\Phi^{-1}(\langle\ S'\ \rangle)$ as follows. Let $S' = \{g_1/z^{a_1}, \ldots, g_k/z^{a_k}\}$. Then $\Phi^{-1}(\langle\ S'\ \rangle) = \langle\ g_1, \ldots, g_k\ \rangle : z^\infty$. The correctness follows from Lemma 2.

To see how we can compute saturation with respect to z in a partial Laurent polynomial ring, we briefly revisit the results on *pseudo-Gröbner basis* in [13].

4.3 pseudo-Gröbner Basis

Gröbner bases are defined for ideals in rings $k[x_1, \ldots, x_n]$ ([4, Chapter 2]). This notion has been generalized for binomial ideals in partial Laurent polynomial

rings, called pseudo-Gröbner bases in [13, Section 5]. Here we reproduce some relevant results.

Definition 1. *A binomial* $a\mathbf{x}^{\alpha} + b\mathbf{x}^{\beta} \in (k, X, L)$ *is said to be* **balanced** *if* $x_i \in X \setminus L$ *implies* $\alpha_i = \beta_i$.

Definition 2. *For every finite binomial set* G, G_1 *and* G_2 *will denote its partition, where the former will represent the set of non-balanced binomials and the latter will represent the set of balanced binomials of* G.

Definition 3. *A binomial basis* $G = (G_1, G_2)$ *of a binomial ideal* I *will be called a pseudo Gröbner basis with respect to a given term-order, if* G_1 *reduces every binomial of* I *to* $0(\mathsf{mod}(G_2))$.

Theorem 1. *[13, Theorem 3] Every binomial ideal in* (k, X, L) *has a pseudo-Gröbner basis with respect to any term-order.*

The Buchberger's algorithm to compute Gröbner basis has been adopted to compute pseudo-Gröbner basis in [13, Algorithm 4]. Finally, the following theorem shows that saturation can be computed in similar way as in $k[x_1, \ldots, x_n]$.

Theorem 2. *[13, Theorem 3] Let* (G_1, G_2) *be a pseudo Gröbner basis of a homogeneous binomial ideal in* (k, X, L) *with respect to a graded reverse lexicographic term order with the variable* $x_i \notin L$ *being the least. Then* $(G_1' = G_1 \div x_i^{\infty}, G_2' = G_2 \div x_i^{\infty})$ *is a pseudo Gröbner basis of* $I : x_i^{\infty}$.

Here $S \div x^{\infty}$ is the result of the division of each polynomial in S by the largest possible power of x.

5 Computing $\mathsf{A}(I)$

As mentioned in the previous section, we will describe the steps 4, 6, 13 and 14 of the algorithm in context of five problems – (i) radical of a binomial ideal, (ii) the saturation of a binomial ideal with respect to all variables in the ring, (iii) computing toric ideal, (iv) the minimal prime ideals of a binomial ideal, and (v) cellular decomposition of a binomial ideal.

5.1 Radical Ideal: A = Radical

Theorem 3. *Let* R *be a Noetherian ring,* $r \in R$ *a non-zero-divisor, and* $I \subseteq R$ *be an ideal. Then,* $\sqrt{I + \langle\, r \,\rangle} \cap \sqrt{I : r^{\infty}} = \sqrt{I}$, *for some* $r \in R$.

Proof. We know that every radical ideal in a Noetherian ring has a prime decomposition. Let the prime decomposition of $\sqrt{I}$ be $\sqrt{I} = P_1 \cap P_2 \cap \ldots \cap P_n$. Let the collection of the primes in the decomposition be denoted by $\mathcal{P}$. Define two ideals $\mathcal{P}_r = (\cap_{r \in P \in \mathcal{P}} P)$, and $\overline{\mathcal{P}_r} = (\cap_{r \notin P \in \mathcal{P}} P)$. It is easy to see that $I + \langle\, r \,\rangle \subseteq \mathcal{P}_r$. Hence, $\sqrt{I + \langle\, r \,\rangle} \subseteq \mathcal{P}_r$. Next, we want to show that $\sqrt{I : r^{\infty}} \subseteq \overline{\mathcal{P}_r}$.

Let $f \in I : r^\infty$. Then, $r^n f \in I$ for some $n \geq 0$. This implies that for all $P \in \mathcal{P}, r^n f \in P$. In particular, if $r \notin P$, then $f \in P$. We deduce that $I : r^\infty \subseteq \overline{\mathcal{P}_r}$, and hence $\sqrt{I : r^\infty} \subseteq \overline{\mathcal{P}_r}$. Putting the two observation together we have $\sqrt{I + \langle\ r\ \rangle} \cap \sqrt{I : r^\infty} \subseteq \mathcal{P}_r \cap \overline{\mathcal{P}_r} = \sqrt{I}$

The converse containment $\sqrt{I} \subseteq \sqrt{I + \langle\ r\ \rangle} \cap \sqrt{I : r^\infty}$ is obvious. □

The following theorem will help us in the formulation of step 14.

Theorem 4. *Let R be a Noetherian ring, $r \in R$ a non-zero-divisor, and $I \subseteq R$ be an ideal. Then, $\sqrt{I} = \Theta^{-1}(\sqrt{\Theta(I + \langle\ r\ \rangle)}) \cap \Phi^{-1}(\sqrt{\Phi(I : r^\infty)})$.*

Proof. We will continue to use the notations defined in the previous theorem. From the proof of Theorem 3, we have $I + \langle r \rangle \subseteq \mathcal{P}_r$. From the containment preserving property and the commutation with intersection property of Θ, we have $\Theta(I + \langle r \rangle) \subseteq \Theta(\cap_{r \in P \in \mathcal{P}} P) = \cap_{r \in P \in \mathcal{P}} \Theta(P)$. Similarly $\sqrt{\Theta(I + \langle r \rangle)} \subseteq \sqrt{\cap_{r \in P \in \mathcal{P}} \Theta(P)} = \cap_{r \in P \in \mathcal{P}} \sqrt{\Theta(P)}$. The last equality is due to the fact that intersection of radicals is equal to the radical of intersections.

As the Ps are primes, from Lemma 1 we know that the $\Theta(P)$s are primes and since prime ideals are radical, we have $\sqrt{\Theta(I + \langle r \rangle)} \subseteq (\cap_{r \in P \in \mathcal{P}} \Theta(P))$. Hence $\Theta^{-1}(\sqrt{\Theta(I + \langle r \rangle)}) \subseteq \mathcal{P}_r$.

Similarly, starting from the following relation given in the proof of theorem 3 $I : r^\infty \subseteq \overline{\mathcal{P}_r}$ we can deduce that $\Phi^{-1}(\sqrt{\Phi(I : r^\infty)}) \subseteq \overline{\mathcal{P}_r}$. Combining the two results gives $\Theta^{-1}(\sqrt{\Theta(I + \langle r \rangle)}) \cap \Phi^{-1}(\sqrt{\Phi(I : r^\infty)}) \subseteq \sqrt{I}$.

To prove the converse, from Lemmas 1 and 2 we have
$\sqrt{I} \subseteq \Theta^{-1}(\sqrt{\Theta(I + \langle\ r\ \rangle} \cap \Phi^{-1}(\sqrt{\Phi(I : r^\infty)})$. □

We will not use the $\mathsf{A}(f(I))$ branch of the reduction for this problem. Thus, Theorem 3 shows that the *combine* step (step 14) is intersection. Also, we will have $V = \emptyset$. The base case computation in step 4 of the algorithm is trivial because all binomial ideals in a Laurent polynomial ring are already radical as shown below.

Theorem 5 (Corollary 2.2, [6]). *Let J be a binomial ideal in the ring (k, X, ϕ). Then, if k is algebraically closed and $\mathsf{char}(k) = 0$, then $J : (\Pi_{x \in X} x)^\infty$ is radical.*

Corollary 1. *Let k be an algebraically closed field, with $\mathsf{char}(k) = 0$. Then, all binomial ideals in (k, X, X) are radical.*

Proof. Let J be a binomial ideal in the ring (k, X, X), where $X = \{x_1, \ldots, x_n\}$. Consider the ideal localization map, Φ_n, from $(k, X, X \setminus \{x_n\})$ to (k, X, X). Under this map, we know that $\Phi_n^{-1}(J)$ is saturated w.r.t x_n. Similarly, if we consider the map Φ_{n-1} from $(k, X, X \setminus \{x_{n-1}, x_n\})$ to $(k, X, X \setminus \{x_n\})$, then the ideal $\Phi_{n-1}^{-1}(\Phi_n^{-1}(J))$ is saturated w.r.t. x_{n-1}. So we have $\Phi_n^{-1}(J) = \Phi_n^{-1}(J) : x_n^\infty$. Hence, $\Phi_{n-1}^{-1}(\Phi_n^{-1}(J)) = \Phi_{n-1}^{-1}(\Phi_n^{-1}(J) : x_n^\infty) = \Phi_{n-1}^{-1}(\Phi_n^{-1}(J)) : x_n^\infty$ (Lemma 2) Thus, $\Phi_{n-1}^{-1}(\Phi_n^{-1}(J))$ is saturated w.r.t. $\{x_{n-1}, x_n\}$. Continuing this argument we see that $\Phi_1^{-1}(\cdots(\Phi_n^{-1}(J))\cdots)$, in the ring (k, X, ϕ), is saturated w.r.t. $\{x_1, \ldots, x_n\}$. From the previous theorem $\Phi_1^{-1}(\cdots(\Phi_n^{-1}(J)))$ is radical. Now, by repeated application of Lemma 2 we deduce that J is radical too. □

Analysis: The proposed algorithm uses two out of the three branches of the *Divide-and-Conquer* strategy (Algorithm 1), so if n is the number of variables in the input ideal, this algorithm requires 2^n Gröbner basis computations. Compare this with $n!$ computations in [6, Algorithm 9.1].

5.2 Saturation : A = Saturation

Suppose I is saturated with respect to $\{x_{i_1}, \ldots, x_{i_j}\}$ then we begin the computation with $V = \{x_{i_1}, \ldots, x_{i_j}\}$. For this problem, we only use the $\mathsf{A}(I : x^\infty)$ branch of the reduction. The base case for this algorithm occurs when $X \setminus L = V$ (step 6). As Φ preserves saturation (Lemma 2), the ideal is already saturated in this case. Since the algorithm uses only one branch of the reduction, step 14 is redundant.
Analysis: In this proposed algorithm, the number of variables in the image space is 1 in the first iteration, 2 in the second iteration, and so on. Symbolically, if $\mathcal{G}(k)$ denotes the time complexity of Buchberger's algorithm in a k variable ideal, then the cost of the proposed algorithm is $\sum_{k=1}^{n} \mathcal{G}(k)$, where n is the number of variables in input ideal. On the other hand, the cost of the Sturmfels' algorithm [14, Lemma 12.1] is $n\mathcal{G}(n)$.

5.3 Toric Ideals: A = Toric

Pure difference prime binomial ideals are called toric ideals. So, they are a special class of general binomial ideals and, as pointed out in Section 1, perhaps the most useful of all binomial ideals from an application perspective. The goal in this case is also to saturate a given binomial ideal, but we are guaranteed that the saturated ideal will be a toric ideal. The solution of Section 5.2 applies to toric ideals as well and our proposed algorithm do not exploit the fact that the solution is known to be a toric ideal. But there are algorithms that do, namely the *project and lift* algorithm due to Hemmecke and Malkin [10], and it is much faster than the Sturmfels' Algorithm alluded to in the previous section.
Analysis: Using the notation $\mathcal{G}(k)$ introduced in the previous section, the cost of *project and lift* algorithm is $\sum_{i=k}^{n} \mathcal{G}(i) + k\mathcal{G}(k)$, where k is dependant on the input. n, as in the previous cases, denote the number of variables in the input ideal. We note that the cost of the proposed algorithm is $\sum_{i=1}^{n} \mathcal{G}(i)$. Thus, the proposed algorithm matches *project and lift* in the worst case, and does better in all other cases.

5.4 Prime Decomposition: A = Prime

In this case, as in the computation of a radical, the $\mathsf{A}(f(I))$ branch will not be used. We will first handle the base case, i.e. how to compute the minimal primes of a binomial ideal in a Laurent polynomial ring (step 4). To do this, we will mention (without proof) a set of results from [6].

Definition 4. *A* **partial character** *on* $\mathbb{Z}^n$ *is a homomorphism* ρ *from a sublattice* L_ρ *of* $\mathbb{Z}^n$ *to the multiplicative group* $k^*(= k\backslash\{0\})$. *A partial character will always refer to the tuple* (ρ, L_ρ).

For a binomial ideal I in (k, X, X), let $L(I) = \{\ \alpha \mid \mathbf{x}^\alpha - c \in I\ \}$. It is easy to verify that $L(I)$ is a lattice. We define a function ρ as $\rho(\alpha) = c$, where $\mathbf{x}^\alpha - c \in I$. Thus, $(\rho, L(I))$ is a partial character. Conversely, given a partial character (ρ, L), we will define a binomial ideal as $I(\rho) = \langle \mathbf{x}^\alpha - c | \alpha \in L, \rho(\alpha) = c \rangle$.

Theorem 6. *For any proper binomial ideal in* (k, X, X), *there is a unique partial character* ρ *on* $\mathbb{Z}^n$ *such that* $I = I(\rho)$.

Definition 5. *If* L *is a sublattice of* $\mathbb{Z}^n$, *then the saturation of* L *is the lattice* $\mathsf{Sat}(L) = \{\ m \in \mathbb{Z}^n \mid dm \in L$ *for some* $d \in \mathbb{Z}\ \}$.

We can compute $\mathsf{Sat}(L)$ for any lattice L by a change of variables in (k, X, X).

Definition 6. *If* (ρ, L_ρ) *is a partial character, any partial character* $(\rho', Sat(L_\rho))$ *is called a* **saturation** *of* (ρ, L_ρ) *if* ρ' *coincides with* ρ *when restricted to* L_ρ.

Theorem 7. *If* g *is the order of the group* $\mathsf{Sat}(L_\rho)/L_\rho$, *then there are* g *distinct saturations of* ρ: $\rho_1, \ldots, \rho_g$. *Also* $I(\rho) = \cap_{j=1}^{g} I(\rho_j)$.

Theorem 8. *The radical of a cellular ideal is of the form* $I(\rho) + M(\mathscr{E})^{(d)}$ *(*d *is vector with all 1s), and its minimal primes are the lattice ideals with the saturations of* ρ.

So in a Laurent polynomial ring, to determine the set of minimal primes of a binomial ideal $I = I(\rho)$, all we need to do is to compute the saturations of ρ. The lattice ideals corresponding to these saturations are the associated primes of $I(\rho)$. The minimal of these ideals constitute the prime decomposition.

Now, let us discuss how we can combine the results from the modulo and the localization branch (step 14). From the recursive calls of the algorithm we have computed the minimal primes of $\Theta(I + \langle\ r\ \rangle)$ and $\Phi(I : r^\infty)$. Let the set of minimal primes be denoted by $\mathcal{P}_\Theta$ and $\mathcal{P}_\Phi$, respectively. So, we have $\sqrt{\Theta(I + \langle\ r\ \rangle)} = \cap_{P \in \mathcal{P}_\Theta} P$, $\sqrt{\Phi(I : r^\infty)} = \cap_{P \in \mathcal{P}_\Phi} P$. From Theorem 4, we have $\sqrt{I} = \Theta^{-1}(\sqrt{\Theta(I + \langle\ r\ \rangle)}) \cap \Phi^{-1}(\sqrt{\Phi(I + \langle\ r\ \rangle)})$. Thus $I = (\cap_{P \in \mathcal{P}_\Theta} \Theta^{-1}(P))$ $\cap\ (\cap_{P \in \mathcal{P}_\Phi} \Phi^{-1}(P))$. We know that Θ and Φ map primes to primes (Lemmas 1 and 2). The desired set of prime ideals is $\{\ \Theta^{-1}(P) \mid P \in \mathcal{P}_\Theta\ \} \cup \{\Phi^{-1}(P) \mid$ $P \in \mathcal{P}_\Phi\}$. We just need to remove the redundant ones.

Analysis: In this case we have only used the modulo and localization branches. So, the cost of the algorithm is 2^n Gröbner basis computations for an input ideal containing n variables. The cost of algorithm 9.2 proposed in [6] is 2^n iterations, where in each iteration a Gröbner basis and a cellular decomposition is computed. Our proposed solution, thus, has gotten rid of the necessity of computing cellular decomposition in each of the 2^n iterations.

5.5 Cellular Decomposition: A = Cellular

In this section we will generalize the notion of **cellular ideals** to partial Laurent polynomial rings, establish that every ideal has a cellular decomposition, and use our framework to compute such a decomposition.

Let (k, X, L) be the underlying partial Laurent polynomial ring. For a given set of variables $\mathscr{E} \subseteq (X \setminus L)$ and an integer vector $d = (d_i)_{i \in (X \setminus L) \setminus \mathscr{E}}$, the ideal $M(\mathscr{E})^{(d)}$ is defined as $\langle \{ x_i^{d_i} \mid i \in (X \setminus L) \setminus \mathscr{E} \} \rangle$.

Definition 7. *An ideal I of (k, X, L) is* **cellular** *if for some $\mathscr{E} \subseteq (X \setminus L)$, we have $I = I : (\prod_{i \in \mathscr{E}} x_i)^\infty$ and I contains $M(\mathscr{E})^{(d)}$ for some vector d.*

Observation 1. *An ideal I is cellular iff $\exists \mathscr{E} \subseteq (X \setminus L)$ and an integer vector $d = (d_i)_{i \in (X \setminus L) \setminus \mathscr{E}}$, such that $I = (I + M(\mathscr{E})^d) : (\prod_{i \in \mathscr{E}} x_i)^\infty$. It is denoted by $I_{\mathscr{E}}^{(d)}$.*

Lemma 3. *Φ^{-1} preserves cellular ideals.*

Proof. Let Φ^{-1} be a map from (k, X, L) to $(k, X, L \setminus \{x\})$, where $x \in L$, and consider the cellular ideal $I = I_{\mathscr{E}}^{(d)}$ in (k, X, L). As $\Phi^{-1}(I)$ is saturated w.r.t. x, it corresponds to the cellular ideal $\Phi^{-1}(I)_{\mathscr{E} \cup \{x\}}^{(d')}$, where d' is the same vector as d, except that it does not contain the component corresponding to x. □

Lemma 4. *Let $s \in \mathbb{N}$ be such that $I : r^s = I : r^\infty$ in some Noetherian ring R. Then, $I = (I + \langle r^s \rangle) \cap (I : r^s)$.*

Proof. Let $g \in (I + \langle r^s \rangle) \cap (I : r^s)$. Then $g = i + hr^s \in I : r^s$ for some $i \in I, h \in R \implies gr^s = ir^s + hr^{2s} \in I$. This, coupled with the fact that $I : r^{2s} = I : r^s$, we have $g \in I$. □

Now we state how to compute a cellular decomposition of I. The computation will not use $\mathsf{A}(\Theta(I))$ branch of the reduction. $f(I)$ is defined as $I + \langle x^s \rangle$, where $s \in \mathbb{N}$ is such that $I : x^s = I : x^\infty$. By using Lemma 3, we see that cellular decomposition of $\Phi(I : x^\infty)$ gives us a cellular decomposition of $I : x^s$. To combine the decompositions of $\mathsf{A}(I : x^s)$ and $\mathsf{A}(f(I))$, we use Lemma 4.

Ideals in the base cases (i.e., $X = L \cup V$) are already cellular because variables in $V = X \setminus L$ are nilpotents of the ideals. Hence, there is no computation required in steps 4 and 6.

Analysis: As our algorithm uses only two branches, the cost of our algorithm is 2^n Gröbner basis computations for an input ideal containing n variables. Algorithm 9.3 of [6] also needs to perform the same number of Gröbner basis computations. So, in this case, we do not see an improvement in the performance of our algorithm over existing algorithms. Advantage, if any, is the proposed generalized and unified approach which, according to the authors, is much simpler and cleaner.

References

1. Bigatti, A.M., Scala, R., Robbiano, L.: Computing toric ideals. J. Symb. Comput. 27(4), 351–365 (1999)
2. Buchberger, B.: A theoretical basis for the reduction of polynomials to canonical forms. SIGSAM Bull. 10(3), 19–29 (1976)
3. Conti, P., Traverso, C.: Buchberger algorithm and integer programming. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes pp. 130–139 (1991)
4. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
5. Eisenbud, D.: Commutative Algebra with a View toward Algebraic Geometry. Springer Verlag, New York (1995)
6. Eisenbud, D., Sturmfels, B.: Binomial ideals. Duke Mathematical Journal 84(1), 1–45 (1996)
7. Fulton, W.: Introduction to toric varieties, Annals of Mathematics Studies, vol. 131. Princeton University Press, Princeton, NJ (1993)
8. Geiger, D., Meek, C., Sturmfels, B.: On the toric algebra of graphical models. The Annals of Statistics 34(3), 1463–1492 (2006)
9. Gilmer, R.: Commutative semigroup rings. University of Chicago Press, Chicago, Illinois (1984)
10. Hemmecke, R., Malkin, P.N.: Computing generating sets of lattice ideals and Markov bases of lattices. Journal of Symbolic Computation 44(10), 1463–1476 (2009)
11. Hosten, S., Sturmfels, B.: Grin: An implementation of Gröbner bases for integer programming. Integer Programming and Combinatorial Optimization (1995)
12. Kesh, D., Mehta, S.K.: Generalized Reduction to Compute Toric Ideals. In: Proceedings of the 20^{th} International Symposium on Algorithms and Computation(ISAAC). pp. 483–492. Honolulu, Hawaii, USA (16–18, December 2009)
13. Kesh, D., Mehta, S.K.: A Saturation Algorithm for Homogeneous Binomial Ideals. In: Wang, W., Zhu, X., Du, D.Z. (eds.) Combinatorial Optimization and Applications: 5^{th} International Conference, COCOA 2011. Lecture Notes in Computer Science, vol. 6831, pp. 357–371. Springer, Zhangjiajie, China (4–6, August 2011)
14. Sturmfels, B.: Gröbner Bases and Convex Polytopes, University Lecture Series, vol. 8. American Mathematical Society (December 1995)
15. Tayur, S.R., Thomas, R.R., Natraj, N.R.: An algebraic geometry algorithm for scheduling in the presence of setups and correlated demands. Mathematical Programming 69(3), 369–401 (1995), citeseer.ist.psu.edu/tayur94algebraic.html
16. Thomas, R., Weismantel, R.: Truncated Gröbner bases for integer programming. Applicable Algebra in Engineering, Communication and Computing 8(4), 241–256 (4 1997), dx.doi.org/10.1007/s002000050062
17. Thomas, R.R.: A Geometric Buchberger Algorithm for Integer Programming. Mathematics of Operations Research 20, 864–884 (1995)
18. Urbaniak, R., Weismantel, R., Ziegler, G.M.: A variant of the Buchberger algorithm for integer programming. SIAM J. Discret. Math. 10(1), 96–108 (1997)